Crossroads
The ACM Student Magazine

A Human's Eye View: Motion Blur and Frameless Rendering

by ***Ellen J. Scher Zagier***

Abstract

Frameless Rendering (**FR**) is a rendering paradigm which performs stochastic temporal filtering by updating pixels in a random order, based on most recent available input data, and displaying them to the screen immediately [1]. This is a departure from frame-based approaches commonly experienced in interactive graphics. A typical interactive graphics session uses a single input state to compute an entire frame. This constrains the state to be known at the time the first pixel's value is computed. Frameless Rendering samples inputs many times during the interval which begins at the start of the first pixel's computation and ends with the last pixel's computation. Thus, Frameless Rendering performs temporal supersampling - it uses more samples over time. This results in an approximation to motion blur, both theoretically and perceptually.

This paper explores this motion blur and its relationship to: camera open shutter time, current computer graphics motion-blur implementations, temporally anti-aliased images, and the Human Visual System's (HVS) motion smear quality **(see 'quality' footnote)** [2].

Finally, we integrate existing research results to conjecture how Frameless Rendering can use knowledge of the Human Visual System's blurred retinal image to direct spatiotemporal sampling. In other words, we suggest importance sampling **(see 'sampling' footnote)** by prioritizing pixels for computation based on their importance

to the visual system in discerning what is occurring in an interactive image sequence.

Introduction

Good user interaction in computer graphics is best achieved when there is a tight coupling between user input and system response. The delayed reaction by the computer graphics systems is known as **latency**. Latency is a serious deterrent to achieving seamless user interaction. This is particularly noticeable in Virtual Environments systems using a Head Mounted Display interface. If the scene does not change as soon as the user's head moves, the goal of simulating the real world is thwarted.

Frameless Rendering is an image updating and display strategy designed to reduce apparent latency in virtual environment applications. Information is presented to the user as soon as it becomes available. Further, this information is based on the most current user input data. Frameless Rendering has been shown to successfully smooth motion and reduce latency. Frameless Rendering introduces distracting artifacts that appear as noise due to the stochastic sampling. But the noise has structure which has the advantage of producing a motion blur side effect.

This paper explores what motion blur and Frameless Rendering have in common as well as differentiating between the two. Included are related topics such as temporal aliasing, camera exposure time, and motion smear.

With the help of existing research, an approach for importance sampling in regions projecting onto lowest retinal velocity is suggested.

Previous Frameless Rendering Results

Frameless Rendering: Researchers at UNC Chapel Hill illustrated that using *Frameless Rendering* [**1**] can offer a more fluid animation than traditional double buffering. This translates to quicker response time in interactive applications, with only a slight degradation of image quality. The proof of concept was illustrated via simulations where a random percentage of pixels was updated at each time increment and compared with a double-buffered animation with equal pixel computation budget. The double-buffered frame updates occur only after all pixels have been computed. For example, a 5 *fps* double-buffered animation is compared with a Frameless Rendering 15 *fps* update. In the latter case 33% of the pixels are updated every 67 milliseconds (*ms*) resulting in more fluid motion. Pixels are selected randomly without replacement so as to eliminate the tearing of artifacts associated with single buffering. The Frameless Rendering animation sequence has new, current information when it hits the screen. It has some pixels that are as much as 200 *ms* old, but some which are only 67 *ms* old. For an equivalent pixel budget, the double-buffered animation exhibits visibly abrupt motion and only updates with new information every 200 *ms*. Further, computation began using system state 200 *ms* earlier. So, the pixels are already 200 *ms* old by the time they first hit the screen and up to 400 *ms* old by the time the start of the next frame's update to the display.

Frameless Anti-aliasing: Spatial anti-aliasing is computationally expensive. Good anti-aliasing with 16 samples per pixel can multiply compute time by an equivalent factor of 16 or higher. In applications requiring tight coupling of user input to system response, the slowdown impedes task execution. In a frameless environment, though, there is no reason to be confined to an all-or-nothing extreme. If there is time for *x* number of samples to be computed at time *t* then all *x* samples should be computed and displayed. If there is not enough time to compute all samples then Frameless anti-aliasing theory [**8**] suggests updating as many samples as possible, but choosing those samples pseudo-randomly to avoid visible artifactual patterns.

A side-by-side comparison illustrates a clear advantage to updating samples as computed versus waiting for all samples to be computed. The smooth motion of a slightly aliased image is preferable in many applications to the staccato motion of fewer, but higher quality frames. Frameless Anti-aliasing exhibits the automatic adaptive refinement feature of Frameless Rendering, that is, pixels converge to their current value as motion slows to a halt.

Frameless Raytracing: A Frameless Rendering Raytracer is implemented by creating a hybrid of Frameless Rendering - randomized order of pixel updates - and the graphics hardware z-buffer. The result is a virtual environment implementation of a real-time raytracer. Although there are some distracting visual artifacts, it captures many reflection effects, including shadows, and performs at interactive rates (approximately 4 *fps*).

Frameless Rendering as a Temporal Supersampler

Supersampling in computer graphics refers to a technique originally developed to ameliorate the problems associated with spatial aliasing or ``jaggies''. A lower frequency alias can pose as the actual frequency because the values where the original signal is sampled are identical to the values of the lower frequency signal. Figure **1** illustrates the origin of aliasing artifacts.

Figure 1

By calculating samples above the final pixel resolution and averaging those values into the final samples, a smoothing effect occurs and the higher frequency is perceived.

A signal can occur in time as well as in space. When motion is misinterpreted from an image sequence, this indicates that an inadequate temporal sampling rate is to blame. This can be due to low frame rates, high speed object motion, or a combination of the two. The result is temporal aliasing. This is when a (typically lower) frequency signal, an alias of the actual signal, is perceived.

Temporal supersampling is in principle similar to spatial supersampling. If we calculate samples above the final frame rate and use those values, we are performing temporal supersampling. This will allow for an accurate perception of higher frequencies than if no supersampling was performed.

Frameless Rendering is a temporal supersampler because a higher rate of time samples is used than if it was not used. Although Frameless Rendering may show snapshots **(see "snapshots" footnote)** at the same rate as it samples inputs, it is a temporal supersampler when contrasted with its double-buffering counterpart. Frameless Rendering is a spatial subsampler as well. That is, fewer spatial samples are computed for each Frameless Rendering snapshot. Motion blur implementations in computer graphics [3, 6] perform temporal supersampling as well. They simulate the action of an open camera shutter.

Motion Blur

Motion blur is the effect that arises when a camera shutter remains open for an extended period of time **(see "time" footnote")** and the motion that has occurred over this interval is visible in a single snapshot. Motion blur, when used in film or animation sequences, has nice qualities such as capturing the perceptual effect of high speed motion more accurately. The introduction of motion blur to a computer-generated image sequence typically involves some form of temporal supersampling. That is, a higher density of samples is taken in the time dimension than the final frame rate of the image sequence.

Early attempts at motion blur were designed as a solution to temporal aliasing as well as mimicking a camera shutter.

One strategy for introducing motion blur is to convolve each frequency domain image with a point spread function [7]. The point spread function encapsulates the motion in the scene over an interval. The result is a blurred spatial domain image.

Another method, again with the goal of capturing the motion occurring in between frames, is a straightforward temporal supersampling method based on spatial supersampling techniques [6]. A complete image sequence is computed at a higher frame rate than the final 30 *fps* playback rate. The experiment illustrated in the paper computes frames 4 times as densely, a total of 120 frames of information per second of final animation. These more densely packed frames are averaged together with a filter, to produce the final 30 *fps* motion. A static frame exposes visible, discrete pieces of the image, an artifact inherent to this method. In many cases, this artifact is visible temporally as well, and the desired motion blur effect is unachievable. Some Frameless Rendering implementations exhibit a similar artifact when inputs are not sampled densely enough. See Figure 2.

Figure 2

A method very similar to the temporal supersampling method just described was developed for raytraced images. It is known as distributed ray tracing [3]. Distributed ray tracing assumes an *a priori* expense for spatial anti-aliasing, such as 16 samples per pixel. These samples are typically derived from system input state at a single instance in time. Distributed ray tracing's spatial samples are distributed temporally as well as spatially to introduce a motion blur effect. That is, the samples are computed from an interval of input states.

The above methods offer an important benefit: Their long-duration open shutter simulation alleviates the aliasing artifacts due to high velocity motion. There are tradeoffs, though. The one tradeoff they have in common is that they are too computationally expensive to permit realtime motion blur. Frames must be precomputed and then played back at high frame rates.

A useful realtime motion blur scheme considers an object's trajectory between time t_1 and t_2. It sweeps the volume over this time interval, essentially interpolating contours. It achieves interactive frame rates by making use of high-speed hardware-assisted renderers. It is a good early attempt at real-time motion blur accommodating many situations that may arise in synthetic environments. Still uncharted, though, is handling occlusions, complex rotations, fast-moving objects, and independently moving vertices.

A natural by-product of Frameless Rendering is that rays are distributed in time and there is no increased computational expense. It is based on actual sampling of motion, and although it does not claim to be a motion blur solution, it does not suffer from producing inaccurate information. Because it increases apparent frame rates, it can actually perform in real-time in cases where its double-buffered counterpart is unable to. See Table **1** for a comparison of Frameless Rendering with other motion blur schemes. The method labeled `Temporal Supersampling' is the method described in this section. It is assumed to use evenly distributed samples over time, although the researchers experimented with other alternatives.

Table 1

Motion Picture Industry: The motion picture and television industry has long been concerned with taking shortcuts in image information content without sacrificing the overall perceptual quality of images. When stills from hand-drawn animated sequences are analyzed, there is often little resemblance to the visual effect of the temporally viewed sequence. A well-known example of this is an automobile accelerating across a television screen. Some of the in-between cells **(see "cells" footnote)** have little or no detail at all. The car appears as an elongated blur, but temporally, a high fidelity representation of the car is inferred for the entire sequence.

The findings and tricks of the motion picture and television trade have been useful for developing computer graphics shortcuts. Frameless Rendering can use this knowledge to substitute the randomness of the choice of pixel update with perceptually-driven updating. This will assist in reducing the visible artifacts associated with Frameless

Rendering.

Motion Smear: The Human Visual System at work

Motion blur is useful for performing temporal anti-aliasing and for imitating the behavior of a camera shutter. Another compelling argument for employing motion blur is to copy the behavior of the human eye. An open shutter with a duration of *120-125ms* better approximates the motion-induced motion smear appearing on the retina, than frames where all pixels have been computed from the exact same time step.

Exploiting Motion Smear: Visual information is accumulated over an interval of approximately 125 *ms* [2]. The retinal image, an integration of this information, is blurred or smeared. The newest information has the greatest photoreceptor response. An appropriately weighted integral can model this integration over time. A camera shutter integrates evenly over time. In `rear curtain' photography, the use of a flash at the end of the exposure interval allows for emphasis of last-in information. It is a step closer towards emulation of the eye's `open shutter' action.

Figure 3 shows images created by leaving the camera shutter open for a duration of 125 *ms*. Notice in the highway images how the vehicles in the distance are not blurred. The analogous condition exists on the retina. Although the distant cars are moving at the same velocity, their velocity on the final projection plane, is relatively low, and produces little or no blur.

Figure 3

Frameless Rendering can use a model of retinal information decay to model environments to match the final retinal activity. This will ameliorate the visible artifacts associated with Frameless Rendering. But studies have shown that blurring based on image velocity is often unable to fool the eye.

Studies show that humans are sensitive to spatial blur with image velocities up to at least 9.4 degrees/second [5]. How then is the motion picture and television industry able to successfully substitute blurred frames without introducing visible image degradation? The conflict is resolved by understanding that the eye acts as a filter [4]. Because the eye tracks motion, the final retinal velocity may be lower than the original image velocity.

An important observation is that if the eye is moving, an object that otherwise had zero velocity will now have a positive magnitude of velocity. So, there will still be blur on the retina, but ironically it will be due to the stationary object. For example, consider a scene with a single moving object in the foreground against a non-homogeneous (in color) background. In this case, there will always be retinal smear. If the eye doesn't track, the motion smear is due to the image plane velocity. If the eye tracks precisely, then it is still the magnitude of the image plane velocity that determines the motion smear.

Because of the eye's part in filtering an image, there are many cases where an artificially motion-blurred sequence will introduce detectable blur, thereby decreasing the overall image quality. The reason that blurred motion picture in-between frames do not result in decreased image quality is twofold. First and foremost, animators examine and modify sequences over and over again until they `look' right. The image degradation may have been visible in early attempts. Secondly, the technique is likely employed only when one or more of the following non-orthogonal conditions hold:

- Low predictability of motion: If the observer does not know what to expect, then he or she cannot easily track the motion.
- Motion acceleration: Object acceleration is correlated with an inability to eye-track at the same rate as the motion [4]. This is because a changing velocity is difficult to predict. So, it is a sub-class of the category above. If a single object maintains a constant acceleration, though, it is likely that the eye can closely

mimic this motion.
- Complex, untrackable scenes: Consider a sequence with accelerated camera motion and many objects contributing sudden and unpredictable motion to the scene. The eye is not capable of doing effective tracking on the whole of the action. As a result, the eye as a filter does very little to change the original image plane velocities before projection onto the retina.

In light of the above information, I believe that if the image presented to the eye matches the expected retinal image, no blurring will be detectable. The goal of a good Frameless Rendering implementation is to consider the final retinal velocity and use this in importance sampling in areas of lowest retinal blur.

How can we determine what the final retinal image will be? Eye-tracking coupled with prediction can make an effective determination of how fast the eye is moving. It can be further assisted by *a priori* knowledge of how the human eye works in a variety of conditions. Simple cases can be addressed first:

- only the camera is moving
- a single object is moving
- two objects are moving with known trajectories

More complex cases such as simultaneous camera and object motion, and many objects in motion concurrently, require extensive future study.

Without actually computing the final retinal velocity, importance sampling based on image velocity can be used when the conditions described above exist: low predictability of motion, objects undergoing acceleration, and/or complex, untrackable scene motion.

Finally, it is important to throw a wrench into all of this theory. The perceived motion is not always a straightforward matter. There are documented cases of induced motion, although no actual motion is presented to the observer [**9**]. An example of apparent motion in the absence of actual motion, is the `waterfall' effect. If a person stares at a waterfall for a long period of time, and then looks away, they will perceive motion of equal velocity and acceleration in the opposite direction. This will occur in similar situations where there is constant flow in one direction and an observer is fixed on the motion for some time. Another effect occurs when we see a train moving alongside us when actually it is the train we are on that is moving. These effects are well known, and they are intimately connected to eye-tracking. Of course the most obvious case is the inferred motion from a sequence of static frames. This case is so well accepted as a feature of the human visual system that throughout the paper it is assumed to be `actual' motion.

Conclusion

Motion blur is a useful computer graphics technique. It attacks temporal aliasing and simulates the action of a camera's shutter. It has not been explicitly used for simulating the human visual system's integrating characteristic. This motion smear phenomenon means that at any instant in time, the image on the retinal is a weighted integral of the previous 120-125 *ms* of information. Motion blur is effective in smoothing motion, not just because of its increased effective sampling rate, but because it integrates information over time. At any instant in time, the motion-blurred image the observer sees contains information from an interval of time steps.

There seemed to be conflicting data concerning the effectiveness of introducing blur before the eye receives the information. The motion picture industry presents blurred images to human observers and high quality sequences are perceived. Some computer-generated, motion-blurred sequences also effect a high quality perception. In fact, the quality is enhanced because of reducing temporal aliasing artifacts. But Girod's work [**5**], and other corroborated studies, have found that human observers can detect artificial blur under many conditions.

The images we see on television successfully exploit the HVS motion smear quality because the sequences contain high image acceleration, low predictability of motion, and complex scene motion. These qualities contribute to the eye's inability to track effectively, and thus the retinal velocity matches the original image velocity. Such image sequences exploit the retinal smear phenomenon rather than image space integration.

An improved Frameless Rendering implementation will consider the final retinal velocity and use this to importance sampling in areas of lowest retinal blur.

There at least two methods we can currently avail ourselves of in computing the final retinal image. We can exactly compute or predict the final retinal velocity. This can be accomplished by using eye-tracking data. In the presence of motion, there will invariably be regions of relative importance to the eye. This is because a moving eye will result in all stationary information being smeared on the retina. A less expensive, albeit also less precise, estimation of final retinal velocity comes from understanding the human eye's inability to track unpredictable motions and motion in opposing directions.

Acknowledgements

I would like to thank the following people: Dr. Gary Bishop, Dr. James Coggins, Dan Crawford, Marc Olano, and Andrei State.

I would also like to thank the Link Foundation Fellowship in Advanced Simulation and Training and the NSF/Darpa Science and Technology Center (STC) for Computer Graphics and Scientific Visualization, NSF Cooperative Agreement #ASC-8920219.

Footnotes

...quality

Motion Smear: A description of the integrated information on the retina occurring over a 125 *ms* temporal window.

...sampling

Importance Sampling: The computer graphics world is a discrete world, although the real world is continuous. Some form of sampling is inherent in discretizing a continuous representation. Importance sampling refers to increasing sampling for important areas, and decreasing sampling for unimportant areas. There are as many measures of importance as there exist importance sampling techniques.

...snapshots

Snapshot: The static image in display at any instant in time. Note: At an instant in time, the static image in a raster CRT contains only a few scanlines of illuminated phosphors. When we refer to snapshot we mean an entire display grid equal to the full image resolution.

...time

Exposure Time: Open camera shutter duration. The film acts as an integrating medium to accumulate the total radiant energy of the objects in the scene.

...cells

Inbetweening: In cell animation, boundary cells indicating the overall action in a sequence are drawn by high level designers, and the cells representing the motion in between these high level cells are drawn by in-betweeners. The creation of these cells is known as in-betweening.

References

1

Gary Bishop, Henry Fuchs, Leonard McMillan, and Ellen J. Scher Zagier. Frameless rendering: Double buffering considered harmful. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 175-176, July 1994.

2

David Burr. Visual processing of motion. In *Trends in Neuro Sciences*, volume 9, No. 7, July 1986.

3

Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 137-45, July 1984. Monte Carlo distribution of rays to get gloss, translucency, penumbras, depth of field, motion blur.

4

Michael P. Eckert and Gershon Buchsbaum. The significance of eye movements and image acceleration for coding. In A.B. Watson, editor, ***Digital Images and Human Vision***, pages 90-97, 1993.

5

Bernd Girod. Eye movements and coding of video sequences. In T. R. Hsing, editor, ***SPIE Visual Communications and Image Processing***, pages 398-405, 1988.

6

Jonathan D. Korein and Norman I. Badler. Temporal anti-aliasing in computer generated animation. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, volume 17, pages 377-388, July 1983.

7

M. Potmesil and I. Chakravarty. Modelling motion blur in computer-generated images. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, volume 17, pages 389-399, July 1983.

8

Ellen J. Scher Zagier. Frameless antialiasing. Technical Report UNC-CS-TR-95-026, Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

9

Robert Sekuler and Randolph Blake. ***Perception***. McGraw-Hill Publishing Company, New York, 1990.

Author Bio

Ellen J. Scher Zagier can be reached at:
Department of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3175.
Email: scher@cs.unc.edu