Real-time Frame Rate Up-conversion for Video Games

or how to get from 30 to 60 fps for "free"

Dmitry Andreev
LucasArts

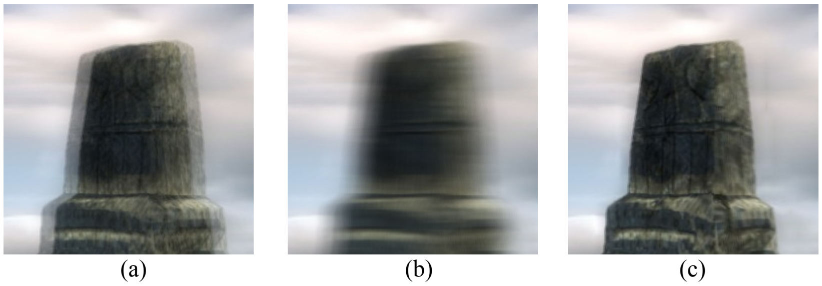

Figure 1 Observations of eye tracking object movement on a 60 Hz display.
(a) 30 fps rendering, (b) 30 fps rendering with motion blur, (c) our method.

Abstract

Currently in the video game industry, the holy grail of rendering is achieving ultra realistic visual quality while maintaining a consistent 60 frames per second. However, the cost of doing so means having half the time to construct an image when compared to simply running at 30 fps. Most 30 fps games roughly split up their rendering time in half, standard scene and alpha rendering, and all the other glamorous post processing effects and shadows. This inevitably makes it extremely difficult for game studios to achieve high quality visual images with post processing and dynamic shadows while maintaining a consistent 60 frames per second.

Video games that run at 30 fps suffer from very noticeable motion "flickering" artifacts due to fast movements of objects and/or the camera. One common technique to help remedy this flickering artifact is to introduce motion blur. In one way this technique helps smooth out the perceived motion of objects, but is only able to compensate properly relative to the virtual camera. In other words, the game does not know where the user's eyes are focused to and tracking, so in practice in interactive applications, the images tend to be more blurry all the time even in areas where the user's eye is tracking an object's motion and expecting it to be sharp. Running an application at 60 fps on a 60 hertz display eliminates multiple expositions of a single frame to the eye, reducing "flickering" artifacts and increasing sharpness of the moving objects under the eye tracking conditions. Motion blur only needs to be added to compensate for motions in the image that move faster then the eye can track. Figure 1 shows the difference in motion from a 30 fps rendering image, 30 fps with motion blur, and the 60 fps respectively on a 60 hertz display.

Film typically has an easier work around to this problem because of the nature of the entertainment medium not being user interactive. The presentation of shots, usage of depth of field, and limiting the use of fast motions around the focal point are common techniques that are used because the user's eye can be guided to focus on certain parts of the image. Each shot is tuned specifically assuming the user's eye is focusing on certain material, thus avoiding the problem completely.

The proposed solution is a novel technique that combines the best of both high quality rendering at 30 fps with the natural motion of objects refreshing at 60 fps, introducing very minimal overhead in terms of memory and performance. The basic concept is to approximate the middle frame between what has previously rendered and what is currently being constructed and present it as the new "predicted" image exactly in the middle of rendering at a 30 fps rate, thus empowering a product to still "feel" as if it is refreshing at 60. Televisions use a similar trick to achieve refresh rates near 120 hertz. However, for video games, more information is present regarding the frames construction, such as depth and velocity, and creating the predicted frame can be significantly simplified. This proposed technique is useful for all real-time user interactive applications to help guarantee that a very high quality of rendering is achieved, by allowing more time to construct a frame, but still refresh at a higher rate such as 60 frames per second.

Additionally, further techniques are considered such as rendering more static / slow moving parts of the scene at an even lower refresh rate, such as 15 fps, because some tests have shown that the predicted images are "good enough" approximations of the slower moving data from the user's perspective. This would thereby increase the time an application would have to construct a frame, and still maintain 60 frames per second. Also, always maintaining 60 fps by automatically presenting more predicted frames during the instances that the game slows down. This would additionally balance visual quality of fast moving objects with refresh rate consistency.

CR Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation.

Keywords: Frame rate increase, up-conversion, image interpolation, velocity rendering, display synchronization, interactive

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007