The ImageSwitcher: A Proposed System Architecture Designed to Reduce VR Lag

David C. Banks
Department of Computer Science
Mississippi State University, USA

Abstract

Latency contributes to image error and to motion sickness in head-tracked graphics displays. We attack the problem of latency by exploiting parallel scene generation, using multiple graphics engines to render images corresponding to a cloud of viewpoints. The real-time position of a 3D tracker is used to select among several just-generated images, rather than to generate a new image, for each frame. The system's scalable architecture is comprised of off-the-shelf components. When future PC-class machines offer inexpensive high-speed 3D graphics, the design of the system will become economically attractive.

1 Introduction

The design goal for a synthetic, immersive 3D environment is to present to the human viewer a believable virtual world. There are numerous obstacles to achieving this goal; this paper addresses the problem of system lag. Because of this latency, an image corresponds to where the human viewer *was* looking, not where the viewer *is* looking. Latency in a typical VR system is attributable to many sources. There is latency associated with reading the position of a 3D tracker, with converting the reading into a transformation that matches the viewpoint, with applying the transformation to the scene's geometry and rendering the geometric primitives, with synchronizing the frame-buffer's image to the video display, and with scanning the video image out to the display. In a see-through head-mounted display (STHMD), the synthetic image of the virtual world should register perfectly onto the image, passing through the semi-transparent display device, of the real world. In practice the two are mis-aligned even when the viewer's head is stationary, but especially when the viewer moves. System latency is also a culprit in the vertigo or nausea some users experience with non-see-through head-mounted displays.

Reducing lag is of paramount importance in developing immersive environments [Holloway95]. Yet market forces are not necessarily aligned with the goal of reducing system latency. Each year graphics engines become cheaper and their throughput becomes greater; after all, there is always a market for generating more complex scenes in a unit time (although that unit of time may be on the order of minutes for a frame in a studio animation). Color displays become smaller and brighter: the preview screen on hand-held video cameras makes this a commodity item. The range of 3D trackers increases: it is desirable to have a large working envelope in which to engage a virtual world. Noticeably absent from the list of market factors is an influential market force that rewards reductions in latency by factors as dramatic as those enjoyed by improvements in price, throughput, size, and range.

It is easy to imagine that in the near-term future personal computers (or even hand-held interactive games) will deliver the same graphics performance as today's mid-range graphics stations. The cost will be hundreds, not thousands, of dollars, but a single image will still require the same amount of time to work its way through the graphics pipeline. Meanwhile the midrange graphics machines will offer phenomenal throughput, but unless they undergo a radical re-design the they will exhibit latencies that are essentially the same as machines of today, to the frustration of those who develop immersive systems.

This paper describes a novel graphics system architecture that is currently being investigated. The system, called the ImageSwitcher, can be considered as a design point at an extreme edge of the space of VR system architectures. It assumes that latency due to image generation is the foremost concern that demands to be addressed. Our proposed solution is to marshall parallel image generators, any of whose results can be delivered to a display device at a moment's notice, to provide a pool of images from which the appropriate one may be selected based on the latest report from a 3D tracking system. Software emulations of the system are under development at the present time. We anticipate that a prototype system will be constructed in the fall of 1997.

The paper is organized as follows. Section 2 presents an overview of the system architecture for the ImageSwitcher. Section 3 describes additional opportunities for high-quality image generation that such a system allows. Section 4 puts the ImageSwitcher in the context of previous and related work.

2 The ImageSwitcher

One strategy for reducing system lag in an immersive environment is to redesign some of the system's individual hardware components. The potential payoff is high, and new technology can emerge as a result. But unless there is a pathway to transfer the new technology to industry, the resulting design can languish in the lab. A different strategy (which the ImageSwitcher adopts) is to design a system architecture using off-the-shelf components. If it is successful, the design can be readily duplicated at other sites.

We propose to assemble off-the-shelf components into a graphics system called the ImageSwitcher that trades higher parallelism for lower latency. By rendering an image *after* a viewpoint is determined, the typical VR system introduces a significant amount of latency due to image-generation and video synchronization. The ImageSwitcher will instead generate, in parallel, multiple images corresponding to a cloud of viewpoints in the neighborhood of the predicted one. These images are fed into a video switcher. Then, when the current viewpoint is determined, the system selects the best image to pass through the switcher and on to the display. By inserting a dedicated computing node to negotiate between the tracking device, the graphics engines, and the video switcher, the system will introduce its own negligible

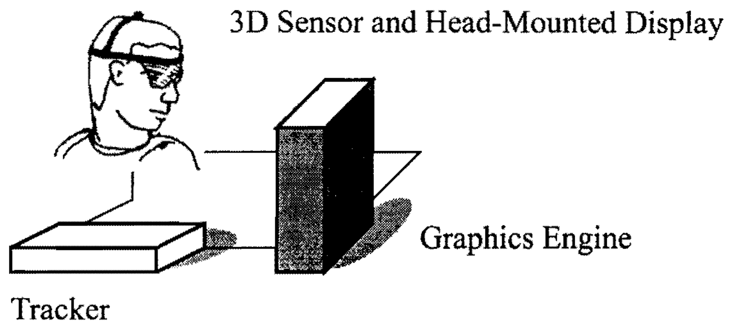

Figure 1. A simple architecture for a VR system.

latency in exchange for eliminating more significant latencies due to image-generation and video synchronization.

The most expensive components of the ImageSwitcher are its multiple graphics engines providing parallel image-generation. The design we propose is expensive at present, but it anticipates the advent of low-cost, high-performance graphics machines (in the form of personal computers or set-top boxes) that, when the term of the project is complete, can be substituted for their high-cost predecessors. As an example, the recently-announced "Nintendo 64" promises real-time anti-aliasing, texture mapping, depth-buffering, and a steady frame-rate for an expected list price of $250. The ImageSwitcher's "glue" – everything except the graphics engines – can be replicated at a modest cost at labs already possessing multiple graphics engines in order to harness their rendering performance to produce similar low-latency systems.

A simple architecture for a VR system includes a 3D tracker, a graphics engine, and a display (either attached to the head or fixed in 3D space), as illustrated in figure 1. The head's position is reported by a tracker, the position and orientation are reported to the graphics engine, the graphics engine renders the scene (either in mono or stereo), and the image is scanned out for display. Each step of the process introduces its own amount of latency.

The ImageSwitcher design uses multiple graphics engines, working in tandem, to reduce the overall latency of the VR system. Its architecture is illustrated in figure 2. The goal of the ImageSwitcher is to have the correct image available on demand as soon as the tracker's position has been reported (and the user's viewpoint thereby determined). Each graphics engine G_k renders an image I_k for the tuple V_k of view-parameters. This n-tuple defines the viewpoint and the image-rectangle within the image-plane. The set V_k is centered, in the space of tuples, at $\underline{V}$, which is the predicted value of the view-tuple for the next frame to be displayed. When the user's actual view-tuple V is determined, the video switcher is used to select the closest image and pass it along to the display device. The latency introduced by the switcher is negligible. Even in the worst case, the image delievered to the user should be no more jittery than in a conventional system: the head has had less time to move before an image is displayed, and a larger set of images is available near the user's viewpoint.

A dedicated master node (such as a personal computer) serves as the bridge between the tracker, the graphics engines, and the switcher. It is essential that the latency produced by this node be kept as small as possible. The task of viewpoint-prediction is performed by the graphics engines. The master node maintains a table of view-tuples, one per graphics engine. The graphics engines predict how a suitable cloud of viewpoints should be distributed. Each engine negotiates for a point within the cloud and informs the master of its location when the image is generated. At each video frame the master node selects the image corresponding to the viewpoint nearest to the user's and signals the switcher to pass the corresponding video input to the display. The video outputs of the graphics engines are synchronized against a common synch signal. The amount of communication required at each frame is small: the images are routed through video cables only.

An obvious concern arises about number of samples in the cloud of viewpoints. After all, the viewpoint's position and orientation inhabit a 6-dimensional space. With a meager 3 samples in each direction, there would be $3^6 = 729$ graphics engines in the system! But as the architects of the address recalculation pipeline [Regan94] have observed, viewpoint-orientation can be corrected as a post-process: pann and scroll are close approximations to pitch and yaw (for objects far from the center of rotation), eliminating two degrees of freedom. The error due to roll is expected to be small - the human neck opposes rapid roll. Neglecting the orientation due to roll reduces the dimension of the sample-space of viewpoints down to just the 3D position of the user. The most acute registration errors in a VR system are in the xy-plane of the screen, thus the viewpoints parallel to this plane demand the most sampling. This observation suggests that a flat ellipsoid will enclose the distribution of viewpoints assigned to the graphics engines. The samples can thus be distributed in an elliptical pattern, where the major axis is aligned with the maximum error vector of the user's viewpoint (projected on the image plane) and is proportional in length to the magnitude of the error. The minor (perpendicular) axis likewise has length proportional to the error in the perpendicular direction, projected on the image plane. Figure 3 illustrates a possible distribution of viewpoints.

3 Algorithm Opportunities

The ImageSwitcher is designed to reduce lag due to image generation in a VR system. But the array of graphics engines can be exploited to produce other beneficial effects. Chief among them are motion blur, focus, and adaptive refinement.

Motion Blur

With a video mixer, viewpoints along the trajectory of the user's head motion can contribute their images to the final display, using a video mixer rather than a single-frame video switcher. The resulting motion blur can add realism to the scene and reduce aliasing due to undersampled images between consecutive positions along the path of the viewpoint.

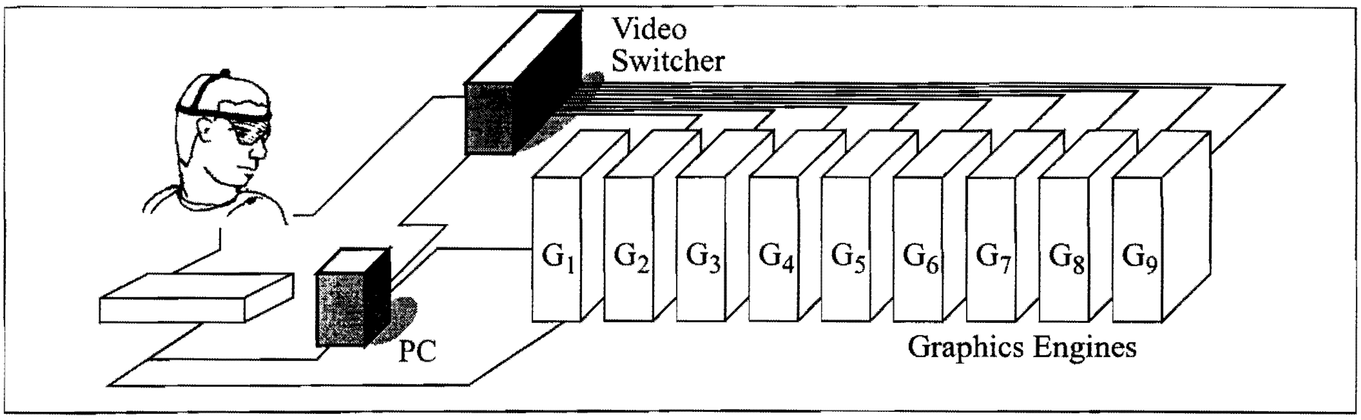

Figure 2. The system architecture for the ImageSwitcher.

Focus

The ImageSwitcher, equipped with a video mixer, can also produce images that use focus as a depth cue by perturbing the viewpoint along the view direction. The resulting images, when blended together, produce a depth-of-field effect in the same manner as a camera with a large aperture. This scheme is essentially a parallel implementation of the sequential technique for producing depth-of-field with an accumulation buffer [Akeley90].

Adaptive Refinement

Even if the user's head is stationary, a head-mounted tracker will continue to report slight movement due to static error in the tracker. The error distribution, however, will be tightly concentrated compared to the distribution that results from a moving tracker. As a result, the samples V_k become packed closely together in space and fewer pre-computed images are required in order to produce a good match with the user's viewpoint. In such a case, some of the graphics engines can be taken "off-line" to produce higher-quality images asynchronously at a lower frame rate. While the tracker's reported position remains in the same small volume, the individual image samples can be improved and then made available to the video switcher for display. This technique overcomes a deficiency in conventional VR systems, where progressive refinement is prohibited by the constant jitter in the reported position of a stationary user.
An undesirable side-effect of this technique is that it introduces aliasing in the space of image-quality: whenever the displayed image changes from one quality-level to another, the result is visible to the user as a sudden change in what should be nearly a static image. A simple remedy to this problem is to use a video mixer rather than a switcher, so that two or more images are blended together to produce the final display.

4 Previous and Related Work

This section describes related work and puts the ImageSwitcher in the context of that work. Some of the features of the system are inspired by or are related to previous work: image composition, multidimensional sampling, viewpoint prediction, latency reduction, and image-warping.

Image Composition

The process of image-composition is familiar to animators. Components of an image are generated separately and blended together to produce the final result [Porter84]. It is not uncommon for sub-images to be blended together in graphics or video systems, perhaps by consulting a matte (or stencil) to select the region to be composited. The ImageSwitcher, using a video mixer, applies this technique in a very simple way by treating the entire image area as the matte. The PixelFlow machine is somewhat similar: it uses multiple rendering nodes to generate partial images which are then assembled to produce a final picture [Molnar92]. The Pixel-Flow machine uses a sort-last architecture [Molnar94] to defer a portion of the visibility calculation until the individual images are created. Each image contains only a subset of the scene's total geometry. Two images are composited by consulting the depth pixel-by-pixel. The state vector of the pixel with the nearer depth is retained; the other state is discarded. Two of the great benefits of this architecture are that (1) for large datasets, Pixel-Flow's performance scales linearly with the number of renderers, and (2) the latency generated by the composition network is small.

The design of the ImageSwitcher is inspired somewhat by that of Pixel-Flow. But instead of using custom logic to select colors from partial images on a per-pixel basis, the ImageSwitcher uses the video switcher/mixer to select/assemble an entire, complete image. As a consequence the scene geometry is replicated on each graphics node of the ImageSwitcher (which limits the potential scene-complexity) but the custom compositing hardware is replaced by a cheap, available commodity product.

Multidimensional Sampling

In ordinary ray-tracing [Whitted80], a ray from the viewpoint to a pixel is intersected with the geometry in a scene to determine the visibility and the shading at the pixel. In order to reduce aliasing, a distribution of rays may be used to sample the geometry, the viewpoint, and the light sources. [Cook84, Kajiya86, Mitchell91]. If the geometry is moving, the distribution of samples must include both space and time. To produce an image that accounts for depth-of-field, a distribution of pinhole-positions (using a pinhole-camera metaphor) must be used as well. Area light-sources are multi-sampled in order to produce soft shadows. Ray-traced images that account for all these samples are necessarily slow by comparison to a simple polygon renderer because they incur an enormous expense at each pixel containing geometry. But these

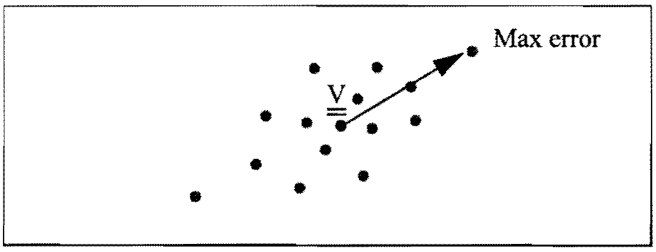

Figure 3. The distribution of viewpoints V_k.

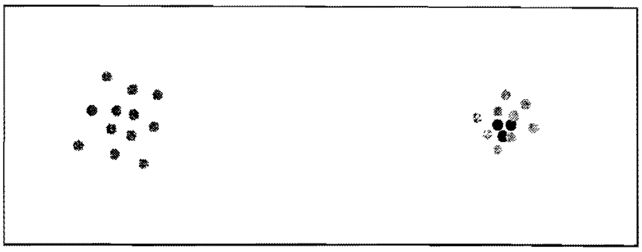

Figure 4. The distribution of viewpoints V_k for a jittery tracker (left) and for a nearly-stationary tracker (middle) Some of the images from closely-spaced viewpoints can be re-assigned (black dots at right) in order to produce higher-quality images at a lower frame rate.

images provide a capacity for image quality that is unmatched by other image-generation techniques, in part because the distribution of samples can be made sufficiently dense to reduce aliasing to an acceptable level.

The ImageSwitcher applies the notion of multidimensional sampling to an entire image rather than to a single pixel. That is, an image can be considered as a single point in an abstract space of images, each of which is represented by a set of view-parameters. An image and a point within view-parameter (VP) space are equivalent. One can elect to display only the point in VP-space nearest to that of the user. Such a scheme is similar to the visibility calculation for opaque geometric primitives along a ray, where only the closest primitive is displayed at a pixel. The chief difference is that the distances along a ray lie in a totally-ordered 1-dimensional space where the minimum distance is well-defined. Since VP-space is multi-dimensional, a metric must be used to assign distances to each of the samples. If several candidate points in VP-space (or equivalently, images in the set-of-images) are equally near the desired one, it is reasonable to average them. In this regard, blending images together is akin to de-aliasing pixel-samples by convolving them with a filter function. The accumulation buffer [Akeley90, Mammen89] adopts such a strategy, although it filters images sequentially rather than in parallel.

The expense of evaluating multiple samples per pixel may have seemed unreasonably large when it was first proposed, but it has become standard practice because the alternative (aliasing) is so undesirable. Mitchell, for example, used more than a thousand samples for each pixel in his images. The very notion of ray-tracing itself even seems wasteful: only one intersection-test between object and ray yields a value that contributes to the image (for opaque objects). The ImageSwitcher is similar: it may select only one image out of many for final display to the user. The expense of generating all the "wasted" images may seem unreasonably large, but it is a powerful weapon for reducing latency due to image-generation. It is certainly difficult to imagine an ImageSwitcher with thousands of graphics engines being used to create a final picture. On the other hand, Akeley used only 23 samples to produce effective images with the accumulation buffer; this number seems like a more realistic target, and it is encouraging to see previous success using a number of image-samples in this range.

Viewpoint Prediction

If the user's viewpoint could be accurately predicted, lag would no longer contribute error to the image that the graphics system displays. Several efforts have been undertaken to predict the viewpoint as accurately as possible [Azuma94, Azuma95, Emura94, List94, So92]. The prediction may be based on position, velocity, and acceleration of the user's viewpoint, and in general the prediction improves the accuracy of the displayed image. Yet Azuma and Bishop report that

> at predicted distances of ~100 ms [milliseconds] or more, the jitter in the predicted outputs often reaches objectionable levels. ... Thus, prediction cannot compensate for arbitrary amounts of latency; to be effective, it must be combined with efforts that minimize system lag.

Even on a graphics supercomputer (Pixel-Planes 5), latency due to image-generation is reported to be between 54 and 57 ms for minimal datasets [Olano95], contributing more than half of the amount deemed unacceptable. The ImageSwitcher removes this latency due to image generation by pre-generating a distribution of images. Moreover, by reducing the overall system latency the ImageSwitcher improves the error-behavior of any prediction scheme, which in turn reduces the number of images that must be pre-generated.

Latency Reduction

In addition to the latency incurred during the rendering of an image, a VR system must endure the latency that results from synchronization of the image-generation with the video-refresh. While the video controller is scanning out the contents of a frame buffer, the typical graphics system renders a new image into a second frame buffer. When rendering is completed, the buffers swap. A fundamentally different approach, called frameless rendering, permits the graphics system to render into the same buffer that is being displayed [Bishop94]. In frameless rendering the image is only partially rendered at the time of scan-out. To spread the error uniformly over the image, a frameless renderer only fills randomly-assigned pixels within each primitive. The partially-updated image has lower latency at the expense of being undersampled.

This scheme is similar to the ImageSwitcher in that both are designed to reduce latency. The frameless renderer incurs less than one frame of latency by rendering scattered pixels during a frame-interval. The ImageSwitcher incurs zero frames of latency, by pre-rendering images before they are demanded. Both the frameless renderer and the ImageSwitcher offer the possibility of progressive refinement, which improves the image-quality whenever the viewpoint remains stationary.

Image Warping and Addressing

An image can be treated as a geometric primitive in its own right. If each pixel is equipped with depth information for the primitive it

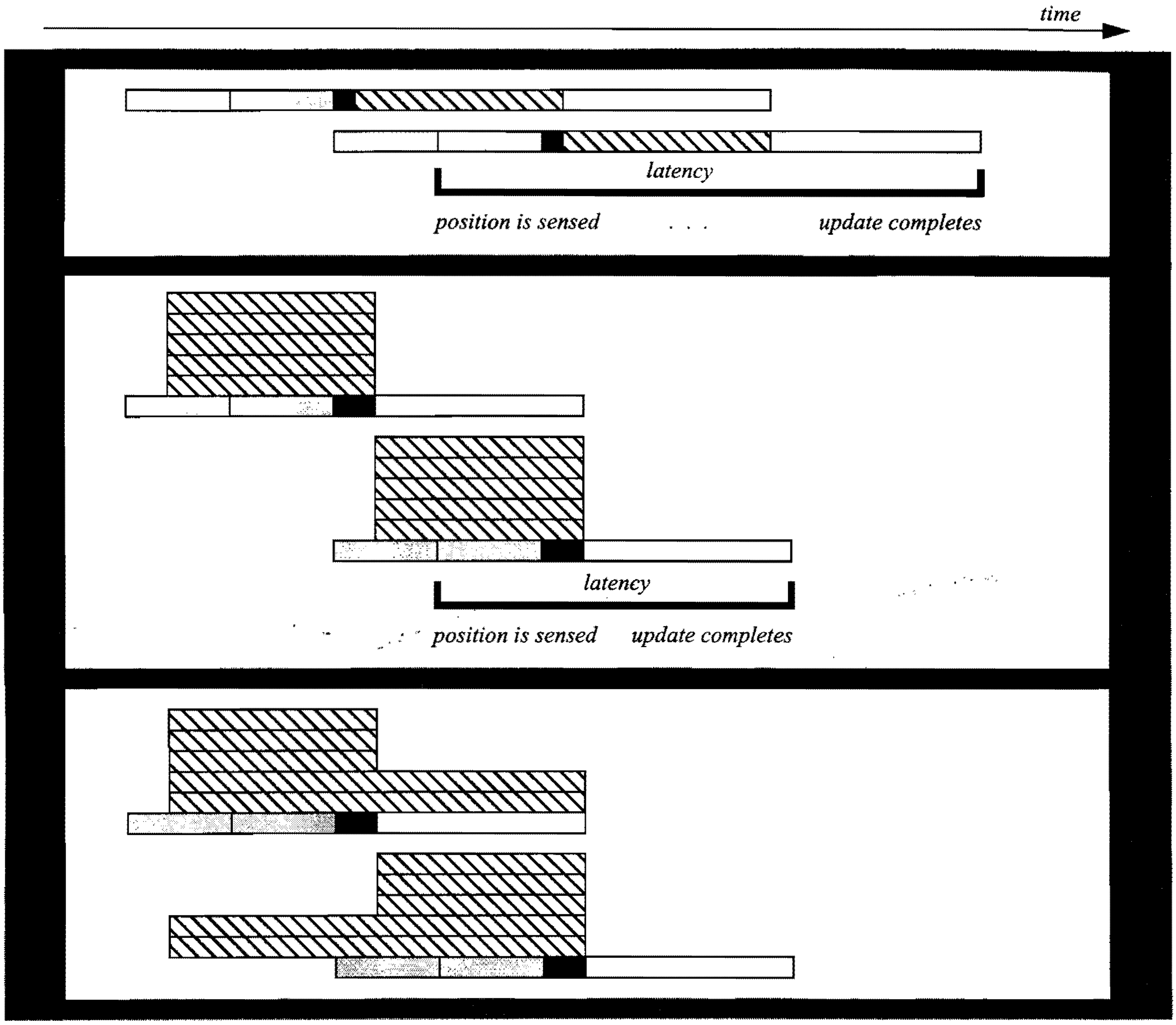

Figure 5. Latency in a conventional double-buffered VR system (top) includes lag due to the 3D tracker, due to viewpoint determination, due to image generation, and due to updating the display. While these numbers are device-dependent, 9ms, 3ms, 17ms, and 17ms (respectively) are not unreasonable measures for high-performance hardware. The "extra" tracker report might be incorporated into a viewpoint prediction, which is part of viewpoint determination. In the ImageSwitcher (middle), parallel image generation is performed while the viewer's position is sensed and while the viewpoint is determined. Displaying the best-match image results in lower latency as long as the If the tracker is nearly stationary (bottom), some images may be generated at a higher level of quality and used later if they are still near the viewpoint.

displays, that 1-pixel-large primitive has 3-dimensional (x, y, z) coordinates. Pointlike primitives can be captured by a range-scanner [Besl88] or extracted from a rendered image that is augmented with a depth map. A less obvious way to extract range information is to use different viewpoints to generate an image. If each pixel-sized primitive in the first image has a corresponding pixel-sized primitive in the second image, the vector between their locations establishes the optical flow at the pixel. The optical flow of an entire image can be used to interpolate between images [Chen93, Chen95, McMillan95], or even to extrapolate an image one frame into the future.

This technique is very promising as a way to reduce registration error due to latency. After an image is generated, it can be warped according to the optical flow deduced by reading the most recent location of the user's 3D tracker and consulting the pixels' positions and depths. The warping can be done prior to scanning out the frame to the display. The technique suffers from the drawback that the final image may have "cracks" where no corresponding pixel exists in the pre-warped image. This artifact is particularly acute when the viewpoint translates so that a nearby object no longer occludes a distant object. In order to correctly display the extrapolated image, the system would have to re-render the distant object (which was occluded in the pre-warped image).

Another image-based approach opts to render the entire 360-degree projection of the image onto the faces of a cube surrounding the center of projection. A portion of this total image is transferred to the display device based on the user's view-direction using an address recalculation pipeline [Regan94]. The system has minimal latency when the position of the viewpoint remains fixed: rotating the view-direction becomes equivalent to selecting a different subwindow of the set of rendered images. By combining image composition (in the style of PixelFlow) with the address recalculation pipeline, a system can even render different portions of the scene at different rates and merge them together for the final display. Separate images of the geometric objects can be warped before composition in order to produce the illusion that the objects have moved (or that the viewpoint has moved with respect to the objects).

Light Field Rendering

If several images have been computed for a set of viewpoints, a new viewpoint can be chosen (and its image reconstructed) by considering the light field [Levoy] sampled by the original viewpoints. If the original viewpoints are dense enough or close enough to the desired one, the reconstructed image suffers from little error when linear interpolation is applied. This linear interpolation, however, introduces its own latency: four variables are being interpolated. Even so, it shares the notion with the ImageSwitcher that multiple views can be exploited to reconstruct a desired one. If this per-pixel interpolation can be performed sufficiently fast, this approach could become an attractive alternative to video switching.

5 Conclusions

This paper proposes a novel architecture for a virtual-reality system called the ImageSwitcher whose image-generation lag is significantly reduced compared to conventional VR systems. The scheme uses multiple graphics engines to precompute images. The final image is selected or merged by video hardware at negligible latency. The impact of the project can be substantial, because VR systems demand low latency and because image-generation contributes significantly to system latency. In the near-term future, graphics engines are likely to drop dramatically in their price. The ImageSwitcher provides a mechanism for exploiting the low-cost machines in the future, and its design can be replicated at other sites. Software simulations of the system are currently being developed. It is anticipated that the first prototype will be tested in 1997.

Bibliography

Akeley90 Kurt Akeley, "The Accumulation Buffer: Hardware Support for High-Quality Rendering," Proceedings of ACM SIGGRAPH 90, pp. 309-318.

Azuma94 Ronald Azuma and Gary Bishop, "Improving Static and Dynamic Registration in an Optical See-Through HMD." Proceedings of ACM SIGGRAPH 94, pp. 197-204.

Azuma95 Ronald Azuma and Gary Bishop, "A Frequency-Domain Analysis of Head-Motion Prediction," Proceedings of ACM SIGGRAPH 95, pp. 401-408.

Besl88 P. J. Besl, "Active Optical Range Imaging Sensors," Machine Vision and Applications, Vol. 1, 1988, pp. 127-152.

Bishop94 Gary Bishop, Henry Fuchs, Leonard McMillan, and Ellen Scher Zagier, "Frameless Rendering: Double Buffering Considered Harmful," Proceedings of ACM SIGGRAPH 94, pp. 175-176.

Chen93 Shenchang Eric Chen and Lance Williams, "View Interpolation for Image Synthesis," Proceedings of ACM SIGGRAPH 93, pp. 279-288.

Chen95 Shenchang Eric Chen, "QuickTime VR - An Image-based Approach to Virtual Environment Navigation," Proceedings of ACM SIGGRAPH 95, pp. 29-38.

Cook84 R. L. Cook, T. Porter, L. Carpenter, "Distributed Ray Tracing." Proceedings of ACM SIGGRAPH 84, pp. 137-145.

Emura94 Satoru Emura and Susumu Tachi, "Compensation of Time Lag Between Actual and Virtual Spaces by Multi-Sensor Integration." Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 463-469.

Holloway95 Richard Holloway, Registration Errors in Augmented Reality Systems, PhD Dissertation, Department of Computer Science, UNC-Chapel Hill, 1995.

Kajiya86 James Kajiya, "The Rendering Equation," Proceedings of ACM SIGGRAPH 86, pp. 143-150.

Levoy96 Marc Levoy and Pat Hanrahan, "Light Field Rendering," Proceedings of ACM SIGGRAPH 96, pp. 31-42.

List94 Uwe List, "Nonlinear prediction of Head Movements for Helmet-Mounted Displays." Technical report AFHRL-TP-83-45 [AD-A136590], Williams AFB, AZ: Operations Training Division, 1994.

Mammen89 Abraham Mammen, "Transparency and Antialiasing Algorithms Implemented with the Virtual Pixel Maps Technique," IEEE Computer Graphics and Applications, July 1989.

McMillan95 Leonard McMillan and Gary Bishop, "Plenoptic Modeling: An Image-Based Rendering System," Proceedings of ACM SIGGRAPH 95, pp. 39-46.

Mitchell91 Don Mitchell, "Spectrally Optimal Sampling for Distribution Ray Tracing," Proceedings of ACM SIGGRAPH 91, pp. 157-164.

Molnar92 Steven Molnar, John Eyles, and John Poulton, "PixelFlow: High-Speed Rendering Using Image Composition." Proceedings of SIGGRAPH 92, pp. 231-240.

Molnar94 Steven Molnar, Michael Cox, David Ellsworth, and Henry Fuchs, "A Sorting Classification of Parallel Rendering," IEEE Computer Graphics and Applications, Vol. 14, No. 4 (Jul 1994), pp. 23-32.

Olano95 Marc Olano, Jon Cohen, Mark Mine, and Gary Bishop, "Combatting Rendering Latency," 1995 Symposium on Interactive 3D Graphics, pp. 19-24.

Porter84 Thomas Porter and Tom Duff, "Compositing Digital Images," Proceedings of ACM SIGGRAPH 84, pp. 253-259.

Regan94 Matthew Regan and Ronald Pose, "Priority Rendering with a Virtual Reality Address Recalculation Pipeline," Proceedings of ACM SIGGRAPH 94, pp. 155-162.

So92 Richard So and Michael Griffen, "Compensating Lags in Head-Coupled Displays Using Head Position Prediction and Image Deflection." Journal of Aircraft, Vol. 29, No. 6 (Nov-Dec 1992), 1064-1068.

Ward92 Mark Ward, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs, "A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems," 1992 Symposium on Interactive 3D Graphics, pp. 43-56.

Whitted80 Turner Whittted, "An Improved Illumination Model for Shaded Display," Communications of the ACM, Vol. 23, No. 6 (June 1980), pp. 343-349.