High Fidelity for Immersive Displays

Gernot Schaufler
GUP
Kepler University Linz, Austria
gs@gup.uni-linz.ac.at

Tomasz Mazuryk **Dieter Schmalstieg**
Institute of Computer Graphics
Vienna University of Technology, Austria
mazuryk@cg.tuwien.ac.at dieter@cg.tuwien.ac.at

ABSTRACT
Head-tracked immersive displays suffer from lag and non-uniform frame rates. A novel rendering architecture is proposed that combines head prediction with dynamic impostors for 3-D image correction and achieves bounded frame rates and significantly reduced lag.

Keywords
virtual reality, head tracking, immersion, lag, prediction, uniform frame rates, impostors

INTRODUCTION
The aim of virtual reality applications is to create the feeling of immersion by presenting convincing stimuli, in particular images, to the user, and allowing application control by direct interaction. Head-mounted displays are used to replace the user's view of the world with the computer output. Tracking of head movements allows direct control of the viewpoint in 3-D.

The degree of immersion in today's VR systems is not really satisfying. Limitations of quality are largely due to two factors:

- System lag is unacceptably long, causing image "swimming" and "overshooting" behavior.
- Non-uniform and low frame rates cause discomfort and prevent effective interaction with the system.

ENHANCING THE RENDERING PIPELINE
Proper software design of virtual reality systems must be used to deal with these disturbing factors. In particular the lag introduced by the hardware must be compensated for and strategies must be built into the system which deal with potential overload of the hardware. Important approaches are summarized in this chapter. In the next chapter a software design is proposed which actually removes the two severe quality restrictions discussed in the introduction.

Head prediction
To partly compensate for the lag, one can predict head movements and use the predicted head position for rendering. By the time the image is presented to the user, the predicted value will better match the real head position than the measured one, and lag influence will be reduced. Unfortunately the prediction accuracy decreases rapidly with longer prediction intervals. Moreover precision is further compromised by the noise present in magnetic tracker data. Mazuryk and Gervautz have presented an algorithm based on a modified Kalman filter [1]. Separate filters are used to correct the values for the position, velocity, and acceleration values, which are then used to compute the predicted values.

Model culling
Typically only a small portion of a large scene is visible at any time, so for a large geometric database, the hardware will spent most of the time dealing with objects that are not visible. A database preprocessing step for every frame can discard the invisible portions. Several approaches have been presented, such as viewing frustum culling, potentially visible set determination [2] and hierarchical Z-buffer rendering [3].

Approximate rendering with dynamic impostors
Even the most expensive hardware cannot provide uniform frame rates as long as there are no restrictions on scene complexity and user movement. Fortunately, it is not always necessary to generate an image with maximum quality, as it was found to be less disturbing to have reduced image quality than to have a highly variable (and possible low) frame rate. It may be better to render an approximate image than to present a frame late.

Schaufler has proposed a method for such approximate rendering based on dynamically generated impostors [4]. Separate images are produced for every object and stored in texture memory. The final image is composed by rendering a single polygon for every object with the object's pre-rendered image texture-mapped onto it. During times of high frame-to-frame coherence, most of the image data from the previous frame can be re-used.

The method includes level-of-detail (LOD) rendering based on multiple, progressively coarser geometric representations of polygonal models [5]. Selection of the appropriate approximation that gives the best possible image for a fixed frame time is done using a heuristic as presented in [6].

IMMERSIVE DISPLAY RENDERING PIPELINE
VR systems must be designed as "closed-loop simulations" where the user interacts with the system and immediately expects the results of his actions to be reflected in the system's output. Though the mentioned contributions can greatly enhance any VR system, it is the successful combination of all these techniques which yields top of the line performance.

To make the best use of the prediction mechanism, we split the rendering process into two stages. The first stage will render the objects. Despite optimizations with impostors and LODs, this step takes up so much time that prediction cannot be completely accurate. Thus, the second stage will generate the final image taking into consideration new tracker data. Image deflection is used

to better reflect the actual head position of the user. The use of impostors allows us to apply image deflection in 3-D which is far superior to conventional 2-D deflection. For an overview of our rendering architecture see fig. 1.

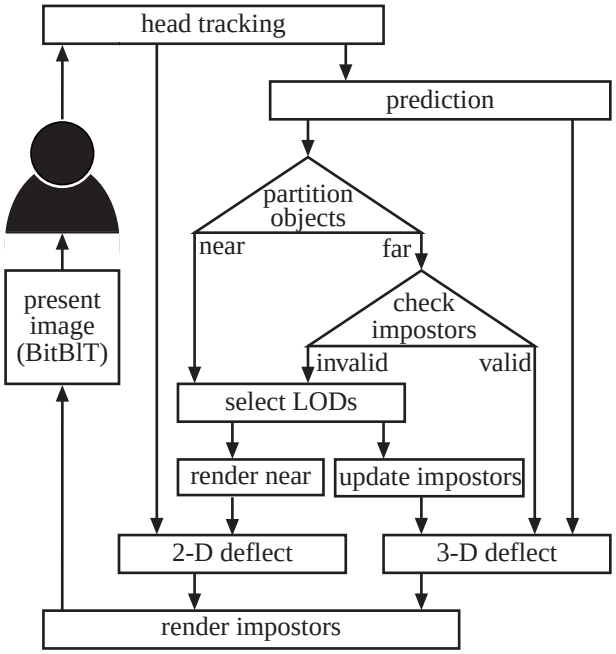

Fig. 1: Rendering pipeline for immersive displays

First stage: rendering
After obtaining the camera position by predicted head tracking, we select those objects that are potentially visible according to their bounding volume. These objects are partitioned into two sets: distant objects are represented by impostors (textures), while objects close to the observer are directly rendered into the frame buffer (impostors are not useful for close objects due to little coherence in their images and high texture memory requirements). Every impostor is examined for being still valid (for details, see [4]), only the invalid impostors are re-rendered, thus saving rendering time. For every object that must be rendered (either into a texture map or directly into the frame buffer), an appropriate geometric level of detail is chosen.

Second stage: image composition and deflection
The final image is composed from the frame buffer containing near objects and the impostors. Composition is done by simply rendering additional polygons textured with the impostors into the half-ready frame buffer.

Even with impostors and levels of detail, rendering takes time; usually newer tracker data becomes available as image generation progresses. Therefore we re-read the tracker and employ image deflection to compensate for the prediction error made in the first stage. We use different deflection strategies for the frame buffer and the impostors:

- The frame buffer can only be panned and scrolled in 2-D, which allows corrections to made be perpendicular to the line of sight (see [1]).
- In contrast, the impostors are 3-D polygons that can be transformed in 3-D to reflect the most recent head position. Panning, scrolling, rotation, and zooming can be achieved by simply modifying the camera transformation before rendering the impostor polygon. Thus the range of prediction errors that can be compensated for with 3-D deflection is significantly larger than in 2-D. Accuracy of the final image is greatly improved, in particular as 3-D deflection is able to compensate large amounts of perspective distortion in distant objects, caused by fast translatory head movements.

When the final image must be presented to the user, the standard approach is to use double buffering. Mazuryk et al. have shown [7], that using hardware buffer swapping can introduce a considerable delay, as the vertical retrace of the raster device must be waited for. With hardware accelerated bit-block transfer (BitBlT) operations it is safer and faster to copy the new image into the frame buffer. Especially in the situation when the vertical retrace has just been missed, the rendering system would block for a whole frame. With 2-D hardware acceleration the copying step itself is so fast that disturbing flickering does not occur.

CONCLUSIONS

This paper presented a rendering architecture for high fidelity immersive applications. The approach is suitable for typical man-in-the-loop graphics applications with head-tracking. It uses a multi-stage rendering pipeline and combines head tracking with prediction, LODs and dynamic impostors, which should lead to significantly reduced system lag and near uniform frame rates.

REFERENCES

1. Mazuryk T., Gervautz M. Two-Step Prediction and Image Deflection for Exact Head-Tracking in VEs. *Proc. of EUROGRAPHICS'95* (1995), 29-41
2. Teller S., Séquin C. Visibility Preprocessing For Interactive Walktroughs. *Proceedings of SIGGRAPH'91* (1991), 61-69
3. Greene N., Kass M., Miller G. Hierarchical Z-Buffer Visibility. *Proc. of SIGGRAPH'93* (1993), 231-237
4. Schaufler G. Dynamically Generated Impostors. *GI Workshop on Modeling, Virtual Worlds, Distributed Graphics* (Bonn, Germany 1995)
5. Schaufler G. Exploiting Frame to Frame Coherence in a VR System. *To appear: Proc. of VRAIS'96* (1996)
6. Funkhouser T., Sequin C. Adaptive Display Algorithm for Interactive Frame Rates During Visualisation of Complex Virtual Environments. *Proceedings of SIGGRAPH'93* (1993), 247-254
7. Mazuryk T., Schmalstieg D., Gervautz M. Zoom Rendering: Improving 3-D Rendering Performance With 2-D Operations. *Techn. report*: ftp://ftp.cg.tuwien.ac.at/ pub/TR/95/ TR-186-2-95-09Paper.ps.gz