2^{nd} International Workshop on ICT's, Arts and Cultural Heritage, November 2003
Athens, Greece

Image-Based Techniques for Enhancing Virtual Reality Environments.

D. Christopoulos *christop@fhw.gr*
A. Gaitatzes *gaitat@fhw.gr*
G. Papaioannou *gepap@fhw.gr*

Abstract
This paper describes the methodological aspects of the application of various established image-based graphics techniques in virtual reality applications, in order to visually enrich and extend the common capabilities of virtual environment visualization platforms. The paper describes these techniques and goes to the extent of explaining various practical implementation issues. Examples and application case studies are provided to demonstrate the enhancements.

1. Introduction

Virtual Reality is usually associated with large surround screen projection screen installations, stereo displays, head mounted displays and other high-tech hardware encountered in theme parks around the world. But virtual reality is in essence the art of illusion of simulating a fake environment where the spectator is immersed and cut-off from reality as effectively as possible, no matter the means to achieve this. Immersive environments have a significant impact on the memorization of information and the contextual linking of entities and meanings discovered in a virtual tour, rendering VR an effective tool for education, science, culture, arts and entertainment.

It is true, that hardware technology in haptics, displays and graphics processing, reaches new levels of perfection and speed every day, but at the same time, the demand of the spectators is increasing with equal or more rapid rates. High detail representations encountered in pre-rendered graphics are too slow to render in real-time systems and the same holds true for the light interaction and visual effects. Instead, VR and real-time visualization requires carefully simplified models and fast and clever algorithms to brute force rendering of off-line systems.

The classical computer graphics is model based rendering. The model specifies the geometry of the scene. usually as a 3D mesh and the surface properties. Images of the scene are generated by projecting the 3D model onto an image plane and determining the color of the image pixels using the surface properties. In order to increase realism the models complexity has to be increased. This can result in cost increase since the model construction will be more time consuming and complex and in an increase of the rendering time since more primitives has to be processed.

Alternative approaches to scene modelling and rendering are known to be the techniques of image based modelling and image based rendering. In the image-based approach, the world is modeled by a collection of example images and these are used to generate images or movies representing the scenes appearance at arbitrary points. The techniques often allow for shorter modelling times, faster rendering speed and improved levels of photorealism.

This paper summarizes the attempt to visually enhance Virtual Environments through the use of image based techniques already applied individually in the game industry and broadcast applications. The described techniques are already available and can be realized by commercially available visualization libraries like SGI’s OpenGL Performer™ and OpenGL Shader™, but can also be implemented with widely used APIs like OpenGL™ or DirectX™.

2. Video Source Overlays

Merging 3D worlds and real action is a common practice in cinematography. Chroma-keyed video sequences of actors and props filmed in front of a blue screen are easily overlaid and mixed with pre-rendered animation sequences and imaginary scenes. But what if this experience should be transferred to the VR set? Real actors may participate as virtual guides into the reconstructed world or be a part of the background activity of a marketplace and give an impression of familiarity, natural movement and look to the polygonal scenery [1].

The biggest problem is that most video formats do not provide an alpha channel (transparency) and, worse, most media libraries do not support this extra information, not to mention cross-platform portability issues. Therefore, it is difficult to infuse a blending mechanism in the video itself. Porting the video post-process notion of chroma-keying to a real-time rendering system is also a poor solution. An acceptable quality video source, especially for close-up inspection (e.g. a virtual guide) cannot be of a resolution less than 256X512 pixels wide for a figure or object covering 1/4-1/3 of the viable screen area. Filtering this image to derive an alpha value from the RGB hue at a frame rate of 15-30 fps is too demanding a process to have on a VR system already struggling to render all the textured geometry and handle the game engine events.

In our implementation, we have opted for an OpenGL® blending mode that uses double pass rendering to simulate masking [2] (Figure 1). The masking operation is as follows: An inverted mask is generated from the original source, which is white where the image should appear transparent, and black in the solid areas. Additionally, the background colour of the original image (e.g. blue for a blue screen) is blackened. The mask is rendered first with a disabled depth test and a blending function of (GL_DST_COLOR, GL_ZERO). Then, the normal image is rendered with a blending function of (GL_ONE, GL_ONE) on top of the mask.

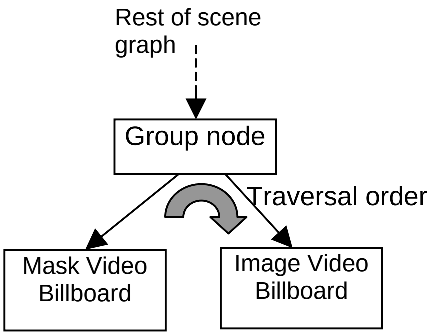

Figure 1: *Video Overlay on billboard geometry in a scene graph.*

Figure 2: *A Video source (girl) is superimposed as a billboard in the scene. The partial silhouette haloing is a result of the chroma-keyed transparent cloth and hair.*

This technique was originally applicable to low-level immediate OpenGL® rendering, but the idea was ported to a scene-graph-based environment, based on OpenGL Performer™. The original video is post processed to fit a power-of-two size, as the video source is used as a texture. Using chroma-selection, an inverted masking video is produced and saved in a separate video file. In each frame draw call, a time-synchronised frame is loaded from the buffer of both channels (mask and image) and placed on an identical piece of geometry. In the case of our VR engine, a billboard was adopted as the underlying geometry (Figure 2), as the original use scenario demanded an image of a real actor always moving in front of the user, in a guided tour.

Whenever the video source is to be applied as a normal video texture and not as an overlay in front of the rest of the scene, a more conventional, occlusion-culling-safe but more expensive alternative is used. The colour and transparency channel videos are blended frame-by-frame as they are being loaded by setting the alpha channel of the colour video to the greyscale value (or simply any RGB component) of the transparency video for each image pixel. Then, the image is applied as a normal texture to any surface using blending and alpha culling for the transparency. Although this approach is more robust, the heavy computational cost makes it more appealing to small video images (up to 512X256, for a frame-rate of 10-15fps on a 8-processor Onyx2).

3. Image-Based Modelling

High geometric detail is a key factor for the generation of a convincing virtual world. For parts of the scene that are far away from the observer or objects that the user does not interact with, it is a common practice to replace the geometric complexity with textures [3]. An important psychological aspect of the VR environments is that by constantly moving through the virtual world, the representation accuracy is mostly felt rather than meticulously observed. Therefore, if heavy, life-like texturing replaces detailed scene models,

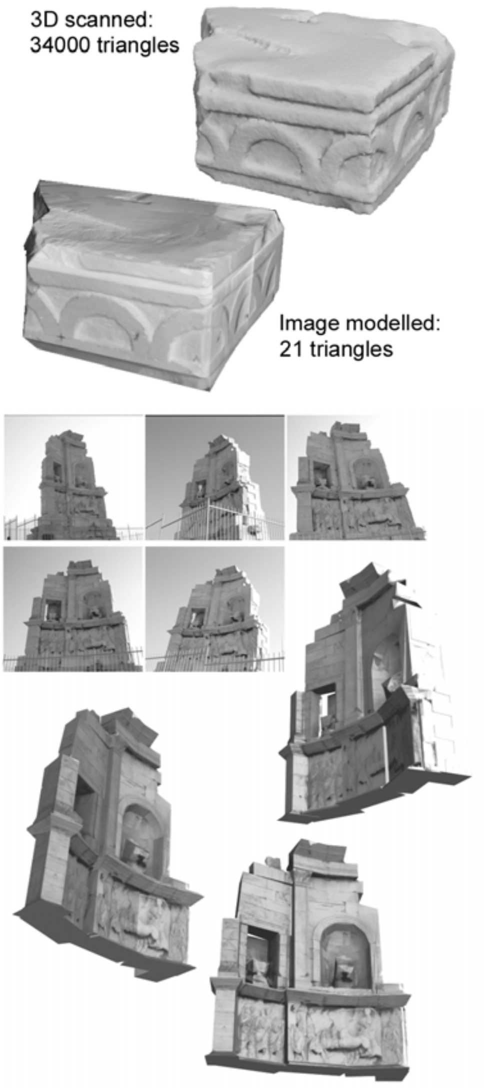

Figure 3: *Image-based modelling. Above: Visual detail is acquired with conventional photography and imprinted as textures on simple geometry. Below: The monument of Philopappou in Athens. Textures based on real pictures of the site cover a rough approximation of the monument's geometry.*

the image is going to "fit in" the senses and the impact is going to be greater than overloading the system with modelled details that no one is going to get too close to notice.

Extending this idea, image-based modelling can be used for the generation of simplified geometry from photographs of real sites or objects. The big advantage is that one gets at the same time a low polygon representation of the scenery and properly registered appearance and photorealism, directly from the recorded imagery (Figure 3).

Image- or photogrammetry-based modelling works well with both distant outdoor entities, like tree lines and whole buildings, and closely inspected crude objects like slabs and columns. The main disadvantage is the view dependency of the texture maps, which implies that in order to avoid distortion or exposing unregistered and untextured parts of the scene, one has to restrict the navigation activity close to certain paths. Evidently, image-based modelled objects have as low a polygon count as the modeller wishes but on the other hand, they suffer from the same restriction that applies to the illumination maps: The textures derived from the image capture and warping procedure cannot be tiled, thus increasing the memory utilisation of the graphics subsystems.

The procedure of extracting the geometry from a sequence of images itself is a fairly known but tedious semi-automatic task that involves defining and calibrating virtual cameras in 3D space that correspond to the photographs taken and the detection of common reference points in the original photographs [4]. Using computer vision methods, these points are back-projected in 3D space and can be used as building reference points to model a crude approximation of the observed geometry. By projecting, merging and clipping the original photographs on the polygons generated by hand, one can get the textures required to display the original objects in a convincing manner.

4. Image-Based Rendered Characters

The rendering of outdoor and indoor environments requires the synthesis of what are often considered two separate problems: the real-time visualization of large scale static environments, and the visualization of characters and probs.

In most applications the bandwidth and rendering capacity is used for the outdoor and indoor environments and buildings, which are usually the points of interest. In order to enrich these environments characters and other props might be used. To render many animated characters and props as complex polygonal models will result in non interactive frame rates, therefore a balance has to be made between the amount of models used and their complexity. Unfortunately using exclusively low polygon models comes into contradiction with the rest of the environment, which might be of better quality and is therefore bound to break the suspension of disbelief.

In order to minimize geometric complexity and achieve a balance between amount and quality of models, image based rendering and impostors are often used for specific models. Especially human representations have been successfully modelled using image-based approaches [5]. The principle of image based rendering techniques is to replace parts of the polygonal content in the scene with images. These images can be computed dynamically, as is the case with relief texture mapping [6], or a priori and are then placed as impostors into the environment using simple geometry like planes, which always face the viewer. Usually the images for the impostor are taken from different positions around the original object and are selected at run-time depending on the viewpoint position and the frame of animation.

In our recent application "Feidias' Workshop" we implemented IBR techniques for visualization of animated characters and discovered their usability and impact in VR and educational applications. Each character was replaced by an impostor, whose texture was changed depending on the viewpoint position and the frame of animation (Figure 4). Thus there was information for all the possible views of the impostor. The geometry used for the impostor was a quadrilateral facing always the user (billboard). Pre-rendered view-dependent textures were preferred instead of dynamic techniques like relief textures due to the serious computational overhead imposed on the processors, especially in the case of multiple IBR nodes.

Figure 4: *Image-based rendered characters from one of our recent applications.*

4.1. IBR advantages

Using IBR impostors has various positive impacts on the application if implemented properly. Since the rendering of the images is usually done offline the rendering speed in real-time applications is significantly faster in comparison to using polygonal models.

The original models from which the images are taken can be detailed, high-polygon-count models. These models, although being too expensive for real-time use, are perfect candidates for IBR techniques, since the geometric complexity does not deter the offline renderer and provides convincing visual details for the illumination interplay. This possibility of high-detail has been very beneficial in most applications. The human eye is accustomed to detect details on depicted human figures and therefore, complex pre-rendered imagery increases their photo-realism and natural appearance.

Modelling human characters for real-time use is a difficult and daunting job, which only skilled modellers and artists can perform efficiently. A low polygon character aims at being efficient in structure, polygon count and texture usage, while at the same time avoids sacrificing too much quality. This trade-off involves careful and skilled modelling. Models needed for IBR do not have these restrictions since they are rendered offline. Often commercial packages can be used to construct an initial high-polygon human model. Alternatively, the source for the images used on the impostor need not necessarily be modelled, but instead, they can also come from real photographical images or video captures.

Using impostors has proven to be ideal for characters that perform an isolated action in the background and which the user does not interact with or gets too close to. View-dependent impostors exploit frame-to-frame coherency efficiently and require less viewing angle instances if they are always far enough from the viewer. This way, there are slow and constrained changes to the visible sides of the depicted subject, thus requiring a sparse sampling of the viewing direction space.

Animation for characters rendered using IBR techniques can be realized faster and incorporated easier into an existing engine or framework than other full 3D animation techniques like morphing or kinematics. It is highly applicable for filling the environment with activity making it interesting and avoiding the typical VR walkthrough in empty environments with buildings and static trees.

4.2. IBR pitfalls and disadvantages

Although the use of IBR impostors may seem easy at first, it also has certain pitfalls and difficulties, which also apply to the video textures. The most annoying artifact when using impostors is the popping that occurs in the transition from on image (view direction) sample to another. Because the image samples were taken from certain positions there is no information about the depicted geometry for viewing positions in between them. This results in sudden and abrupt changes.

When getting too close to the impostor, pixelisation often occurs. The bigger the texture used the better the image quality and the more the texture usage on the graphics system. Using too small textures makes the image look blocky and using too big textures drains the bandwidth with texture downloads.

Because there is no information for the Z-buffer when rendering the impostor, rendering artifacts where the impostor plane intersects 3D geometry can occur, so careful scene design is important, in order to leave enough space around the impostor to allow it to freely rotate without bumping on the surrounding objects.

4.3. Overcoming IBR disadvantages

To mitigate the problem of popping between the image samples, a single-pass multitexturing algorithm is applied which blends the closest impostor view instances together.

In order to minimize texture usage and texture bandwidth, tests have to be performed about what texture sizes to use, so as to maximize the useful area that an image object occupies. The texture size depends on the object itself and the usage of the impostor in the scene. Impostors further away can have smaller texture sizes.

For easy production and preview of the impostors a special tool was developed. This IBR model pre-processor takes into account the desired texture size, the 3D model extends and the number of positions and elevations (rings) around the model from where the images would be captured. The software enables the user to interactively position and scale the model inside a window that has the size of the specified texture and also preview the model in various positions from where a capture will be made. This way, it is easy to place the model so that it occupies the maximum area of the image window without intersecting the image boundaries.

When rendering the images of the impostors, multi-sample antialiasing is used in order to reduce the pixelisation in the image texture and enable the usage of smaller, smoothed-out textures.

The view-dependent image sample density can be optimised with respect to the viewable range and number of samples for each impostor separately. The character or depicted object may be sampled from certain views from which it is known to be observed more frequently. For instance, if the character is to be viewed only upfront and rarely from above, viewpoint elevation ranges between -20 and 20 degrees should be more densely sampled than the rest, which, in some cases (see section 5) may be not sampled at all. This technique imposes some complexity on the rendering and transition between samples since each elevation can have different amount of views.

5. Case Study: The Workshop of Feidias

Feidias' Workshop (Figure 5) is an interactive virtual experience in the laboratory of the sculptor Feidias. It takes place in Ancient Olympia, amongst an accurate reconstruction of the famous statue of Zeus and the sculptor's tools, materials, and moulds used to construct it. Visitors become the sculptor's helpers and actively participate in the creation of the huge statue, by using virtual tools to apply the necessary material onto the statue and add the finishing touches.

We chose to incorporate characters depicting workers and Feidias, the sculptor itself, using IBR in the form of animated impostors. This technique was chosen because of its low polygon count since the detail of the models in the scene was high. The models and some animations were created using Curious Labs POSER© and Discreet 3Dstudio Max© in a fraction of the time compared to morph target 3D models we created for full 3D character animations in previous projects. The texture sizes used was 128x256 and 256x256 depending on the pose of the character. Despite this being a very low texture size, because of the blending and antialiasing techniques used to create the final images, the impostors looked quite satisfying even from up close.

Figure 5: *Image from Feidias' Workshop where most of the techniques described are visible.*

By careful positioning the characters the possibilities of artifacts when viewed from various angles, which may be caused by intersection of the impostor plane with the 3D models, were avoided. Furthermore we noticed that usually the characters were seen from upfront. Therefore the views perpendicular to the character (elevations between -20 and 20 degrees) were optimised using more sample images. In some cases it was sufficient to have sample images only for these views, thus providing no sample images for other elevations. Due to the fact that each impostor had a very sparse set of images associated with it, we opted for a closest neighbour selection criterion of the view-dependent textures rather than a blending mechanism. Interpolation between inconsistent views tended to produce blurry results and, as the characters were mostly viewed from arbitrary angles, they always had an insubstantial appearance.

The result was a reconstruction of a populated and active workshop in which the visitors would also observe and learn about the habits and appearance of workers during that time. The use of characters even as impostors enriched the environment tremendously making it an interesting experience for all ages and types of visitors.

References

1. A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, “Video Textures”, *Proceedings of SIGGRAPH 2000*, ACM, pp. 489-498, 2000.
2. The Neon-Helium Productions Tutorials: http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=20
3. P. Sander, X. Gu, S. Gortler, H. Hoppe and J. Snyder, “Silhouette Clipping”, *Proceedings of SIGGRAPH 2000*, ACM, pp. 327-334, 2000.
4. O. Faugeras, *Three-Dimensional Computer Vision*, MIT Press, 1993.
5. F. Tecchia, C. Loscos and Y. Chrysanthou, “Image-Based Crowd Rendering”, *IEEE Computer Graphics & Applications*, 22(2):36-43, 2002.
6. M. M. Oliveira, G. Bishop, D. McAllister, “Relief Texture Mapping”, *Proceedings of SIGGRAPH 2000*, ACM, pp. 359-368, 2000.