Eurographics Symposium on Virtual Environments (2004)
S. Coquillart, M. Göbel (Editors)

Foveated Stereoscopic Display for the Visualization of Detailed Virtual Environments

Guy Godin, Philippe Massicotte and Louis Borgeat

National Research Council of Canada, Ottawa, Ontario, Canada

Abstract

We present a new method for the stereoscopic display of complex virtual environments using a foveated arrangement of four images. The system runs on four rendering nodes and four projectors, for the fovea and periphery in each eye view. The use of high-resolution insets in a foveated configuration is well known. However, its extension to projector-based stereoscopic displays raises a specific issue: the visible boundary between fovea and periphery present in each eye creates a stereoscopic cue that may conflict with the perceived depth of the underlying scene. A previous solution to this problem displaces the boundary in the images to ensure that it is always positioned over stereoscopically corresponding scene locations. The new method proposed here addresses the same problem, but by relaxing the stereo matching criteria and reformulating the problem as one of spatial partitioning, all computations are performed locally on each node, and require a small and fixed amount of post-rendering processing, independent of scene complexity. We discuss this solution and present an OpenGL implementation; we also discuss acceleration techniques using culling and fragments, and illustrate the use of the method on a complex 3D textured model of a Byzantine crypt built using laser range imaging and digital photography.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Viewing Algorithms; I.3.7 [Computer Graphics]: Virtual Reality

1. Introduction

1.1. Motivation and context

As the complexity of virtual environment models continues to increase, driven by improved creation tools and techniques as well as powerful commodity graphics hardware, most immersive displays still provide levels of image resolution that have remained almost unchanged for the last decade. In the particular case of wall-type displays, screen pixel size typically can reach several millimeters. This limitation is likely to become an issue as virtual environment displays become increasingly used for exploring models with a high level of details in geometry and texture. Increasing the actual visual resolution of a display can be achieved by combining several units into a seamless unified display. Such systems are now common in virtual reality, for example in CAVE and PowerWall displays. For scientific visualization and design review, significant work has been done in combining large numbers of off-the-shelf devices into tiled displays (see for example the collection of papers in [FL00]). The projectors are aligned using either rigid mechanical setups or automatic camera-guided techniques to achieve resolutions of tens of millions of pixels. Recently, similar strategies have been proposed for creating large-scale tiled stereoscopic displays [BGA*03, KRK03].

Another technique for increasing the apparent display resolution adds small high-resolution insets within a larger, lower-resolution field of view. Apart from their resolution (and possibly other appearance factors such as color or brightness), the images in the low and high resolution areas appear as a unified display to the user. Such high-resolution inset methods have been proposed and used extensively in flight simulators and head-mounted displays (e.g. [YTK89, Fer95, YRR95, Eva04]); in the hybrid technique of [LIWL03], a dual-resolution view is presented to the user with the combination of a head-mounted display and projected background images. When the location of the inset in the visual field is coupled with gaze tracking, these methods can yield an impression of increased resolution in the

entire field of view for a single-user system. The concept of variable-resolution images is also used with uniform resolution devices: in this case, a portion of the image appears at the maximum available resolution, while the remainder is displayed at a reduced resolution, as a way to limit transmission bandwidth or computational load. Advantages and perceptual aspects of variable-resolution displays have been studied in [PN02].

The wide availability of digital projectors has considerably simplified the design and implementation of dual-resolution (or foveated) displays by removing the need for complex electronic and optical designs: in the "focus+context" system [BGS01] a high-resolution LCD panel display is surrounded by a very large screen on which a lower resolution image is projected; a pair of projectors is used in the Escritoire project [AR03] to create a desktop that incorporates a high-resolution area for improved document viewing. A high-resolution inset was also shown as one of the operation modes of the PixelFlex system [YGH*01]. Issues of appearance matching and screen irregularity in a two-projector foveated display are addressed in [TWH03]. These systems are not gaze-contingent, as they are often targeted to multi-viewer usages. With projectors, dual-resolution displays are significantly simpler and cheaper to realize than tiled displays; obviously, the drawback is that the gain in resolution is limited to a subset of the display.

Recently, we introduced a *stereoscopic* version of projector-based foveated displays [GLB04a]. However, in extending the approach to stereo, we identified a specific issue not present in monoscopic foveated displays: the visible boundary between the high-resolution inset and the low-resolution periphery creates a stereoscopic depth cue with a disparity which, in general, does not match that of the underlying scene. This creates a competition between two perceived layers of depth. The effect is particularly strong when virtual objects are located in front of the screen. We proposed a first solution where the apparent position of the boundaries between inset and periphery is moved (by drawing a black region in the inset and replacing this portion of the display by the low-resolution image) until it lies over corresponding points of the scene. These matching points can be found explicitly because the scene geometry is known. The proposed method reduces the computational cost of boundary matching by using the depth buffer in each view as a proxy for the visible part of the scene [GLB04b].

1.2. Contributions

In this paper, we propose a new computational method for our stereoscopic foveated display, which also addresses the same perceptual issue along the inset boundary as [GLB04a]; but instead of solving the problem at the *image boundary* level, we formulate it using a *spatial partitioning* approach. Our method is aimed specifically at detailed models of virtual environments: as other foveated displays, it provides access to a high-resolution inset, or fovea, within a larger display which appears as a unified stereoscopic display. In the context of large virtual models, the high cost of rendering a frame precludes any technique requiring more than one rendering pass or even computations on the scene model itself. Furthermore, we want to remove the need for post-rendering readback or exchange of depth buffer values as in [GLB04b], since the communication between the nodes introduces an additional undesirable lag in the navigation unless fast interconnections are used. The new method adds a very low computational cost, is applied locally on each rendering node, and is independent of the complexity of the scene model.

The solution is presented for projectors that are only approximately aligned, thus requiring that each image be warped to match a common screen-centered coordinate system. The warping can be applied as part of the rendering pass, with a form of the matrix designed to ensure that the finer resolution of the depth buffer around the near clipping plane is preserved.

Figure 1: *Screen photograph of the foveated stereoscopic display (the view for a single eye is shown for clarity).*

Figure 1 shows a screen photograph of our display. To avoid the blurring of superimposed left and right views, only the right eye view is shown. The brighter inset area contains the same number of pixels as the surrounding image (1024×768). The screen size is 3m by 2.3m, thus a viewer standing at 1.5m (for a 90° horizontal field of view) would see 3mm pixels, subtending an angle of about 6.8 arcmin in the center. The inset projection area is approximately 1m wide, yielding a resolution of 2.3 arcmin, which still does not match the limit of visual acuity, but provides a significantly enhanced experience of the virtual world.

The core of the method proposed here is the scene-adaptive assignment of drawing between the inset and the periphery in order to alleviate the boundary artefact identified in [GLB04a]. The paper first describes the geometry of a stereoscopic foveated system. Then the general method of spatial partitioning between low and high resolution display

is introduced, and its correctness is established. We outline an OpenGL-based implementation, and we discuss fragment program implementation and scene graph culling. The form of the projection matrix for viewing frusta of projectors that are not aligned with the screen is then discussed. We finally show an example of the display of a detailed model, followed by discussion.

2. Foveated Stereoscopic Display

2.1. Foveated projector configuration

Foveated displays superimpose a high resolution inset image within a larger, lower apparent resolution image. By analogy with biological vision, the high resolution central area is called the *fovea*, and the outer lower resolution area is referred to as the *periphery*. However, this terminology should not be interpreted as necessarily implying the use of gaze-tracking in order to follow the physiological fovea in a single user system: the system described here inserts a fixed high-resolution area. A *monoscopic* foveated display is obtained with overlapping projectors by blackening the portion of the peripheral image corresponding to the fovea, and replacing it by the smaller (thus higher on-screen resolution) fovea image. The two images are geometrically aligned so that their contents superimpose. This alignment may be automatic or manual, and may require image warping and trimming if the projectors are not prealigned to project rectangularly on the screen. These issues are discussed in [AR03].

Builders of projector-based tiled displays have worked on compensating for the projectors' lack of uniformity in brightness and color (e.g. [MHTW00][KRK03]). In [TWH03], similar methods were applied to a foveated system. However, assuming that two projectors of identical power are used for the fovea and the periphery, the apparent brightness of the fovea will be naturally increased. To be useful, a fovea must be significantly smaller than the periphery so that the gain in resolution justifies the setup complexity: for example, if it is one third of the width and height of the periphery, then the relative gain in resolution is equivalent to a 3×3 tiled display, and only $1/9$ of the pixels of the peripheral image are unused. But this also means that the fovea area is 9 times brighter than the periphery. Attempting to match the brightness of the fovea will require significant attenuation of the fovea, or the use of a much more powerful projector for the periphery. Furthermore, the cost of projectors increases very rapidly with their power. The increased brightness provided by foveation may actually be seen as an additional advantage of the technique. These considerations lead us to preserve a brighter fovea as a focus of interest area. In the systems proposed in [BGS01] and [AR03], the foveal area also remains discernable.

Creating a *stereoscopic* display by combining two foveated displays (one for each eye) yields the ambiguous depth perception problem along the boundary identified in [GLB04a, GLB04b]. The boundary between fovea and periphery visible in both eyes is perceived at a depth which depends on the position of the quadrilateral footprint of the projector image on the screen. But this perceived depth will in general not correspond to that of the scene that is drawn at the boundary. This ambiguity is illustrated in Figure 2(a), where points along the boundary can be matched either to the boundary or to the scene in the other eye view, thus creating a conflicting depth perception at a given location in the visual field. The solution in [GLB04b] exploits a key advantage of using overlapping projectors for foveated displays: the apparent screen position of the boundaries between fovea and periphery can be moved (within the limits of the on-screen footprint of the fovea projector) by drawing a portion of the fovea in black and replacing it by the corresponding area in the periphery. The boundary is positioned over matching scene points in both eyes (Figure 2(b)).

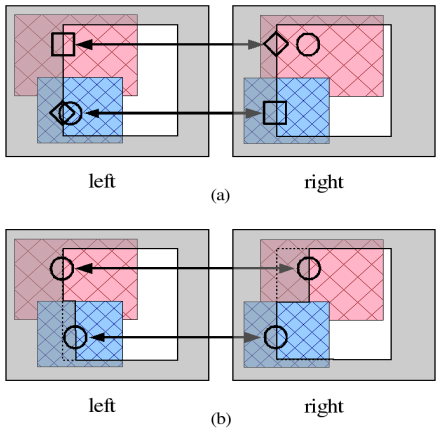


Figure 2: *Stereo matching ambiguity along the boundary. (a) points along the boundary in one eye (square) can be matched in the other eye either to the boundary (diamond) or to the scene (circle); (b) the boundary is displaced so that it matches the underlying scene (circles).*

The method proposed here also suppresses the ambiguous depth perception due to the fovea boundaries. It improves on our previous scheme [GLB04b] by addressing it as a *space partitioning* problem instead of as boundary matching, which, combined with a redefinition of the boundary positioning constraint, will be shown to remove the need for inter-node communications and depth buffer readback.

2.2. Projection geometry

We first review the imaging geometry for a foveated stereoscopic viewing configuration. It is described for a setup composed of four projectors and using the passive polarization stereoscopic display technique. A similar argument can be made in a time-multiplexed (or "active") stereo system, in this case the on-screen footprints are the same for the left and right views of the fovea or periphery.

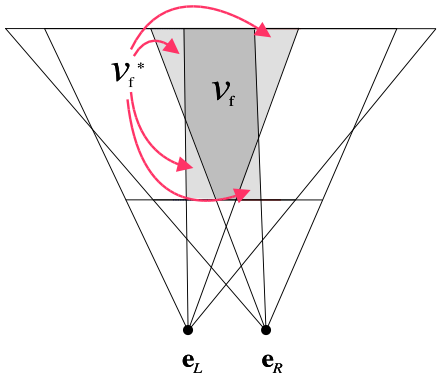


Figure 3: *Viewing geometry. The fovea frustum is contained within the frustum of the peripheral image. Only points in the $\mathcal{V}_f$ area can be perceived in stereo at high resolution; points in $\mathcal{V}_f^*$ are visible only in one of the two fovea images.*

Two pairs of projectors illuminate, for each eye, a periphery and a fovea area on the screen. The area occupied by each projector image is a quadrilateral, not necessary a rectangle since the projectors are not required to be perfectly aligned. The fovea quadrilateral is contained within the quadrilateral of the periphery of the same eye. Figure 3 shows the viewing frusta for the four images. The screen is assumed to be planar. The viewer's left and right eye positions with regards to the screen, $\mathbf{e}_L$ and $\mathbf{e}_R$, are known, or at least assumed. A proper stereoscopic view is obtained by rendering the scene using projection matrices reproducing the view for each eye position. For each of the projectors, each of the four sides of the quadrilateral, along with the eye position associated to its view, defines a plane. Common near and far clipping planes are applied to all four views, forming a frustum volume for each projection: $\mathcal{F}_{p_L}$ and $\mathcal{F}_{p_R}$ for the periphery and $\mathcal{F}_{f_L}$ and $\mathcal{F}_{f_R}$ for the fovea images, in the left and right eyes respectively. The images formed by each of these projection frusta are defined as I_{p_L}, I_{p_R}, I_{f_L}, and I_{f_R}. A necessary condition for a point $\mathbf{p}$ to be visible in I_{p_L}, I_{p_R}, I_{f_L} or I_{f_R} is that it is enclosed in the frustum $\mathcal{F}_{p_L}$, $\mathcal{F}_{p_R}$, $\mathcal{F}_{f_L}$ or $\mathcal{F}_{f_R}$, respectively.

The construction of the projection matrices corresponding to these frusta in the general case is deferred to Section 4, but the reader will easily recognize that, in the special case of a rectangular projection footprint, the frustum is in the form obtained using the OpenGL `glFrustum` function. Given that the eye point is the same for fovea and periphery in each eye view, and that the on-screen fovea projector area is contained within the periphery area, then $\mathcal{F}_{f_L} \subset \mathcal{F}_{p_L}$ and $\mathcal{F}_{f_R} \subset \mathcal{F}_{p_R}$. And because the purpose of the system is to produce stereopsis, then the projectors must be arranged so that $\mathcal{F}_{f_L} \cap \mathcal{F}_{f_R} \neq \emptyset$ and $\mathcal{F}_{p_L} \cap \mathcal{F}_{p_R} \neq \emptyset$. Only the portion of the scene contained within $\mathcal{F}_{p_L} \cap \mathcal{F}_{p_R}$ can possibly be perceived in stereo, and only the portion in $\mathcal{F}_{f_L} \cap \mathcal{F}_{f_R} \subset \mathcal{F}_{p_L} \cap \mathcal{F}_{p_R}$ can appear in stereo *and* at high resolution.

2.3. Stereoscopic viewing

A point in space can be perceived in depth by stereopsis if it is visible in both eyes: we will refer to such points as *binocular* points. Due to occlusions, some points in the scene will be visible in the view for one eye and not for the other eye. Such points are called here *monocular* points. The perception of their spatial position depends on cues other than stereopsis, as handled by the human visual system.

The method sought here must avoid the aforementioned boundary artefact, which occurs when points along the boundary between the inset and the periphery in each eye do not correspond to the same underlying position of the scene. The visible difference in appearance between fovea and periphery produces strong stereoscopic cues that conflict with those of the underlying scene. This problem can be restated as: a binocular point of the scene located near the boundary is displayed in the fovea in one eye, and in the periphery in the other, thus on opposite sides of the inset boundary. In this form, we see that the problem is solved if we ensure that *any binocular point of the scene is displayed at the same level of resolution (either fovea or periphery) in both eyes.* This rule ensures that the boundary artefact as depicted in Figure 2 is eliminated: indeed, any two neighboring binocular points located in one view across a boundary (one in periphery and one in fovea) will also form a boundary in the other view, and thus the boundary will be perceived at a spatial position located between the two points.

Monocular points can appear in either fovea or periphery, but of course, there is an advantage in displaying them at high resolution whenever possible, to maximize the amount of available visual information. A point of the scene appears in an image if it is included in the corresponding frustum *and* it is not occluded by another part of the scene (or otherwise made invisible). Determining visibility in each image is performed as part of the rendering process, normally using the depth buffer. Given that a point is visible in one image, determining whether it is a monocular or binocular point requires additional computation, for example by using a variant of shadow mapping [Wil78] at the cost of an additional invisible drawing pass from the other eye's point of view, or of reading back and sharing a subset of the depth buffer values between nodes as in [GLB04b]. This method essentially relocated the fovea boundary so that it is always positioned over the nearest binocular points along lines of stereo disparity. But while this requirement is sufficient, it is not necessary. We will see next that, by accepting to downgrade some of the monocular points that lie in one of the fovea images (and thus can appear at high resolution) to the low resolution image, it is possible to entirely avoid the explicit determination of the monocular or binocular status of a point.

2.4. Space partitioning method

We first define the *foveal volume* $\mathcal{V}_f$ as the intersection of the two foveal frusta:

$$\mathcal{V}_f = \mathcal{F}_{f_L} \cap \mathcal{F}_{f_R}$$

The volume $\mathcal{V}_f$ is a polytope with at most 10 faces (since the near and far planes are shared between the two eye views). It appears in dark grey in the 2D view of Figure 3. A necessary condition for a point $\mathbf{p}$ to appear in stereo in the fovea images I_{f_L} and I_{f_R} is that $\mathbf{p}$ lies inside the foveal frusta of *both* eyes, that is, $\mathbf{p} \in \mathcal{V}_f$. It must also be visible (not occluded) in both eyes, i.e. it must be a binocular point. If $\mathbf{p}$ is a monocular point, visible in either I_{f_L} or I_{f_R}, it should preferably be displayed at high resolution to maximize the amount of visual information. This includes monocular points contained in $\mathcal{V}_f$, but also those that lie in $\mathcal{V}_f^* = (\mathcal{F}_{f_L} \cup \mathcal{F}_{f_R}) \setminus \mathcal{V}_f$, that is, in only one foveal frustum. However, a binocular point in $\mathcal{V}_f^*$ *must* be displayed at low resolution, to satisfy the rule of identical level display between views, since it is not displayable in one of the two foveal images. But determining whether a given point is monocular or binocular requires a visibility test from the other eye view, at the cost of additional computation. In drawing a scene, the test would have to be applied to every fragment, a problem comparable to shadow mapping.

If we apply the rule that any point, monocular or binocular, which is included in $\mathcal{V}_f^*$ is *never* displayed at high resolution, we then remove the need for this visibility computation. The only disadvantage is in the loss of possible high-resolution viewing of monocular points located in $\mathcal{V}_f^*$, which typically represent a small proportion of the entire viewing volume.

From these observations, we can now state the scene-adaptive foveated display method as follows:

- Each of the four images composing the display, I_{p_L}, I_{p_R}, I_{f_L}, I_{f_R} is rendered using the corresponding view frustum;
- If a point is located within $\mathcal{V}_f$, it is drawn in the foveal images I_{f_L} and I_{f_R}, subject to occlusion testing in each image;
- This point is also drawn, but in black, at the corresponding location in the periphery images I_{p_L} and I_{p_R}, to ensure proper occlusion testing in the peripheral image.
- Conversely, a point outside $\mathcal{V}_f$ is displayed in I_{p_L} and I_{p_R}, and in black in I_{f_L} and I_{f_R}.

This algorithm is guaranteed to avoid the boundary artefact because it satisfies the following condition: any binocular point is perceived stereoscopically at the same level of resolution (fovea if within $\mathcal{V}_f$, periphery if outside), thus patches of binocular points form regions with boundaries that match in the two views. Whether a point is visible or not, and monocular or binocular, is never explicitly computed. Only inclusion in $\mathcal{V}_f$ needs to be established: this can be accomplished very efficiently, and even more importantly, completely independently on each rendering node.

3. Implementation

3.1. OpenGL implementation

We now present the method's implementation based on OpenGL fixed-function pipeline mechanisms, which can be added to any rendering program as a post-draw operation. The basic task is to determine whether a point displayed in the image belongs to $\mathcal{V}_f$ or not. Determining whether a portion of a scene is inside a volume is akin to the well-known shadow volume method [Cro77]. Stencil-based techniques provide an efficient solution, at the cost of drawing filled areas in the stencil buffer [Hei91][MHE*03]. But here, the shape of the shadow volume presents some particularities: by construction, the volume $\mathcal{V}_f$ is a convex polyhedron, the near and far clipping planes are almost always part of the volume, and in each view, up to four faces of the volume project as lines, since they correspond to the original planar faces of the foveal frustum of the current eye view. Instead of evaluating the inclusion of each fragment in $\mathcal{V}_f$ and then drawing it in color or in black, the four images are first completely rendered , and then, in a post-draw pass preceding the frame buffer swap, the individual pixels are blackened as required by the method.

This operation can be performed using a *z-pass* type of stenciled shadow volume computation, with the depth clamping extension enabled:

1. Calculate the geometry of $\mathcal{V}_f$ by intersecting the two foveal frusta; if the viewer's position with regards to the screen does not change, this computation needs to be performed only once;
2. Clear stencil bits;
3. Draw the front facing polygons of $\mathcal{V}_f$ (without changing the color or the depth buffers), incrementing the stencil when the depth test passes;
4. Draw the back facing polygons of $\mathcal{V}_f$, decrementing the stencil when the depth test passes;
5. Draw a black polygon without depth testing, but with stencil test active: pixel is drawn in black if stencil $= 0$ for the fovea (fragment is outside $\mathcal{V}_f$), if stencil $\neq 0$ for periphery (fragment is inside $\mathcal{V}_f$).

The result is that all image pixels that correspond to scene elements included within $\mathcal{V}_f$ appear in black in the peripheral images, and conversely for the fovea images.

We can avoid computing explicitly the shape of $\mathcal{V}_f$, and in particular the intersections between the planes: in this way, all computations are made on the different GPUs in a consistent manner. We make use of the OpenGL additional clipping planes function. Only the planar faces bounding the foveal frustum of the opposite eye are drawn, and they are clipped by the four sides of the current eye's foveal frustum to form the corresponding faces of $\mathcal{V}_f$. The planes of the current view frustum project to lines, thus need not be drawn: when drawing the foveal image, these lines form the borders of the image; when drawing the peripheral images, the plane

equations of the current eye's foveal frustum are loaded as `glClipPlanes`, and applied to the drawing of the faces of the foveal frustum of the other eye view. The final stencil-tested black polygon drawing (step 5) covers the full screen for the fovea images, but for the periphery images, it is sufficient to draw over a rectangular area covering the fovea, thus significantly saving on pixel filling.

The convexity of $\mathcal{V}_f$ also allows a slightly more efficient method than stencil-based shadow volume for the fovea images. Since $\mathcal{V}_f$ is convex, then there are at most two intersections between $\mathcal{V}_f$ and a ray from the eye to any scene point. Points outside of $\mathcal{V}_f$ can be blackened by simply drawing the planes bounding the frustum with appropriate depth buffer tests:

1. draw back facing polygons of $\mathcal{V}_f$ in black, with a depth test of less or equal;
2. draw front facing polygons of $\mathcal{V}_f$ in black, with a depth test of greater or equal.

Depending on the viewing geometry, it may also be necessary to draw in black, without depth test, the portion of the far clipping plane that is not included in the other eye's frustum. Overall, this solution saves approximately the equivalent of one full-screen fill over the shadow volume technique.

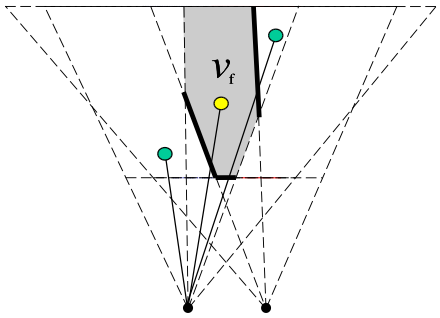


Figure 4: *Stencil-based inclusion test on $\mathcal{V}_f$.*

Figure 4 illustrates the stencil-based counting: a ray from the left eye to the object in $\mathcal{V}_f$ intersects the volume once, hence the stencil position is incremented once. The two other objects, either in front or behind the volume, cause the stencil value to be zero since in one case the ray does not cross the volume, and in the other it is incremented and decremented exactly once, when crossing front and back facing sides of the volume, respectively.

3.2. Using fragment programs

While the algorithm described above applies to any OpenGL fixed-function pipeline code, graphics applications using GPU fragment programs require even less modifications, do not incur the cost of additional fills, and do not even require the availability of stencil buffers. Because $\mathcal{V}_f$ is convex, we can use only the sign of dot products with the equations of the planes bounding $\mathcal{V}_f$ to test whether a point is inside or outside of $\mathcal{V}_f$, and to change the fragment color to black as prescribed by the method. The test is conclusively negative as soon as the point is detected to be on the wrong side of one plane. For the periphery images, at most eight plane equations need to be tested, corresponding to the side boundaries of the foveal frustum in both eyes. For the foveal images, at most four plane equations need to be tested, corresponding to the planes bounding the other eye's foveal frustum, since the four planes of the current eye are implicitly tested by viewport clipping. Testing against the near and far planes may also need to be applied explicitly in addition to the side planes, depending on the warping effect, as will be discussed in Section 4.

3.3. Scene graph culling

The context in which we propose this method is that of complex virtual environments: the full benefit of a foveated approach comes if the model used in the generation of the fovea images is also more detailed than the one used in the periphery, while maintaining similar frame rendering times. Fortunately, large models are likely to be structured as spatially partitioned scene graphs, so that view frustum culling already provides an efficient reduction in drawing time. In our setup, a multi-resolution representation of the model [BFG03] is used in order to adaptively increase the number of drawing primitives displayed in the fovea to match the gain in resolution while maintaining a target frame rate.

In drawing the fovea images I_{f_L} and I_{f_R}, the culling must be applied against the geometry of the frustum of the current eye $\mathcal{F}_{f_L}$ or $\mathcal{F}_{f_R}$, and not against $\mathcal{V}_f$, since points in $\mathcal{V}_f^*$, even though they will be visible in the periphery, still need to be drawn in black in the fovea for proper occlusion testing. For a similar reason, in the periphery, one cannot cull the geometry included in $\mathcal{V}_f$ since it must be drawn in the depth buffer to occlude possible elements appearing in $\mathcal{V}_f^*$ behind the frustum of the other eye (see Figure 5).

4. Projector Alignment

We describe our system for approximately aligned projectors, which means that they do not necessarily project a rectangular area on the screen. This capability is not only useful to avoid the tedious precise alignment of four overlapping projectors, but may prove necessary when the physical arrangement is restricted by the need to keep foveal projectors out of the projection cone of the periphery projectors. The method proposed here is similar to previous work in single-pass rendering and warping using the graphics hardware (e.g. [Ras00]), but we propose a different form of the matrix which optimizes the use of the depth buffer resolution at the near plane.

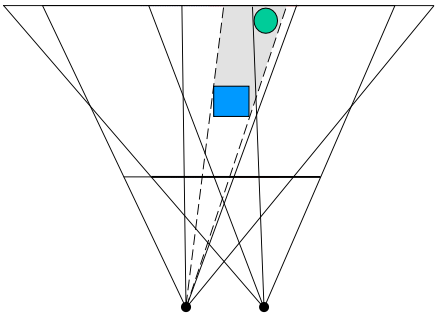


Figure 5: *Occlusion of an object displayed in the low-resolution image by an object in the foveal volume. The square object must be drawn (in black) in the peripheral image and not culled to ensure proper occlusion of the circular object.*

We first construct the ideal projection for each eye, defined by the eye positions and a reference *rectangular* area on the screen. This rectangle may be the footprint of one of the peripheral projectors if it has been physically aligned with the screen. The matrices corresponding to these projections, $\mathbf{P}_L$ and $\mathbf{P}_R$, are of the form given by `glFrustum`. The frusta for each of the four projectors are defined from these reference projection frusta in the following manner.

The positions in the images sent to the projector and their positions on the screen are related by a a 3×3 matrix [HZ00] called homography, or collineation. A given projector image position $\mathbf{x}_p$ projects onto location $\mathbf{x}_s$ on the screen, following the equation $\mathbf{x}_p \cong \mathbf{H}\mathbf{x}_s$, with $\mathbf{x}_p$ and $\mathbf{x}_s$ expressed as homogeneous coordinates. A 3-D point in the scene, $\mathbf{x}_w$, is projected on the ideal screen (and the depth buffer) with the 4×4 matrices $\mathbf{P}_L$ and $\mathbf{P}_R$ corresponding to the viewing frusta: $\mathbf{x}_s \cong \mathbf{P}_{\{L,R\}}\mathbf{x}_w$, with $\mathbf{x}_s$ now augmented to a 4-vector. If the homography for a projector view has non-zero h_{31} or h_{32} terms, then image warping must occur. This warping (as well as other transformations embedded in $\mathbf{H}$) can be applied as part of the rendering process [Ras00] by taking advantage of the 3-D rendering pipeline, instead of warping the resulting image as done for example in [AR03].

Let us assume that $\mathbf{x}_s$ and $\mathbf{x}_p$ are expressed in the interval [-1,1] along both axes, covering the entire reference rectangle or projector image. We define the matrix $\mathbf{G}$ from the elements of the corresponding homography:

$$\mathbf{G} = \begin{pmatrix} h_{11} & h_{12} & 0 & h_{13} \\ h_{21} & h_{22} & 0 & h_{23} \\ 0 & 0 & h_{33}+g & 0 \\ h_{31} & h_{32} & 0 & h_{33} \end{pmatrix}$$

where $g = \min_{(x,y)\in\mathcal{D}}(h_{31}x + h_{32}y)$, with $\mathcal{D}$ defined as the region occupied by the projector footprint in the ideal screen, expressed in the interval [-1,1]. The minimum over $\mathcal{D}$ is quickly found by evaluating the expression only at the four corners of the domain. This matrix premultiplies the projection matrix: $\mathbf{x}_p \cong \mathbf{G}\mathbf{P}_{\{L,R\}}\mathbf{x}_w$. If the result of the multiplication $\mathbf{G}\mathbf{P}_{\{L,R\}}$ is substituted for the projection matrix in the pipeline, then a single pass rendering and warping is obtained. The first, second and fourth lines of $\mathbf{G}$ simply apply the homography to the x and y components of $\mathbf{P}_{\{L,R\}}\mathbf{x}_w$. The third row pre-scales z to ensure that, after transformation and scaling by the dividing term which is a function of x and y, any point that would be included in the original viewing frustum is not clipped. But one consequence is that additional points, not originally part of the viewing volume, are no longer clipped; it is therefore necessary to introduce explicit clipping planes (e.g. using `glClipPlanes`) located at the original near and far clipping planes of the frustum. This matrix differs from the one proposed in [Ras00] by the optimization of the scale factor in z. We would like to preserve as much of the resolution of the depth buffer which, under perspective transform, varies non-linearly and is concentrated at the near plane. But the application of the near clipping plane to the previous matrix removes a portion of the volume, as a function of the warping and the ratio of near and far clipping planes.

The third row of the $\mathbf{G}$ matrix can be modified so that the near clipping plane of the warped volume is exactly aligned with the original clipping plane:

$$\mathbf{G}^* = \begin{pmatrix} h_{11} & h_{12} & 0 & h_{13} \\ h_{21} & h_{22} & 0 & h_{23} \\ -h_{31} & -h_{32} & h_{33}+g & g \\ h_{31} & h_{32} & 0 & h_{33} \end{pmatrix}$$

An explicit near clipping plane is no longer required, and more importantly, the most accurate part of the depth buffer is preserved. The effect of the different warping matrices is illustrated in Figure 6.

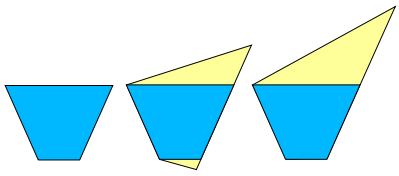


Figure 6: *Viewing frustum and warping: the visible volume of the original frustum (left) is modified by warping with* $\mathbf{G}$ *(center), and with* $\mathbf{G}^*$ *(right) which preserves the original near clipping plane.*

The homography can be estimated using a camera and computer vision methods [HZ00], or by manually pointing in one projector's image over targets displayed in the other projector. Four homographies need to be determined (three if one image is assumed to be the reference rectangle).

Using homography to perform screen alignment presumes

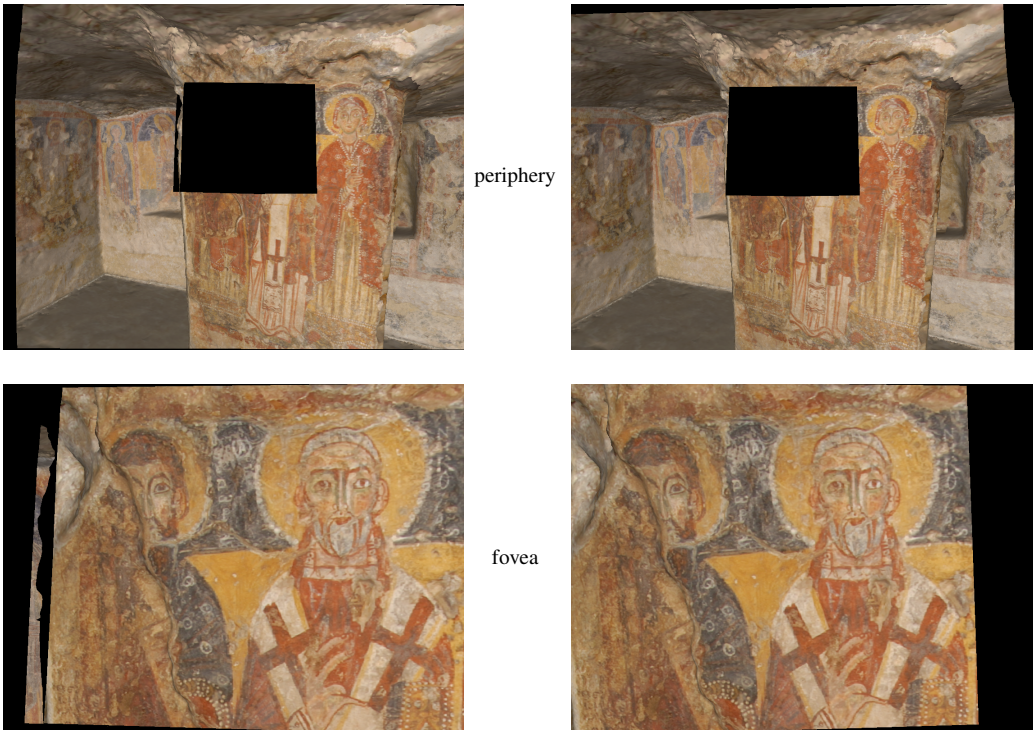


Figure 7: *The four images used for the generation of the view in Figure 1.*

that the screen is flat and that projectors follow a perfect pinhole projection model. Fortunately, many projectors actually have very low optical geometric distortion. In practice, this model has proved a reasonable approximation, yielding subpixel alignment between views.

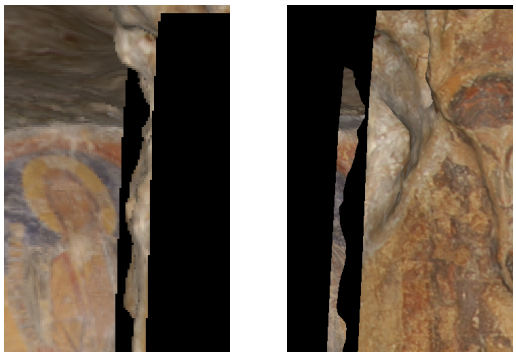
5. Example

Figure 1 illustrates the performance of our foveated display and the spatial partitioning approach: it shows the display as seen by the user (only one eye's view is shown to avoid blurring). This highly detailed 3D textured model represents the Crypt of Santa Cristina in Carpignano Salentino, Italy; it was built using a combination of laser range sensing and digital photography (see [BPEH*02] for more details). The accentuated brightness in the fovea is clearly visible. The contour of the foveal region is not rectangular: we slightly tilted one of the foveal projectors to illustrate the image truncation effect caused by the shape of $\mathcal{V}_f$. The four individual images composing this display are shown in Figure 7. Again, the contouring effect of the foveal volume appears clearly as a black area around the two foveal images. The effect of image warping is also visible.

The right edges of the view of the pillar in the left and right fovea images are stereoscopically matched since they are all binocular points. They form a fovea boundary that lies exactly at the same apparent depth as the surface of the pillar. On the left side of the left eye fovea image, there is a narrow strip of points belonging to the background wall surface behind the pillar, and occluded in the right view. These are monocular points that are included in the fovea image because they lie within $\mathcal{V}_f$. The black strip separating this group of points from the central part of the fovea image is composed of binocular and monocular points all located in $\mathcal{V}_f^*$. With the previous method of [GLB04b], the boundary in the left eye view would have been relocated over the points corresponding to the left edge of the right eye fovea image, and the monocular points on the narrow strip would have instead been displayed in the periphery. Figure 8 shows details of the complex shape of the delineation between the two levels of resolution. It also illustrates the significant difference in image resolution and scene details available to the viewer in the fovea.

Our system is based on passive stereoscopic projection with circular polarizing filters. Four commodity computers are interconnected for initialization, event sharing and buffer refresh synchronization, and generate images displayed over four DMD projectors. This scene was drawn using a custom 3-D viewer in which the method of Section 3.1 was implemented in OpenGL as a post-draw process. Performing all of the additional computations for the foveal display required on average around 1 msec on NVIDIA GeForce FX 5900

Ultra graphics cards. Since this time is mostly spent in filling of stencil or depth buffers, it is expected to continue to rapidly decrease as the speed of raster engines of graphics cards continues to increase.



periphery fovea

Figure 8: *Details of the transition between fovea and periphery along the left side of the column, in the left eye view; the difference in image resolution is also clearly visible.*

6. Discussion

Our system's context of applications is the viewing of detailed models of virtual objects and environments by a group of users. In this case, a mobile fovea linked to gaze tracking cannot be used to create, for a single user, the illusion of increased resolution over the entire field of view. However, a steerable fovea might be useful to avoid the current requirement of displacing the area of interest over the fixed fovea: a mobile fovea would impose only minimal modifications to the algorithms, but there are several practical opto-mechanical difficulties to overcome. On the other hand, there seems to be a natural tendency to locate areas of interest near the center of the screen, especially when several users are involved, and the benefits of increased resolution appear to compensate the inconvenience of a fixed fovea.

The current version of the method uses a "hard shadow" approach between the two levels of resolution. With a careful estimation of the projector-to-screen homographies, cross-fading zones did not prove to be necessary, since we are not trying to match the appearance of the fovea and periphery. We are nevertheless investigating efficient techniques for adding such zones, given the potential complexity of the boundaries between the two resolutions.

With digital projectors, the light levels in black areas are notoriously far from perfect. A foveated configuration imposes high requirements on the projector illumination performance. The fovea images are contained within the peripheral images, thus residual light from the blackened areas will be superimposed on the other images. This effect is probably the greatest performance drawback in a foveated configuration. In our case, using high-contrast projectors, the effect is noticeable mostly on dark scenes or backgrounds. In fact, in most cases, this problem proved to be less distracting than the cross-talk between the left and right eyes due to imperfect separation by polarization.

Our inset projectors were arranged to provide an image which is about one third of the width of the display, thus one ninth of the area. This value represented a compromise: obviously, a smaller fovea would increase even more the resolution, approaching the limit of visual acuity. On the other hand, a smaller fraction of the scene would be available at once in the fovea for examination, and more navigation in the model would be required. We may revise our design as we gather more experience in its usage.

As a direct consequence of the scene-adaptive positioning of the fovea boundary, the inset region is not perceived as a floating frame, but rather as a surface-conforming patch of increased resolution (and brightness) reminiscent, in absence of occlusions, of a spotlight directed on the scene.

7. Conclusion

In this paper, we have presented a new method for the foveated stereoscopic display of detailed virtual models. It provides an area of enhanced resolution within a large virtual environment display. This type of system can be set up at a low cost compared to tiled displays; most graphics applications can be modified to incorporate the image partition method between low and high resolution which is essential to eliminate the boundary artefact. Because it adds only a low and scene-independent computational cost, the technique is particularly suitable for the exploration of the object and environment models of high geometric and texture complexity that are now becoming available through advanced modeling tools and techniques.

Acknowledgments

We would like to thank our colleagues at the Visual Information Technology Group of the National Research Council of Canada, in particular Michel Picard and J.-Angelo Beraldin. We also thank Prof. Virginia Valzano, of the University of Lecce (Italy), for kindly granting us the permission to use the images of the Crypt in this paper.

References

[AR03] ASHDOWN M., ROBINSON P.: The Escritoire: a personal projected display. In *11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (3-7 February 2003), pp. 33–40. 2, 3, 7

[BFG03] BORGEAT L., FORTIN P.-A., GODIN G.: A fast hybrid geomorphing LOD scheme. In *SIGGRAPH Sketches and Applications* (27-31 July 2003). 6

[BGA*03] BRESNAHAN G., GASSER R., ABARAVICHYUS A., BRISSON E., WALTERMAN M.: Building a large scale, high-resolution, tiled, rear projected, passive stereo display system based on commodity components. In *Stereoscopic Displays and Virtual Reality Systems X, SPIE Proc. Vol. 5006* (2003), pp. 19–30. 1

[BGS01] BAUDISCH P., GOOD N., STEWART P.: Focus plus context screens: combining display technology with visualization techniques. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (November 2001), pp. 31–40. 2, 3

[BPEH*02] BERALDIN J.-A., PICARD M., EL-HAKIM S., GODIN G., VALZANO V., BANDIERA A., LATOUCHE D.: Virtualizing a Byzantine crypt by combining high-resolution textures with laser scanner 3D data. In *Proceedings of VSMM 2002* (25-27 September 2002), pp. 3–14. 8

[Cro77] CROW F. C.: Shadow algorithms for computer graphics. In *Computer Graphics (SIGGRAPH '77 Proceedings)* (July 1977), pp. 242–248. 5

[Eva04] Evans and Sutherland: Vistaview, image projection system for full-flight simulation, http://www.es.com/products/displays/vistaview/, (accessed February 2004). 1

[Fer95] FERNIE A.: Helmet-mounted display with dual resolution. *Journal of the Society for Information Display 3-4* (1995), 151–154. 1

[FL00] FUNKHOUSER T., LI K.: Special issue: Large format displays. *IEEE Computer Graphics and Applications 20*, 4 (July/Aug 2000). 1

[GLB04a] GODIN G., LALONDE J.-F., BORGEAT L.: Projector-based dual-resolution stereoscopic display. In *IEEE Conference on Virtual Reality 2004* (28-31 March 2004), pp. 223–224. 2, 3

[GLB04b] GODIN G., LALONDE J.-F., BORGEAT L.: Dual-resolution stereoscopic display with scene-adaptive fovea boundaries. In *8th International Immersive Projection Technology Workshop (to appear)* (13-14 May 2004). 2, 3, 4, 8

[Hei91] HEIDMANN T.: Real shadows real time. *IRIS Universe 18*, 1 (1991), 28–21. 5

[HZ00] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 7

[KRK03] KRESSE W., REINERS D., KNÖPFLE C.: Color consistency for digital multi-projector stereo display systems: The HEyeWall and the digital CAVE. In *9th Eurographics Workshop on Virtual Environments / 7th Immersive Projection Technologies Workshop* (22-23 May 2003). 1, 3

[LIWL03] LOW K.-L., ILIE A., WELCH A. G., LASTRA A.: Combining head-mounted and projector-based displays for surgical training. In *IEEE Virtual Reality 2003* (22-26 March 2003), pp. 110–117. 1

[MHE*03] MCGUIRE M., HUGHES J. F., EGAN K. T., KILGARD M. J., EVERITT C.: *Fast, Practical and Robust Shadows*. Tech. rep., http://developer.nvidia.com, Nov. 2003. 5

[MHTW00] MAJUMDER A., HE Z., TOWLES H., WELCH G.: Achieving color uniformity across multi-projector displays. In *IEEE Visualization 2000* (8-13 October 2000), pp. 117–124. 3

[PN02] PARKHURST D. J., NIEBUR E.: Variable-resolution displays: a theoretical, practical, and behavioral evaluation. *Human Factors 44*, 4 (2002), 611–629. 2

[Ras00] RASKAR R.: Immersive planar display using roughly aligned projectors. In *Proceedings of IEEE Virtual Reality 2000* (18-22 March 2000), pp. 109–115. 6, 7

[TWH03] TSAI Y.-P., WU Y.-N., HUNG Y.-P.: Generating a multiresolution display by integrating multiple projectors. In *PROCAMS 2003* (12 October 2003). 2, 3

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)* (August 1978), pp. 270–274. 4

[YGH*01] YANG R., GOTZ D., HENSLEY J., TOWLES H., BROWN M.: Pixelflex: a reconfigurable multi-projector display system. In *IEEE Visualization 2001* (2001), pp. 167–174. 2

[YRR95] YOSHIDA A., ROLLAND J., REIF J.: Design and applications of a high-resolution insert head-mounted-display. In *Virtual Reality Annual International Symposium '95* (11-15 March 1995), pp. 84–93. 1

[YTK89] YAMAGUCHI, TOMONO, KOBAYASHI: Proposal for a large visual field display employing eye movement tracking. In *SPIE Advances in Intelligent Robotics Systems, SPIE Proc Vol. 1194* (1989), pp. 13–20. 1