Dual-Resolution Stereoscopic Display with Scene-Adaptive Fovea Boundaries

Guy Godin* Jean-François Lalonde Louis Borgeat
Visual Information Technology Group
National Research Council of Canada
Ottawa, Ontario, Canada

Abstract

We present a multi-projector stereoscopic display which incorporates a high-resolution inset image, or fovea. The system uses four projectors, and the image warping required for on-screen image alignment and foveation is applied as part of the rendering pass. We discuss the problem of ambiguous depth perception between the boundaries of the inset in each eye and the underlying scene, and present a solution where the inset boundaries are dynamically adapted as a function of the scene geometry. An efficient real-time method for boundary adaptation is introduced. It is applied as a post-rendering step, does not require direct geometric computations on the scene, and is therefore practically independent of the size and complexity of the model.

1. Introduction

The rapid growth in processor and graphic rendering speed has been sustained for a number of years. Similarly, the size and complexity of the models to be displayed in virtual environments have increased dramatically, in part due to the emergence of 3-D digitizing tools and techniques which produce detailed models of real scenes and objects. However, these trends are not matched by a similar increase in the resolution of currently available unitary displays. The lowering cost of commodity projectors and graphics hardware has encouraged the approach of projector mosaic walls, achieving multi-million pixel displays: several example of such projects are described in [7]. They have found a widespread use in scientific visualization of very large data sets. Stereoscopic mosaic walls have also recently been built [4, 12]. As an alternative way of increasing the apparent resolution of a display, a high resolution viewing area is inserted within a larger lower resolution image. This approach is well-known in flight simulators and head-mounted displays (e.g. [6, 13, 17]). Recent projector-based implementations of this approach are known as “focus plus context” screens [2] and “foveal displays” [1]. Such systems require significantly less equipment and setup effort than mosaic displays while providing access to high visual resolution over a subset of the display.

This paper introduces a dual-resolution *stereoscopic* display system aimed specifically at the interactive rendering and examination of highly detailed 3-D models of objects and scenes. It brings the benefits of focus-plus-context/foveal methods to the projector-based stereoscopic displays of virtual reality. The method relies on commodity hardware, and requires minimum effort for projector setup. It applies image warping as part of the 3-D rendering pass, while ensuring consistency of the coordinate systems by enforcing identical near and far clipping planes.

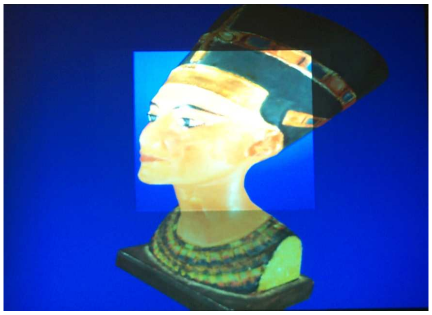


Figure 1. Dual-resolution monoscopic display. The paper extends this approach to stereoscopic displays.

Adding stereoscopic viewing to a foveated display introduces new challenges in system design. As identified in [8] and discussed further in this paper, simply adding an inset in each eye creates ambiguous depth perception between the boundary of the inset and the underlying scene. We present here details of an algorithm that addresses this problem in a general and computationally efficient way.

The paper first describes, in the monoscopic case, the single-pass image warping method required for a dual-resolution system. Then the new stereoscopic display con-

[*] Correspondence to: Guy.Godin@nrc-cnrc.gc.ca

figuration is introduced. We discuss why the straightforward approach of simply combining a pair of dual-resolution monoscopic systems produces a perceptual artefact due to conflicting stereoscopic cues between the boundaries of the fovea and the underlying scene. The solution lies in the scene-adaptive dynamic adjustment of the fovea boundaries which positions them over matching scene features. This technique adds only a low computational cost, which is a key requirement since this operation must be repeated at every change in scene or viewpoint. Finally, implementation issues and results are discussed.

2. Monoscopic dual-resolution system

A dual-resolution monoscopic display is composed of two projectors arranged such that a high-resolution small inset image $\mathcal{I}_1$ is projected within a larger image $\mathcal{I}_0$. Such an inset is often referred to as a *fovea*, by analogy with the biological visual system. The corresponding area of $\mathcal{I}_0$ is blacked out. The projectors need not be physically aligned in any predetermined configuration, except of course for inclusion of $\mathcal{I}_1$ within $\mathcal{I}_0$. Figure 1 shows such a setup. The image for $\mathcal{I}_0$ and $\mathcal{I}_1$ are generated on different 3-D graphics system. The entire display behaves as a single display screen with an enhanced resolution area.

2.1. Projection geometry

The projectors do not need to be aligned in any predetermined way with regard to each other or to the screen. Assuming a pinhole projection model for the projectors, the pixel positions in the two images are related by a 3×3 homography matrix $\mathbf{H}$, such that $\mathbf{x}_1 \cong \mathbf{H}\mathbf{x}_0$, where $\cong$ denotes equality up to a scale factor [10]. Here, $\mathbf{x}_0, \mathbf{x}_1$ are frame-buffer image positions, normalized in the interval $[-1, 1]$ within $\mathcal{I}_1$ and $\mathcal{I}_0$ and expressed in homogeneous coordinates:

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \cong \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} \qquad (1)$$

This invertible transformation determines the mapping between all image positions in $\mathcal{I}_0$ and $\mathcal{I}_1$. The matrix being defined up to a scale factor, we can thus impose, without loss of generality, that $h_{33} \geq 0$. The estimation of the homography matrix for the projector setup is discussed in Appendix. Once $\mathbf{H}$ is known, the image displayed in $\mathcal{I}_1$ can be linearly warped to corresponding locations of the image within $\mathcal{I}_0$ on the screen. Thus, one approach for the insertion of image $\mathcal{I}_1$ into $\mathcal{I}_0$ is to first render the image in a rectangular area enclosing $\mathcal{I}_1$, to read back the image, and to warp it using $\mathbf{H}$. This two-pass approach, used for example in [1], has the advantage of being directly applicable to arbitrary display contents, including existing applications and window management elements. However it may be slowed by the full screen read-back and retexturing, and image quality may be affected by resampling artefacts.

2.2. Single-pass rendering and warping

The work presented here aims specifically at the display of 3-D scenes, therefore the homography can be applied as part of the 3-D rendering pass, at no additional computational cost. Only minimal code modifications are required. The method differs from [16] by improving depth buffer usage and enforcing identical clipping planes. The system is described using the terminology, conventions and coordinate systems of OpenGL [15]. The 4×4 projection matrix $\mathbf{P}$ and model-view matrix $\mathbf{M}$ transform the scene geometry at rendering. A point $\mathbf{x} = (x\ y\ z\ 1)^T$ in the scene, expressed in homogeneous coordinates, is transformed into normalized device coordinates (NDC) using: $\mathbf{x}' = (wx'\ wy'\ wz'\ w)^T = \mathbf{PMx}$. Only points $\mathbf{x}'/w$ within the viewing volume enclosed by $[-1, 1]$ along the three axes are displayable. The x' and y' components of $\mathbf{x}'$ map to screen location, while z' is transformed into the depth buffer value.

The same projection and model-view matrices are applied to render both images, except that for $\mathcal{I}_1$, the transformed points in NDC undergo an additional multiplication by a geometric correction matrix $\mathbf{G}$ constructed from the elements h_{ij} of $\mathbf{H}$:

$$\mathbf{G} = \begin{pmatrix} h_{11} & h_{12} & 0 & h_{13} \\ h_{21} & h_{22} & 0 & h_{23} \\ 0 & 0 & h_{33} + g & 0 \\ h_{31} & h_{32} & 0 & h_{33} \end{pmatrix} \qquad (2)$$

where $g = \min_{(x,y) \in \mathcal{D}}(h_{31}x + h_{32}y)$, with $\mathcal{D}$ defined as the region occupied by $\mathcal{I}_1$, in normalized $\mathcal{I}_0$ coordinates. The minimum over $\mathcal{D}$ is quickly found by evaluating the expression only at the four corners of the domain. This matrix is designed to be applied to the geometry drawn in $\mathcal{I}_1$ expressed in NDC. This transformation is directly achieved for any 3-D point by premultiplying the projection matrix $\mathbf{P}$ (of any form, but here skewed perspective) by $\mathbf{G}$: $\mathbf{P}_g = \mathbf{G}\,\mathbf{P}$. The matrix $\mathbf{P}_g$ is loaded as the projection matrix for the rendering of $\mathcal{I}_1$. Thus, a point $\mathbf{x}$ undergoes the geometric transformation $\mathbf{x}' = (\mathbf{GP})\mathbf{M}\ \mathbf{x}$. It is easy to verify that the first, second, and fourth rows of $\mathbf{G}$ apply the same transformation to the x and y components as in Eq. 1.

The z coordinate must be handled consistently between the two images. Visibility testing by depth buffering must work, i.e. the depth function must remain monotonic, and the clipping planes applied to the scene in $\mathcal{I}_1$ and $\mathcal{I}_0$ must correspond exactly to avoid visual artefacts. In the general case where $h_{31} \neq 0$ and $h_{32} \neq 0$, the dividing term in the homogeneous coordinate $(h_{31}x + h_{32}y + h_{33})$ scales the z component as a function of x and y. Consequently, care must be taken so that points with normalized display coordinates outside of the $[-1,1]$ interval in z before warping, remain outside after multiplication by G. In [16], the z component is multiplied by a factor $h_{33} - |h_{31}| - |h_{32}|$ which ensures that the transformed z values are still included in the $[-1,1]$ interval. Since depth buffer resolution may be limited, the available range should be exploited maximally. The matrix proposed in Eq. 2 also ensures that any transformed point with a normalized z in the interval $[-1,1]$ remains within $[-1,1]$, but using only the smallest possible scale factor. Additionally, as a consequence of the warping transformation, the near and far clipping planes are displaced, and additional points may be included in the visibility volume. This extraneous volume is easily removed by adding two explicit clipping planes located at the original near and far planes in eye coordinates; the `glClipPlane` function in OpenGL efficiently performs this operation. Figure 2 illustrates the geometry of the viewing volumes associated to a perspective transformation $\mathbf{P}$ (wireframe frustum) for an image arrangement as shown in the lower right of the figure. The solid represents the volume for the inset enclosed in $[-1,-1]$ after transformation by $\mathbf{GP}$. The additional volume (in blue) is removed by the explicit clipping planes.

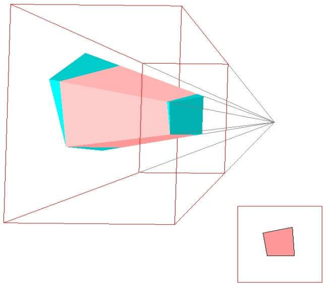


Figure 2. Geometry of the viewing volumes under warping transformation.

While $\mathbf{G}$ aligns $\mathcal{I}_0$ and $\mathcal{I}_1$ relative to one another, it may also be necessary to warp the two images together in order to align them with the desired on-screen boundaries. A similar method is applied, where the homography linking the common image coordinates and the desired screen location yields a matrix $\mathbf{G}_s$ which is applied in both images.

2.3. Appearance matching

One of the major issues encountered in tiled projector systems is the colorimetric and photometric matching along image boundaries so that the display appears as a seamless unit. Previous work has addressed this issue extensively in the context of large tiled projector displays (e.g. [14]). Difficulties arise from the properties of the projectors as well as from the projection surface. Most off-the-shelf projectors exhibit some level of non-uniformity in brightness across the image. There may also be significant differences in color, even between units of the same model. Furthermore, the projection surface itself may produce perceptible artefacts. The physical arrangement of projectors will result in differences in angles of incidence on the surface in the overlap area. If the screen exhibits non-Lambertian reflection or diffusion, which is often the case, then the brightness will also be viewpoint-dependent, and consequently, in the context of a multi-user display, corrections cannot be applied for multiple simultaneous viewing positions. The human visual system excels at perceiving small intensity discrepancies, especially along linear boundaries. Even slight differences, due to insufficient correction or to view-dependent effects caused by the screen, are likely to remain noticeable.

In the context of a dual-resolution display composed of two projectors of comparable power, the brightness of the inset image will be naturally higher due to the smaller area of projection. For example, an inset of one third the width of the display - a typical proportion in our setup - will be nine times brighter than the surrounding. At first it may seem desirable to blend the inset by attempting to match its intensity and color with the larger image. But this approach requires a significant attenuation. This increase in brightness may actually be seen as an additional advantage of a foveal configuration. We choose to preserve the increased brightness; a similar design choice is made in [1]. This difference in brightness helps distinguishing the zone of interest from the context provided by the larger image, as in focus-plus-context methods [2]. A small amount of feathering can be applied between the two images, mostly to hide slight differences due to misalignment, pixel quantification, as well as the transition in resolution.

3. Stereoscopic projection

Passive stereoscopic displays based on two projectors with linear or circular polarization have become one of the most common and cost-effective implementation of 3-D immersive displays. Each computer/projector/polarizer group generates the image for one eye. The viewer wears polarized glasses matched to the projectors. The two projector images must be aligned on the screen so that the viewing areas are superimposed and the appropriate line of stereo disparity (typically horizontal on the screen) is obtained. This requires either careful opto-mechanical alignment of the projectors, or applying homography-based image warping techniques such as in Sect. 2.2. In this last case, the displayed images are also trimmed (by drawing black areas) so that they match along the rectangular screen region within the intersection of the two images (Fig. 3(a)).

3.1. High-resolution stereoscopic insets

One of the contributions of this paper is to extend the projector-based dual-resolution approach to stereoscopic displays. We begin by superimposing two monoscopic dual-resolution systems as described in Sect. 2. This configuration uses four projectors, each driven by a separate computer and graphics card. It is assumed that the projectors are only roughly aligned: the only condition is that the low-resolution images for both eyes must overlap on the projection screen. The same holds for the inset images. The homography matrices relating the low and high-resolution areas in each eye, as well as the two views to the screen, must be determined (see Appendix). The overlapping inset images are also trimmed, but only to ensure a match along the stereo disparity direction (defined as horizontal for now). Figure 3(b) depicts the projection geometry for the stereo insets.

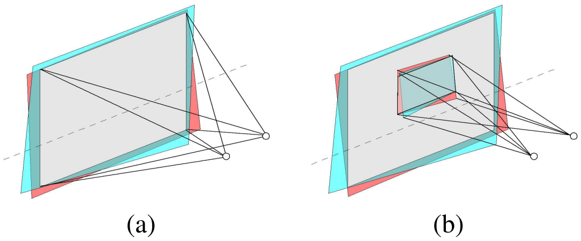


Figure 3. Stereoscopic setup. (a) two projectors; (b) adding insets with two additional projectors, from the same eye positions.

Following the same arguments as for the monoscopic case (Sect. 2.3), we choose to preserve the additional brightness provided by the smaller projection area. However, when superimposing two monoscopic insets, the intensity difference between the low and high-resolution areas create strongly perceptible edge features in both left and right views. These contours become stereoscopic cues that interfere with the depth perception of the scene (Fig. 4(a)): for a given screen location along the boundary in one eye (square), two different matches are possible in the other eye, based either on the intensity edge (diamond), or on the scene contents itself (circle). This matching ambiguity creates two competing layers of perceived depth along the boundary of the inset image, which interferes with the sought stereoscopic perception of the displayed surface. Moreover, we must remember that the images $\mathcal{I}_0$ and $\mathcal{I}_1$ in each eye differ not only in brightness, but also in resolution. Consequently, even when the brightness of all four projected images are matched, the change in resolution may still produce an undesirable effect.

The problem essentially arises from having the boundary between low and high resolution positioned over two different scene locations for each eye. This statement of the problem hints at the proposed solution: *the boundary of the high-resolution inset must be displaced so that it is always located over corresponding scene positions visible in both views.*

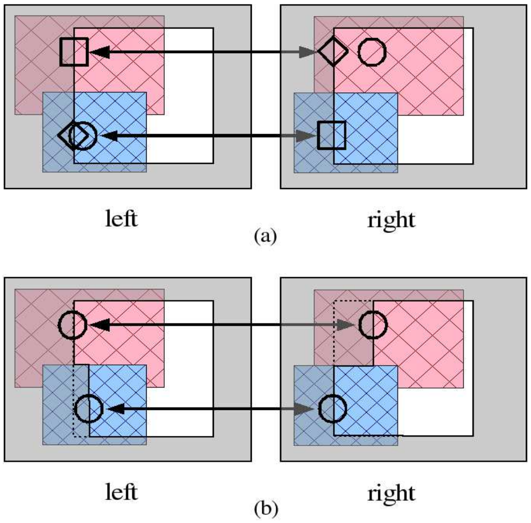


Figure 4. Stereo matching ambiguity at the boundary.

The boundary can be virtually displaced, within the footprint of the inset projectors, by drawing complementary black areas in the low and high resolution images. The outermost matching points (in a stereoscopic sense) in the inset are identified, and the boundary is displaced over these locations (Fig. 4(b)). Then, the perceived depth of the transition between low and high resolution is always identical to that of the underlying scene. This solution requires that the screen position of this boundary be recomputed whenever the underlying scene or the observer's position changes, generally for every displayed frame. Finding an efficient method for adjusting this boundary is therefore essential.

4. Scene-adaptive boundary placement

We propose an efficient method for the positioning of the boundary of the inset images, as prescribed in the previous section. It does not require an additional rendering pass, or even direct access to the scene's geometric model. The only requirement is that the image be rendered using a depth buffer, which will be used as a proxy for the visible portion of the scene in the view.

The goal of the method is to ensure that any non-horizontal portion of the inset boundaries is located, in both views, over the projection of the same scene location. The top and bottom boundaries of the insets have been made

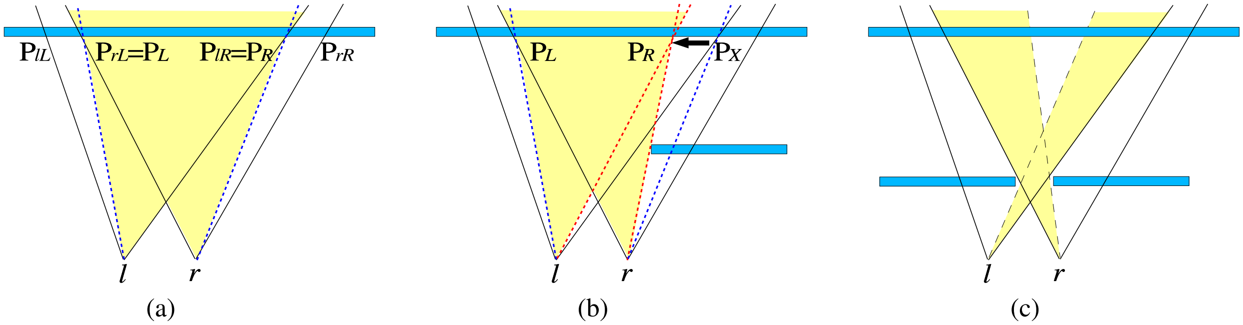


Figure 5. Adapting boundary locations. The left and right frusta for the insets only are drawn. (a) a simple case; (b) the effect of occlusion; (c) limit case of a keyhole.

horizontal by the initial framing (Fig. 3(b)) and do not need adjustment. The largest possible area of the insets should be used. The method operates by finding, in both eyes' views, the leftmost and rightmost points of the high-resolution display area that correspond to identical scene points. Chaining these points and connecting them with the top and bottom horizontal boundaries defines an area which serves as a complementary mask between the low- and high-resolution images. For each eye, the projected screen shape of the boundary will be identical in the low and high resolution images, but will differ between the left and right eyes, as it follows the shape of the underlying surface seen from the two different viewpoint. We first describe the solution in the continuous domain, along each horizontal slice, and then proceed to discuss its implementation.

4.1. Simple case

In a scene where there are no occlusions, that is where any point visible in one view is visible in the other unless it is outside the frustum (Fig. 5(a)), the corresponding boundary points are readily found:

- Find the intersection point P_{lL} between the scene and the *L*eft edge of the frustum (in scene coordinates) associated to the inset image of the *l*eft eye; if there are multiple intersections (which may occur in general but cannot be the case here in absence of occlusion) keep the closest point to the eye position.
- If P_{lL} is enclosed in the frustum associated to the inset image of the right eye (which, in absence of occlusion, implies that it is visible), then keep this point as P_L
- Otherwise, find the first intersection point P_{rL} between the scene and the *L*eft boundary of the frustum associated to the inset image of the *r*ight eye.
- If P_{rL} is enclosed in the frustum associated to the inset image of the left eye, then keep this point as P_L
- Repeat a similar process to identify P_R from P_{lR} or P_{rR}.

The points P_L and P_R are the bounding points of the high resolution image along this slice. By construction, their screen location will necessary project on the edge of the maximum inset projector area in one of the two views. Their corresponding image location in the other view is simply found by re-projecting them in that eye.

The intersection between the scene and the frustum need not be computed explicitly: it suffices to take the values stored in the depth buffer along those boundaries, and apply the inverse of the projection transformations to recover the coordinates of the original point. In absence of occlusion, this simple method computes the shape of the boundaries in constant time, requiring no search. However in practice, occlusion occurs and therefore the visibility of a point cannot be determined solely from its inclusion in a view's frustum.

4.2. Scenes with occlusions

Because of occlusions, the intersection points as computed above may be visible in one view but occluded in the other. The method for the simple case will thus yield a wrong match. One such case is depicted in Fig. 5(b), where intersection point P_X is hidden from view in the right eye. It is necessary to find another scene point that is visible in both inset views.

First, the points P_L and P_R are found using the algorithm for the simple case stated above. Each of the two points is tested for visibility in the other view. If it is not visible, then a matching point is sought along the stereo matching line. The search starts from the boundary point and proceeds inward until a point visible in both eyes is found, or the other side of the frustum is reached (this case is addressed in the next section).

Fortunately, here again it is not necessary to compute visibility on the scene model itself. For each point to be tested along the search line, the underlying depth buffer value is used to recover its original 3-D scene coordinates. This point is then transformed in the other view, and visibility is determined by testing it against the depth buffer of the other view.

4.3. Limit cases

For a given scene geometry and viewing configuration, it may happen that no point in the scene is visible in both eyes within the volume. In this case the method will fail to return boundary points. Such a situation may occur with a very busy scene (e.g. tree leaves) or with the simple "keyhole view" depicted in Fig. 5(c). The absence of points visible in both eyes implies that there are no stereoscopic cues for a correct perceptual surface reconstruction. In such a case, the problem of ambiguous match exposed in Sect. 3.1 becomes irrelevant. Consequently, the boundaries may be located based on other criteria, such as continuity with adjacent slices.

It is important to remember that, in all cases, an erroneous boundary placement only causes a perceptual artefact: the geometry is always displayed correctly on the screen.

4.4. Implementation details

The algorithm, described above in the continuous case, can be implemented almost directly on graphics hardware. The key differences are due to the quantization of the image and to the different raster line orientations between projectors.

The method is implemented as a post-rendering pass. For each view, the image is rendered with the appropriate point of view and warping. In our system, one computer node is used for each image. The four computers run an identical application, each with a complete copy of the scene model. They are inter-connected for frame-refresh synchronization and for the exchange of depth buffer values. The boundary adjustment method only requires the depth buffers and the projection matrices. The mask for screen trimming as well as boundary adjustment is computed on one of the nodes, and transmitted to the three others. Its interior (for $\mathcal{I}_0$) or exterior (for $\mathcal{I}_1$) is filled in black over the already rendered image (with optional feathering), before the image is made visible.

Instead of directly accessing the scene model, we use the depth buffer of the current image to find frustum boundary intersections and to test for visibility. For a given image position and the corresponding value of the depth buffer, (in normalized device coordinates), the original position of that point in the scene is recovered by unprojection: $\mathbf{x} = (\mathbf{GPM})^{-1}\mathbf{x}'$.

In each of the two inset images, the intersection between the side of the frustum and the scene is estimated by sampling and unprojecting points along the image borders. Each point p is tested for inclusion in the other view's frustum by applying the corresponding transformations and testing the resulting coordinates. If this test passes, then, following Sect. 4.2, we still need to verify that it is not occluded: this is achieved by projecting the point in the other image, and reading the depth buffer value at the pixel location (with interpolation). This point is then unprojected as p_c, and tested for $\|p - p_c\| < \tau$, where τ is a threshold based on the scene scale, the pixel size and depth buffer quantizations. A successful test is interpreted as the two points being the same in the scene. In this case, which corresponds to the simple case configuration, the point is kept as the boundary point. If the test fails, it means that we are in the condition of Sect. 4.2: the neighboring image point is selected, and tested again for visibility in the other eye. Iterations continue until a point is found, or the edge of the image is reached. For general projector configuration, a search along horizontal lines will not correspond to the raster scan lines of the image: we use the homography to compute the corresponding search lines in each image.

The method requires the sharing across views, thus computers, of the depth values for testing. Instead of repeatedly transmitting individual requests and coordinates, the depth buffer can also be transfered at the beginning of the loop. Transfer time is minimized by using only the rectangle bounding the inset in the low resolution image. If the scene to draw is very simple, or available in a simplified version, then a local redraw (in the depth buffer only) might be a more efficient alternative, but in practice the dual-resolution method is specifically aimed at complex models and scenes. Another possibility is to use dual-head graphics system in order to avoid communicating between computers.

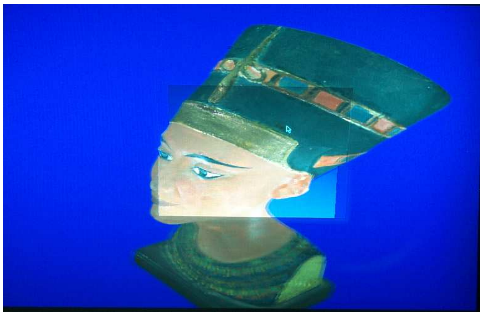
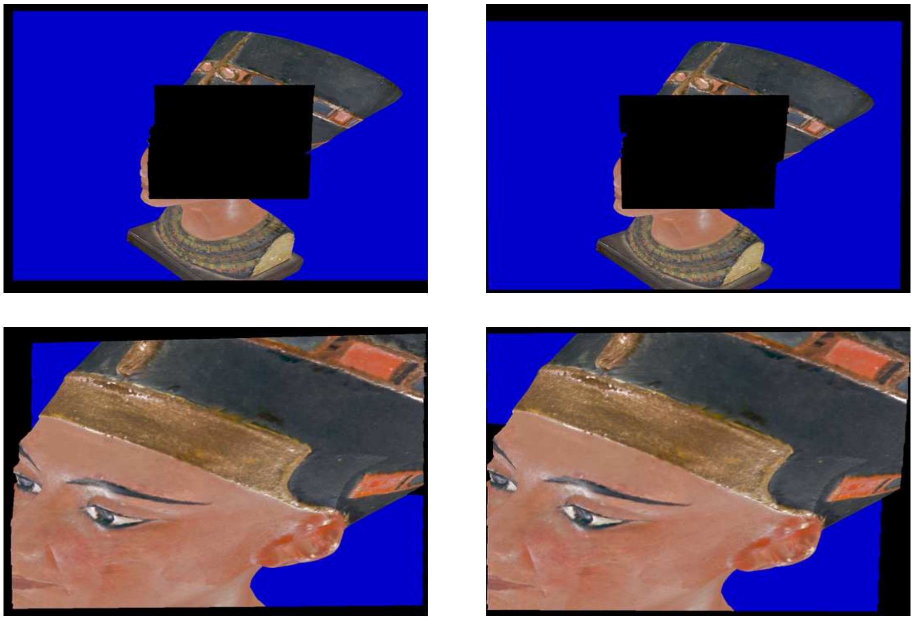


Figure 6. Stereoscopic dual-resolution display with adaptive boundaries (photograph of the screen with superimposed left and right views).

5. Results

The method described in this paper has been implemented in a prototype system composed of four Linux-based PCs with NVIDIA GeForce4 cards feeding four 1024×768 pixels DLP projectors fitted with circular polarizer filters. We project on a polarization-preserving wall screen. The projectors are only approximately aligned, the homography matrices linking the images are estimated us-



left right

Figure 7. Images used for the generation of the view in Figure 6.

ing a video camera (see Appendix). The superimposed right and left views appear in Fig. 6. The shape adaptation of the inset boundaries is clearly visible, especially when compared with Fig. 1. The four images composing the display are shown in Fig. 7. The effects of the homography warping and the black masking are shown along the top and bottom; the adaptive boundary adjustment appears around the nose and right eye area and on the crown in the right view. To the viewer wearing stereo glasses, the boundary does not appear to move laterally on the screen, but rather to conform to the shape of the objects along the depth axis.

6. Discussion

Dual-resolution displays provide access to a central viewing area with a resolution equivalent to that of a larger and more complex system. An inset of width $1/n$ times that of the larger image gives access to a visual resolution equivalent to that of a n^2 tiled display. Of course the viewer needs to navigate in the scene in order to bring a point of interest within the fixed fovea. But the reduction in complexity and cost over a mosaic wall is significant. The scene-adaptive modification of the fovea boundary is an essential component of the stereoscopic dual-resolution system, otherwise the perceptual ambiguity along the boundary significantly reduces viewing comfort. Fortunately this operation is achieved at a relatively low computational cost. The ability of this method to properly manage the transition justifies our initial decision of keeping the additional brightness gained from the smaller projection area of the inset.

To take full advantage of the dual-resolution setup, multi-resolution models of the scene are used [3] and rendered at a resolution based on the screen pixel size. Tighter culling bounds can be applied to the view in $\mathcal{I}_1$, which compensates for the additional load from the increased model complexity. Tight synchronization is required between the machines, in particular between the $\mathcal{I}_0$, $\mathcal{I}_1$ pair in each eye: otherwise, time aliasing may generate a false depth discontinuity around the inset boundary.

This concept was implemented in our laboratory as a stereoscopic single-wall display. Currently, the viewer position is not tracked, since our system is aimed at multi-viewer applications. However viewpoint adjustment is easily added by modifying the projection matrix; supporting head rotation, which redefines the horizontal disparity orientation on the screen, also requires a change in the search directions for matched points in the scene. Searching along a non-raster direction already occurs when warping is applied. A steerable fovea that could be repositioned within the larger image would avoid the main current limitation, which is the fixed position of the fovea within the field of view. This method could also accommodate gaze tracking for single user displays.

The main limitation of the current implementation is its reliance on rapid communication between machines for exchanging depth buffer values. We are currently developing a new method for boundary adaptation where, by relaxing the stereoscopic matching constraints, all computations are performed locally at a minimum and strictly constant cost: we will report later this year [9] on this new approach.

7. Conclusions

In this paper, we have proposed a new method for the stereoscopic display of 3-D models using a dual-resolution projector-based approach. The boundaries of the high-resolution inset, or fovea, are modified at every frame in each eye view in order to avoid ambiguous depth perception along its boundary. The boundary position is adjusted so that it is always located over corresponding locations in the underlying scene. We present an efficient method to perform this adaptation which does not require a second pass of rendering or a direct access to the 3-D geometric scene model. This display technique allows the exploration of the increasingly complex 3-D models now available, and access to details of a scene in the fovea while retaining the overall view of the environment. Because of its general nature, its low computational requirements and its minimal interference with the rendering process, this method can potentially be integrated into various existing virtual reality display systems.

Acknowledgements

The authors would like to thank Philippe Massicotte, Jean-François Lapointe and Shawn Peters for assistance and comments during the development of the prototype system.

Appendix: Estimating the homography

Various techniques for projector alignment using cameras have been proposed already (e.g. [5, 16]). We have implemented a similar homography-based technique, particularly useful here where four projectors must be aligned. The homography $\mathbf{H}_{i,c}$ between each projector i and the camera is determined by projecting a checkerboard target pattern, and using the OpenCV library to detect corner features [11]. Computing a homography requires a minimum of four non-collinear points; solving for an overdetermined system [10] is preferable especially given the camera resolution which is lower than that of the projectors. The homography between two projectors i, j is simply given as $\mathbf{H}_{i,c}\mathbf{H}_{j,c}^{-1}$. To avoid the use of a camera altogether, we can also take advantage of the inclusion of one image within the other. The mouse in $\mathcal{I}_1$ is positioned over targets displayed in $\mathcal{I}_0$: the higher resolution of $\mathcal{I}_1$ with regard to $\mathcal{I}_0$ allows for a subpixel alignment. The left and right views are then aligned by clicking additional targets displayed in the high-resolution insets.

References

[1] M. Ashdown and P. Robinson. The Escritoire: a personal projected display. In *11th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 33–40, 3-7 February 2003.

[2] P. Baudisch, N. Good, and P. Stewart. Focus plus context screens: combining display technology with visualization techniques. In *Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology*, pages 31–40, November 2001.

[3] L. Borgeat, P.-A. Fortin, and G. Godin. A fast hybrid geomorphing LOD scheme. In *SIGGRAPH Sketches and Applications*, 2003.

[4] G. Bresnahan, R. Gasser, A. Abaravichyus, E. Brisson, and M. Walterman. 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*, pages 19–30, 2003.

[5] Y. Chen, D. W. Clark, A. Finkelstein, T. C. Housel, and K. Li. Automatic alignment of high-resolution multi-projector display using an un-calibrated camera. In *IEEE Visualization*, pages 125–130, 8-13 Oct. 2000.

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

[7] T. Funkhouser and K. Li. Special issue: Large format displays. *IEEE Computer Graphics and Applications*, 20(4), July/Aug 2000.

[8] G. Godin, J.-F. Lalonde, and L. Borgeat. Projector-based dual resolution stereoscopic display. In *IEEE Conf. on Virtual Reality 2004*, pages 223–224, 27-31 March 2004.

[9] G. Godin, P. Massicotte, and L. Borgeat. Foveated stereoscopic display for the visualization of detailed virtual environments *(accepted for publication)*. In *Eurographics Workshop on Virtual Environments 2004*, 8-9 June 2004.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[11] Intel Open Source Computer Vision Library, http://www.intel.com/research/mrl/research/opencv/, (accessed Jan. 2004).

[12] W. Kresse, D. Reiners, and C. Knöpfle. Color consistency for digital multi-projector stereo display systems: the HEye-Wall and the digital CAVE. In *9th Eurographics Workshop on Virtual Environments 2003*, May 2003.

[13] K.-L. Low, A. Ilie, A. G. Welch, and A. Lastra. Combining head-mounted and projector-based displays for surgical training. In *Proc. IEEE Virtual Reality 2003*, pages 110–117, 22-26 March 2003.

[14] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proc. Visualization 2000*, pages 117–124, 2000.

[15] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide*. Addison Wesley, 1996.

[16] R. Raskar. Immersive planar display using roughly aligned projectors. In *Proc. of IEEE Virtual Reality 2000*, pages 109–115, 18-22 March 2000.

[17] A. Yoshida, J. Rolland, and J. Reif. Design and applications of a high resolution insert head-mounted-display. In *Virtual Reality Annual Int. Symposium '95*, pages 84–93, 11-15 March 1995.