

A Personal Surround Environment: Projective Display with Correction for Display Surface Geometry and Extreme Lens Distortion

Tyler Johnson, Florian Gyarfas, Rick Skarbez, Herman Towles and Henry Fuchs*

University of North Carolina at Chapel Hill

ABSTRACT

Projectors equipped with wide-angle lenses can have an advantage over traditional projectors in creating immersive display environments since they can be placed very close to the display surface to reduce user shadowing issues while still producing large images. However, wide-angle projectors exhibit severe image distortion requiring the image generator to correctively pre-distort the output image.

In this paper, we describe a new technique based on Raskar's [14] two-pass rendering algorithm that is able to correct for both arbitrary display surface geometry and the extreme lens distortion caused by fisheye lenses. We further detail how the distortion correction algorithm can be implemented in a real-time shader program running on a commodity GPU to create low-cost, personal surround environments.

Keywords: Projector displays, lens distortion correction, GPU programming.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

1 INTRODUCTION

Since the 1970s and the creation of the earliest flight simulators, the best technology for building large, immersive environments has been the projector. Over the last 10 years with the advent of very affordable commodity projectors, display research has exploded with significant focus being placed on developing higher fidelity, more robust, and visually seamless tiled projective displays. From rear-projected to front-projected systems, we have seen a great number of innovative geometric and photometric calibration and rendering methods that have stimulated the creation of many new higher quality, large-scale projective environments.

This ability to create large, immersive environments has lead researchers to investigate the use of projector displays in virtual reality. The CAVETM[4] showed that projectors could provide immersive binocular imagery to a tracked user, making the display cube a viable alternative to head-mounted displays in some applications. Other researchers [10] have experimented with combining projectors and head-mounts to form a hybrid display.

Recent work in projector displays has focused on liberating the projector from its traditional role of displaying only onto a single white-diffuse, planar display surface. Raskar [12, 14, 13], Bimber [2], et al. have described general techniques for pre-warping projected imagery to account for arbitrary display surface shape. Raskar's two-pass rendering algorithm takes ideal undistorted imagery and re-maps it onto a 3D model of the display surface such that the resulting (pre-distorted) projected imagery appears undistorted and perspective-correct to a viewer at a known location.

*tmjohns, florian, skarbez, towles, fuchs@cs.unc.edu



Figure 1: An immersive display built with a single fisheye projector.

These innovations and others have given us the ability to create ad-hoc on-demand displays using surfaces of arbitrary shape.

It should be noted that most of this research has focused on using projectors with standard lenses that can be optically represented with a simple linear pinhole (perspective projection) lens model. In most such systems today, any lens distortion is often just minimized by adjusting the zoom focal length or simply assumed to be very small, which is very often a reasonable assumption.

With the goal of building a personal immersive environment for use in small rooms, our research group has been experimenting with an Epson 715c projector equipped with a custom fisheye lens (Elumens Corporation). Figure 1 shows perspective-correct imagery from a single fisheye projector displayed on three walls in our lab to create a compact and inexpensive immersive environment.

Fisheye projectors have the advantage of being able to create very large display areas even when the projector is very close to the display surface. In front-projection configurations, this wide-angle capability may allow the projector to be positioned between the viewers and the display surface, which has the added advantage of eliminating any user shadowing problems. However, unless the display surface geometry perfectly counters the severe distortion of the fisheye lens, the rendering system for such configurations must pre-distort the image to correctly map it onto the display surface for the viewer.

Wide-angle lens projectors have been used before to create wide field-of-view dome displays with a single projector and in some special applications. The Elumens VisionStation[®] uses a wide field-of-view projector and a specialized display screen to provide an immersive experience to a stationary user. Konieczny et al. [9] use a fisheye lens projector to allow a user to explore volume data by manipulating a tracked sheet of rear projection material. In contrast to these previous approaches, our method does not require a

specialized display screen and can be applied to surfaces of arbitrary shape. This new flexibility can eliminate the substantial cost associated with specialized display surfaces.

In this paper, we describe how Raskar’s two-pass rendering algorithm, which corrects for arbitrary display surface geometry and allows head-tracked users, can be extended to incorporate correction for the extreme lens distortion introduced by wide-angle lens projectors. We show that the obvious extension of adding an additional lens distortion correction pass cannot make use of the full field-of-view of the projector without introducing strong aliasing artifacts. Our new technique for incorporating lens distortion correction that does not suffer from these aliasing artifacts and does not introduce an additional rendering pass is then described. Finally, we demonstrate perspectively correct results using the fisheye-lens projector displaying into a room corner.

2 BACKGROUND

In this section, we describe the original two-pass image correction algorithm for multi-projector displays that we extend and give some background on lens distortion.

2.1 Two-Pass Multi-Projector Image Correction

Raskar describes in [12, 14] an algorithm to correct for image distortion in projector displays resulting from projecting onto arbitrary display surface geometry. The method is based on a two-pass algorithm where the path of light from the projector to the viewer is simulated in order to determine the image that must be projected for the viewer to observe a desired image. This process requires knowledge of the geometric relationship between the surface and projectors in the display, as well as a known viewing position in a common coordinate system and assumes no distortion by the projector lens.

2.1.1 Geometric Calibration

Projector and display surface calibration is accomplished by an up-front calibration process where structured light patterns are projected by each projector in sequence and observed by a pre-calibrated stereo camera pair. The structured light patterns allow precise image correspondences between the cameras and projectors to be established. Using this correspondence information, the display surface geometry is reconstructed in the coordinate system of the cameras via stereo triangulation to produce a 3D point cloud from which a polygonal model is extracted. A projection matrix for each projector is then calculated using correspondences between projected 2D locations in the structured light patterns and reconstructed 3D locations on the display surface.

2.1.2 Correction

Image correction is performed in two rendering passes where in the first pass, the desired image to be observed by the viewer is rendered to texture by the application. In the second pass, the display surface model is rendered with projective texturing where the texture matrix frustum originates at the viewer’s location and overlaps the illuminated area of the projector on the display surface. This matrix assigns texture coordinates in the desired image to the vertices of the display surface model. The calibrated projection matrix of the projector is then used to render the textured display surface model, producing the image that must be projected to provide the viewer with the desired image.

2.2 Lens Models and Distortion

The lens of a camera or projector affects the path that light travels as it enters or exits the device. If the device is to be used in a geometric application such as stereo triangulation, it must be calibrated to take into account the behavior of the lens. The most common lens model for cameras and projectors is the pinhole perspective model. In this

model, straight lines remain straight as they pass through the lens and devices with lenses adhering to the model are termed *linear*.

Linear optical devices have 11 degrees of freedom that can be represented as a 3×4 projection matrix P , defined up to scale, which relates a world point $X = [X, Y, Z, 1]^T$ to its imaged pixel location $x = [u, v, s]^T$ through the linear equation

$$x = PX. \quad (1)$$

Similarly, a pixel x can be related to the ray r along which it would travel if projected by the device via

$$r(\alpha) = C + \alpha(KR)^{-1}x, \quad (2)$$

where we have used the decomposition of P into its intrinsic (K) and extrinsic parameters (R, t) such that $P = K[R|t]$. The $[R|t]$ matrix transforms a world point X into the coordinate system of the device and performs the initial projection of X onto the image plane, where the principal point lies at the origin. The K matrix then transforms points on the image plane to pixel coordinates in the camera or projector image by applying the focal properties of the lens. $C = -R^{-1}t$ is the center of projection of the device in world coordinates.

2.2.1 Distortion Models

Lens distortion is usually considered any deviation exhibited by a lens that is inconsistent with the pinhole perspective model. While lens properties can vary drastically, no physical lens is a perfect pinhole and straight lines in the world will be distorted to a greater or lesser extent by the lens. In practice, cameras are often treated as pinhole devices along with a distortion model that attempts to compensate for deviations from the ideal pinhole model.

Since lens distortion affects where world points are imaged, or in the case of projectors, the direction of the ray along which a pixel is projected, estimation of lens distortion in a device can significantly improve calibration results, especially for wide-angle lenses. The most important type of distortion is radial, which increases with distance from the center of distortion in the image. The center of distortion is usually located at or near the principal point. In general, the amount of radial distortion is inversely proportional to the focal length of the lens.

In the Computer Vision literature [6, 7], camera lens distortion is typically modeled as a process that occurs after the projection of a world point onto the image plane of the camera. Using the decomposition of $P = K[R|t]$, distortion is incorporated into the projection process as

$$x = K\delta_{in}([R|t]X). \quad (3)$$

The δ_{in} operator remaps homogeneous 2D points after their initial projection onto the image plane to model deviations from the pinhole model for light *entering* the lens. This operator will necessarily be non-linear in order to capture non-linearities in the projection process not modeled by the pinhole projection model of Equation (1). Since each homogeneous 2D point on the image plane is the projective equivalent of a ray, this function can also be thought of as operating on rays as they enter the lens. An image plane example of a δ_{in} operator distorting an image border is depicted in Figure 2. The operation depicted in the figure is referred to as a pincushion distortion.

A technique developed by Brown [3], which has been adopted by the Matlab Camera Calibration Toolbox and Intel’s OpenCV, models both radial and decentering lens distortion, where the latter arises from imperfectly aligned lenses. Decentering distortion possesses both radial and tangential components. For the Brown model with two coefficients for radial distortion (k_1, k_2) and two for tangential (p_1, p_2), the distortion operator δ_{in} is

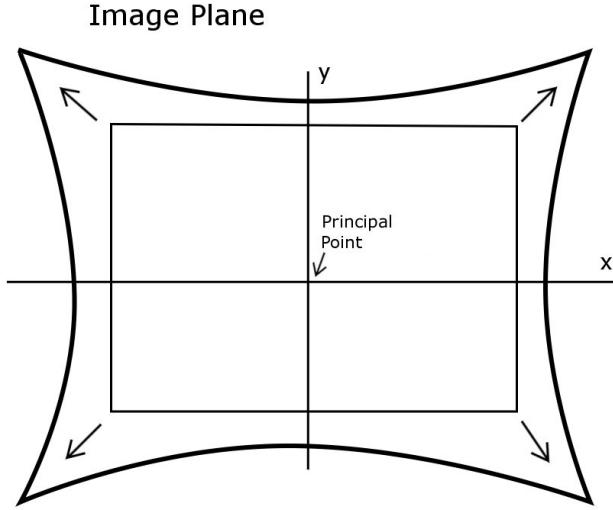


Figure 2: Example effect of lens distortion on the image plane.

$$\delta([u, v, s]^T) = \begin{bmatrix} x(1 + k_1 r^2 + k_2 r^4) + 2p_1 xy + p_2(r^2 + 2x^2) \\ y(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y^2) + 2p_2 xy \\ 1 \end{bmatrix}, \quad (4)$$

where $x = u/s$, $y = v/s$ and $r^2 = x^2 + y^2$.

While this model is not directly invertible, the set of coefficients that invert a specific distortion can be determined using a non-linear optimization based on correcting a set of distorted sample points in the image plane of the device to their original positions [7].

2.2.2 Non-Pinhole Lens Models

Lenses such as fisheye lenses represent a strong departure from the pinhole model and have a different representation. For example, our experimental wide field-of-view projector is equipped with a fisheye lens having a 180° field-of-view. In the pinhole perspective model, the angle θ in radians between an incoming ray and the principal ray is related to the focal length f in pixels and the distance r between the principal point and the incoming ray's pixel by $r = f \tan \theta$. While there are a number of different fisheye lenses with unique properties, our projector lens is of the most common type, an equidistance projection or f-theta lens where $r = f\theta$.

We treat non-pinhole lens models in the same way that lens distortion is treated for lenses using the pinhole model - as functions operating on homogeneous 2D points on the image plane. Given this framework, δ_m for an f-theta lens can be expressed as

$$\delta([u, v, s]^T) = \begin{bmatrix} \arctan(r) * x \\ \arctan(r) * y \\ r \end{bmatrix}, \quad (5)$$

where $x = u/s$, $y = v/s$ and $r = \sqrt{x^2 + y^2}$. The focal length has been omitted from the above equation since it is equal to one on the image plane before the intrinsic matrix has been applied. For the f-theta model, δ_m can also be directly inverted to produce δ_{out} , the distortion that light undergoes as it exits the device. Note that since we treat the f-theta model as a distortion within the pinhole model, a singularity exists for $\theta = \pm\pi$ radians. This is of little consequence in practice since the field of view can be limited to slightly less than 180° to avoid the singularity.

2.2.3 Lens Distortion Correction

When lens distortion correction is performed on camera images, the goal is to take a distorted image captured by the device and use the distortion model to produce the undistorted image that would have been taken by a perfect pinhole camera. This process can be performed in one of two ways. Either we color the undistorted image at each pixel by sampling from the captured distorted image, or we attempt to splat each pixel of the captured distorted image onto the pixels of the undistorted image in some way. The first technique requires calculation of δ_{in} , while the second would require calculation of δ_{out} . Since a sampling procedure is typically preferred, when the distortion model is not easily inverted, lens distortion properties for cameras are usually calibrated in a way that allows the calculation of δ_{in} .

If δ_{in} is known, a pixel $p = [x, y, 1]^T$ in the desired undistorted image will map to a pixel $p' = K\delta_{in}(K^{-1}p)$ in the captured distorted image. The captured image can then be filtered around p' to produce a color for p in the undistorted image.

2.2.4 Lens Distortion Estimation

The direct linear transformation (DLT) technique [1, 5] allows the projection matrix of a device to be computed from a set of 3D-2D point correspondences. This method, however, cannot be extended to calibrate models including non-linear effects such as lens distortion. The solution that is commonly employed in this circumstance is to first perform a DLT calculation to obtain a projection matrix which approximates the linear behavior of the device. This projection matrix is then used as an initial guess in a non-linear optimization that estimates the parameters of the full model to minimize the sum of squared reprojection errors.

If $P = K[R|t]$ is the output of DLT and $X_{1..N}$ is a set of 3D points with a known set of 2D correspondences $x_{1..N}$, the non-linear technique should minimize

$$\sum_{i=1}^N \text{dist}(K\delta_{in}([R|t]X_i), x_i))^2. \quad (6)$$

3 TWO-PASS IMAGE CORRECTION FOR WIDE-ANGLE LENSES

The basic two-pass multi-projector image correction technique reviewed in Section 2.1 works well under the condition that the geometric properties of the display devices do not deviate significantly from the pinhole perspective model. Significant deviations from this model can result in obvious visual artifacts such as ghosting in projector overlap regions and miscorrected imagery. We next describe how we have extended Raskar's original technique to incorporate correction for projector lens distortion and non-pinhole lens models.

3.1 Modified Projector Calibration

To incorporate lens distortion into the geometric model of a projector, we continue Raskar's original treatment of the projector as the dual of a camera. In Section 2.2 we described some lens and distortion models commonly used for cameras. We use these same models for projectors, but modify the calibration process slightly to account for projector/camera duality.

In contrast to cameras, for projectors the light we are interested in travels outward from the device, making it of greatest practical value to calibrate for projector lens distortion in a way that allows the calculation of δ_{out} . This models how light is distorted as it exits the lens and allows the back-projection of each projector pixel into a ray in world space in a manner that accounts for distortions in non-pinhole lenses.

Our projector calibration process is identical to that described in Section 2.2.4 for calibrating cameras with lens distortion, except

that we replace the minimization function for cameras in Equation (6) with

$$\sum_{i=1}^N \text{dist}(PX_i, K\delta_{out}(K^{-1}x_i))^2. \quad (7)$$

Here we have distorted the projected feature locations $x_{1..N}$ at each iteration using δ_{out} , which can model either a pinhole lens with radial and tangential distortion characteristics or a non-pinhole lens model. For models such as the Brown model that are not easily invertible, this will yield distortion coefficients allowing us to pre-distort an image before projection so that the projector becomes a linear device.

3.2 Modified Geometric Correction

In the previous section, we described how a projector can be calibrated in a way that allows compensation for lens distortion by pre-distorting an image before projection. Given such a calibration, it would be a simple process to add an additional third pass to the two-pass correction algorithm described previously to perform this operation. To determine the color for a pixel p in the desired compensation image, we can filter the image rendered during pass two around the pixel $p' = K\delta_{out}(K^{-1}p)$.

The problem with this technique when using wide-angle lenses is that rendering in pass two will not generate enough imagery to fill the field-of-view of the projector after distortion correction. This occurs because the pass two image will be rendered with the pinhole perspective model using the intrinsics of the wide-angle lens. This greatly reduces the field-of-view since a wide-angle lens has a much greater field-of-view than a pinhole lens of the same focal length. One solution to this problem would then be to calculate the set of intrinsics that a pinhole lens would require to have a field-of-view comparable to the projector's wide-angle lens.

Unfortunately, for extremely wide-angle lenses, such as fisheye lenses, this technique has the downside of introducing aliasing artifacts. The reason for this is illustrated in Figure 3, which was created using actual data from our experimental fisheye projector. The rectangle situated at the origin represents the border of an image given to the projector on the image plane before it is distorted by the lens. This is the same as the region of the image plane that the K matrix of the projector will transform to valid pixel coordinates in the projector image. The contour surrounding the image depicts the extent to which the borders of the image are distorted by the f-theta fisheye lens of our experimental projector when the field-of-view is limited to 178° .

If 178° of the horizontal projector field-of-view is to be filled after distortion correction, the new intrinsics K' must transform a region enclosing the entire distorted contour into valid pixel coordinates. Since K is an upper-triangular matrix, the region of valid pixel coordinates must form a parallelogram on the image plane. Clearly, if such a region is to enclose the distorted border, the pixels of the pass-two texture must be stretched over a much larger spatial extent, leading to a low-pass filtering of the pass-one texture and aliasing during distortion correction due to large changes in pixel density over the extent of the distorted region. Also, those pixels not falling within the convex hull of the distorted contour are effectively wasted since no pixels in the compensated image will map to their location.

While increasing the resolution of the textures rendered during passes one and two of the pipeline can reduce the amount of aliasing, we have found that it remains significant up to the maximum texture size that current hardware is able to render. Figure 4a depicts the aliasing that results when this approach is used to generate a 178° field-of-view image using our fisheye projector. Figure 4b shows that significant aliasing is still present when rendering in passes one and two is performed at 4x projector resolution(4096x3072).

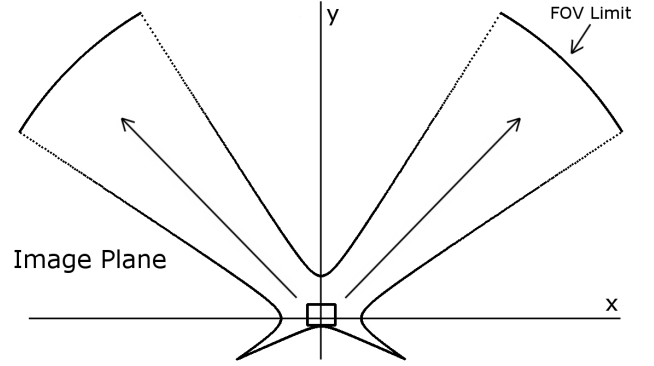


Figure 3: Distortion resulting from fisheye projector lens.

3.2.1 A Better Approach

In this section, we describe our new two-pass rendering solution, which eliminates the aliasing concerns that a three-pass technique can introduce. The objective is to be able to simulate a non-pinhole rendering model in pass two, eliminating the need for a third pass lens distortion correction step.

In pre-process, we use the projector calibration, including distortion properties, in conjunction with the display surface model to determine the 3D location on the display surface that each projector pixel illuminates. Given the projector calibration $P = K[R|t]$ and δ_{out} , each pixel $x_i = [u_i, v_i, 1]^T$ of the projector is back-projected to produce a ray

$$r(\alpha) = C + \alpha R^{-1}\delta_{out}(K^{-1}x_i). \quad (8)$$

This ray is then intersected with the polygons of the display surface model yielding a 3D point on the display surface. This process is repeated until the mapping has been performed at the resolution of the projector.

Using this 2D-3D mapping, we can correct for both the display surface geometry and lens distortion in the second pass by projecting each projector pixel's 3D location into the pass-one texture to produce the pixel's output color. If both the projector and display surface remain static during display, this 2D-3D mapping will remain fixed even though the position of a head-tracked viewer may change.

Graphics card vendors now produce consumer cards with floating-point pipelines that allow the use of textures consisting of four 32-bit floating-point values per pixel. We use this technology to store the projector's 2D-3D mapping directly on the graphics card. A floating-point texture is created at the resolution of the projector where the floating-point location (x, y, z) on the display surface that a pixel illuminates is stored as its (r, g, b) elements in the texture. The alpha component of each pixel in the texture can also conditionally be used as a flag to indicate that the pixel should be left black. This is useful when the geometry of the display surface model does not fill the entire field-of-view of the projector.

At render time, a pixel shader takes as input the floating-point texture of display surface geometry, the desired image from pass one and the viewing matrix of the viewer modified to act as a texture matrix. The shader simply looks up the vertex information for the pixel it is currently shading and projects the vertex into the pass-one texture using the texture matrix to produce an output color. Our GPU implementation allows correction to take place at interactive framerates.



Figure 4: a) Three-pass correction for display surface geometry and fisheye distortion. b) Three-pass correction super-sampled 4x. c) Our two-pass method without super-sampling.

3.3 Modified Edge Blending

In addition to the two-pass algorithm to correct for image distortions due to arbitrary display surfaces, [12] also describes a simple technique for doing edge blending in multi-projector displays, which eliminates areas of increased brightness where projectors overlap. The basic idea is to compute an alpha mask for each projector that gives the attenuation value to apply at each pixel in order to blend smoothly between overlapping projectors.

The alpha masks are computed by observing projector overlap regions with a camera. Attenuation values are then computed in the image space of the camera for each projector by taking into account the number of projectors overlapping at each camera pixel and the distance to the convex hull of each projector’s contribution in the camera image. The attenuation value for projector m at camera pixel (u, v) is computed as

$$A_m(u, v) = \frac{\alpha_m(m, u, v)}{\sum_i \alpha_i(m, u, v)}. \quad (9)$$

In the above equation, $\alpha_i(m, u, v) = w_i(m, u, v) * d_i(m, u, v)$ where $w_i(m, u, v)$ is 1 if (u, v) is within the convex hull of projector i and 0 otherwise. The $d_i(m, u, v)$ term is the distance from camera pixel (u, v) to the convex hull of projector i in the camera image. This technique produces weights that sum to one at each camera pixel and decay gradually to zero at projector edges. Since weights are computed in camera image space, each projector’s weights are transformed into its own image space using the two-pass image correction algorithm.

Since the generation of the alpha masks for each projector relies on the use of the two-pass algorithm, the original technique cannot be used to generate accurate alpha masks for projectors that do not fit the pinhole model. We take a slightly different approach to the problem and use the geometric information from calibration, including the lens distortion model of the projector, to produce an alpha mask for each projector without the additional use of a camera required by the original technique. We calculate alpha masks for each projector as outlined in Algorithm 1.

4 RESULTS

Using our previously described calibration and correction techniques for projectors with lenses deviating from the pinhole model, we have explored a number of different display configurations using our experimental fisheye projector. Each display was calibrated by observing projected structured light patterns with a stereo camera pair used to reconstruct the display surface via triangulation. To extract a polygonal display surface model from the reconstructed point cloud, we have used a plane-extraction technique [11] to closely approximate the piece-wise planar display surfaces in our lab.

Algorithm 1 GENALPHAMASKS

```

1: for each projector  $i$  do
2:   for each pixel  $j$  of projector  $i$  do
3:      $r \leftarrow$  back-project  $j$  using Equation (8)
4:      $X \leftarrow$  intersect  $r$  and display surface
5:      $sum \leftarrow 0$ 
6:     for each projector  $k \neq i$  do
7:        $x \leftarrow$  project  $X$  using Equation (3)
8:       if  $x$  within displayable area then
9:          $sum \leftarrow sum + \min$  dist to projector  $k$ ’s image border
          at  $x$ 
10:      end if
11:    end for
12:     $m \leftarrow$  min dist to projector  $i$ ’s image border at  $j$ 
13:     $A_i[j] = \frac{sum}{sum+m}$ 
14:  end for
15: end for

```

Figure 5 depicts our fisheye projector being used in a virtual reality application using a multi-wall surface in our lab for display. An overhead view of this configuration is provided in Figure 7, which clearly shows the close proximity of the projector to the display surface. The panorama of Figure 11 shows the view from a user’s position very near the projector. This illustrates the nearly 180° immersive environment created by our fisheye projector. Compare this with Figures 8 and 9 where a conventional projector was used, showing the difference in the size of the imagery that is produced.

The panorama of Figure 10 was taken of our fisheye projector displaying on this same multi-wall surface without any consideration for lens distortion. Note how the imagery is warped by the lens, prohibiting proper correction for the distortion due to the display surface geometry. Contrast this with the quality of the correction we are able to achieve in Figure 11 using our algorithm. Any apparent distortion due to the lens has been removed and distortions due to the display surface geometry are also well corrected. The correction does have a slight flaw in the right side of the image where there is a 1-2 pixel error in the correction for the corner geometry. We think this may be due to the fisheye lens of our projector deviating slightly from its f-theta model, which we plan to account for in future work.

Figure 4c is a close-up of our correction method, which shows that our technique is able to correct for the image distortion introduced by both the display surface geometry and the fisheye lens without introducing the aliasing artifacts present in Figures 4a and 4b.

As an illustration of the general applicability of our method, we combined our fisheye projector with a conventional projector

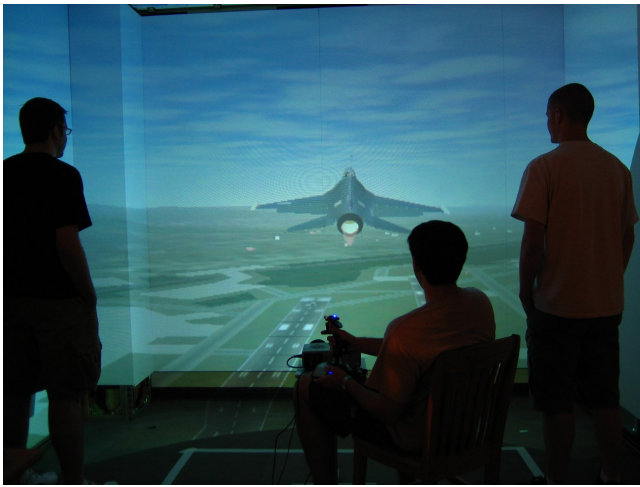


Figure 5: Fisheye projector used in a flight simulator application.

to form a multi-projector display. The conventional projector was calibrated using the Brown distortion model while the fisheye projector was calibrated with the f-theta model. The resulting display is depicted in Figure 6. For this display, we have used our modified edge blending algorithm to blend between the two projectors. Unfortunately, the algorithm currently does not take into account differences in pixel density or black and white levels between the projectors, resulting in some edges being only softened.



Figure 6: Display system combining a conventional projector and a fisheye-lens projector.

5 CONCLUSIONS AND FUTURE WORK

We have demonstrated using camera-based calibration, how a single fisheye-lens projector can be used to create a personal immersive display system in an ordinary room without the need for a specialized display screen. This allows images several times larger than that of a conventional projector to be generated at the same distance from the display surface, making it possible for viewers to stand close to the display surface without shadowing projected imagery.

To correct for the distortion introduced by the fisheye lens, we have extended a previously existing image correction technique for multi-projector displays that supports a head-tracked viewer. Our

extended method is able to incorporate both conventional projectors with slight lens distortion characteristics and non-conventional fisheye-lens projectors with extreme distortion into a single display without sacrificing support for dynamic viewer tracking. Using programmable commodity graphics cards, this technique is able to take advantage of the extremely large field-of-view afforded by fisheye lenses without introducing undesired aliasing artifacts that can occur when performing lens distortion correction.

While fisheye-lens projectors can be used to create immersive displays at close proximity to a display surface, they can suffer from loss of brightness near the periphery of a projected image. Also, conventional projector lenses may be better suited when the spatial resolution of projected imagery is favored over its size, since a fisheye lens will spread the resolution of the device over a much larger field-of-view.

Even though our correction algorithm allows fisheye lens projectors to be used in multi-projector displays without the introduction of aliasing, there are additional aliasing issues that we still plan to address. Since a projection matrix is used to texture the desired image onto the display surface model in pass two, as the viewer approaches the display surface, the field-of-view of the frustum that must be used to texture the entire display surface geometry may approach 180° . This can lead to substantial aliasing artifacts in the corrected image. An existing solution to this problem that we plan to implement is to render multiple views from the viewing position in different directions during pass one.

Also, when filtering the desired image during pass two, we plan to extend our method to take into account the complex ways in which projector pixels may overlap with pixels of the desired image after they are projected onto the display surface model. Currently we use the basic bi-linear filtering approach supported by the graphics hardware, but ideally the desired image would be filtered using an area-weighted sampling technique that is not limited to four pixels.

Our work could also benefit from the use of more general lens models such as [8], which allows both pinhole and fisheye lenses to be modeled in the same way. This would also allow us to model deviations of fisheye lenses from their associated lens model, something we have not yet attempted, and make it possible to utilize a field-of-view larger than 180° .

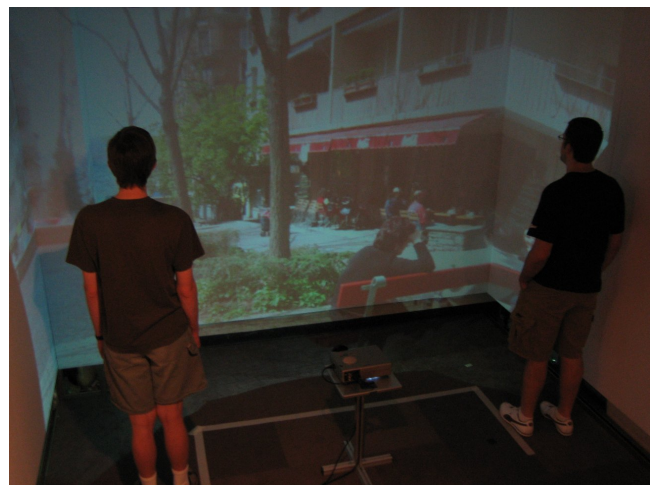


Figure 7: Overview of our immersive display set-up.



Figure 8: Relative image size of a single conventional projector.

ACKNOWLEDGEMENTS

The authors wish to thank D' nardo Colucci of Elumenati LLC, who generously loaned us the fisheye projector used in this research. We would also like to thank Greg Welch for his insightful contributions.

This research was primarily supported by the DARPA DARWARS Training Superiority and DARPA VIRTE (Virtual Technologies and Environments) programs under the Office of Naval Research award number N00014-03-1-0589.

REFERENCES

- [1] Y. Abdel-Aziz and H. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In *Symposium on Close-Range Photogrammetry*, pages 1–18, 1971.
- [2] O. Bimber, G. Wetzstein, A. Emmerling, and C. Nitschke. Enabling view-dependent stereoscopic projection in real environments. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, pages 14–23, 2005.
- [3] D. Brown. Close-range camera calibration. *Photometric Engineering*, 37(8):855–866, 1971.
- [4] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *ACM SIGGRAPH*, 1993.
- [5] O. Faugeras and G. Toscani. Camera calibration for 3d computer vision. In *International Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, pages 240–247, 1987.
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.
- [7] J. Heikkil and O. Silv. A four-step camera calibration procedure with implicit image correction. In *CVPR*, 1997.
- [8] J. Kannala and S. Brandt. A generic camera model and calibration method for conventional, wide-angle and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):451–460, August 2006.
- [9] J. Konieczny, C. Shimizu, G. Meyer, and D. Colucci. A handheld flexible display system. In *IEEE Visualization*, 2005.
- [10] K.-L. Low, A. Ilie, G. Welch, and A. Lastra. Combining head-mounted and projector-based displays for surgical training. In *IEEE Virtual Reality*, 2003.
- [11] P. Quirk, T. Johnson, R. Skarbez, H. Towles, F. Gyarfas, and H. Fuchs. Ransac-assisted display model reconstruction for projective display. In *Emerging Display Technologies*, 2006.
- [12] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, W. B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *IEEE Visualization*, pages 161–168, 1999.
- [13] R. Raskar, J. van Baar, T. Willwacher, and S. Rao. Quadric transfer for immersive curved screen displays. In *Eurographics*, 2004.

- [14] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188, New York, NY, USA, 1998. ACM Press.



Figure 9: Imagery produced by a conventional projector.



Figure 10: Imagery produced by a fisheye lens projector without considering lens distortion.



Figure 11: Imagery produced by a fisheye lens projector using our correction algorithm.