Improved Pre-Warping for Wide Angle, Head Mounted Displays

Daniel Pohl[‡]
Intel Corporation

Greg Johnson
Intel Corporation

Timo Bolkart
Saarland University

ABSTRACT

High-quality head mounted displays are becoming available in the consumer space. These displays provide an immersive gaming experience by filling the wearer's field of view. To achieve immersion with low cost, a commodity display panel is placed a short distance in front of each eye, and wide-angle optics are used to bring the image into focus. Unfortunately, these optics introduce spatial and chromatic distortion into the image seen by the viewer. As a result, the images to be displayed must be pre-warped to cancel this distortion. Pre-warping is typically performed in a post-processing step using a pixel shader. However, this discrete resampling leads to a loss in image quality. Here, we propose a pixel shader with reduced error than existing methods. We also examine in-camera distortion correction which further reduces error by avoiding resampling.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture / Image Generation

1. INTRODUCTION

The availability of compact, inexpensive, and high-quality displays for mobile devices has motivated interest in consumer oriented head mounted displays (HMDs) for immersive gaming. It has been shown that commodity display panels can be combined with inexpensive optics (Figure 1) to achieve immersivity with low cost [5, 4, 6]. Unfortunately, these optics introduce spatial and chromatic aberrations which can be difficult to accurately correct.

For example, spatial distortion due to the lens is visible as a pincushion effect (Figure 2b). This effect can be canceled by pre-warping the image presented on the display panel, with the corresponding barrel distortion (Figure 2c). Such pre-warping can be performed by resampling the image in a post-processing step [1], but at the cost of sampling artifacts.

[‡]daniel.pohl@intel.com

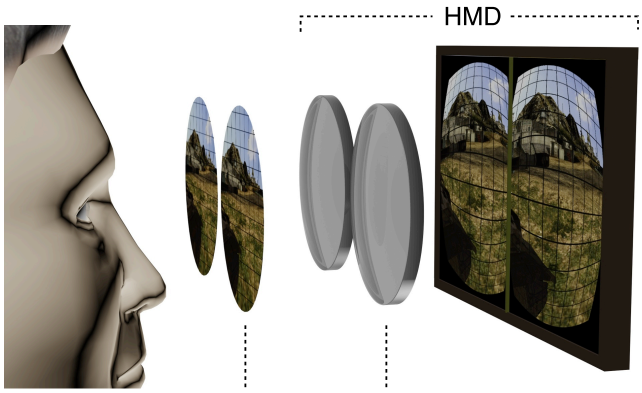


Figure 1: *A low-cost immersive HMD. A commodity display panel is placed a short distance from the eye in order to fill the viewer's field of view, and a wide-angle lens is used to bring the image into focus. Since the lens introduces spatial and chromatic distortion, an image presented on the display must be pre-warped and color adjusted to counter this distortion, so that the image arriving at the retina appears correct.*

In this paper, we evaluate the image quality and performance of image-space and object-space methods for correcting spatial and chromatic distortion in modern HMDs. We show that it is possible to reduce resampling error with minimal impact on performance (compared to a commonly used approach) when distortion correction is performed in image space. We also show an accurate object-space distortion correction method by simulating a barrel distortion in-camera.

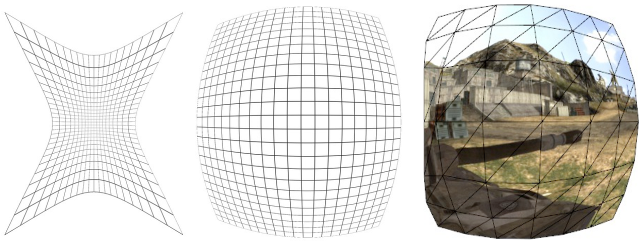
2. RELATED WORK

Spatial distortion in an optical system results from both the shape and material of the lens. This distortion (and its correction) can be described using a Taylor series [1]. A small number of terms is sufficient to capture the barrel distortion seen in Figure 2b. Displacement from the optical axis of the camera (r_{new}) is computed according to Equation 1. Coefficients k_i control the degree of distortion.

$$r_{new} = r\left(k_0 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \quad (1)$$

Color fringing artifacts occur when light of different wavelengths refracts differently through a lens [2]. This chromatic aberration can be corrected by separately resampling the red, green, and blue color channels of an image according to Equation 2. The constant terms c_i are often available from the lens manufacturer.

$$r_{new}^{RGB} = \left[\, r_{new}^{R}\left(c_0 + c_1 r^2\right), r_{new}^{G}, r_{new}^{B}\left(c_2 + c_3 r^2\right) \right] \quad (2)$$



(a) (b) (c)

Figure 2: *Wide angle lenses used in modern HMDs produce a pincushion effect (a) which can be canceled by applying the corresponding barrel distortion (b) to images presented on the display panel. This correction can be applied in image space by resampling the image according to Equation 1 or by texture mapping the image onto a mesh in which the vertices have been displaced according to the same equation.*

Spatial and chromatic distortion correction can be performed in image space or object space.

2.1 Image Space Solutions

We look at image space solutions using distortion meshes and image warping to compensate distortions. One way of presenting an image on the screen is to draw it onto geometry consisting of a screen-filling quad. In the case of using a distortion mesh to compensate the radial distortions geometry is generated which approximates a barrel distortion by moving the vertices according to Equation 1.

Another way of handling the radial distortion is to apply image warping in a post-processing pixel shader [1]. The framebuffer texture is rendered onto a screen-filling quad while using a pixel shader. The shader gets for every pixel the input texture coordinates from which it would usually access the framebuffer texture. Now the shader will transform these according to Equation 1, therefore drawing the image in a barrel-distorted way. In order to minimize chromatic aberration a pixel shader is used that warps the image according to Equation 2. However, warping causes resampling artifacts which manifest in a loss of high frequency details and sharpness: for the new sample point there is usually no accurate information about its color in the original image and one has to guess the right value through interpolation (e.g. interpolation between white and black results in gray). The default implementations of image space solutions against distortions are using bilinear-filtered texture access. In order to achieve better results for warping one modification is to oversample the original image at higher resolution to have more sample points available.

2.2 Object Space Solutions

We look at object space solutions on how to compensate spatial distortions using a vertex and a tessellation shader. Instead of warping the pixels after rendering another approach is to have the scene geometry warped during rendering [7]. In a standard rasterizer this can be done using a vertex shader. After the geometry has been transformed to unit cube coordinates it can be further transformed into the lens coordinate system to apply Equation 1. Using the new value the vertex is transformed back into unit cube coordinates and is drawn. A limitation is that content often is of too low detail to approximate a barrel distortion accurately. What should be a curve after warping is approximated by lines (Figure 3(c)). A tessellation shader helps by adding new vertices, however, it is hard to know in advance how many vertices will be required to get good enough results (Figure 3(d)) and performance will be highly impacted depending on the tessellation.

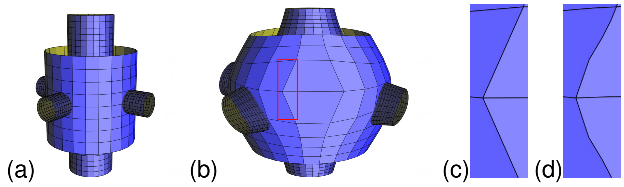


Figure 3: *Barrel distortion applied using vertex or tessellation shaders. A simple geometric model (a) is distorted with a vertex shader (b). A close-up (c) shows that a tessellation shader can more accurately represent the distorted shape by adding vertices (d). Figure courtesy of Philip Rideout [7].*

3. IMPROVED DISTORTION COMPENSATION

We show improvements for image and object space corrections against distortions.

3.1 Improved Image Space Solutions

We investigate an improved approach of image space correction and introduce a new way of handling chromatic aberration. While the previously described pixel shaders to warp the image were using bilinear texture-lookups we investigate instead using bicubic texture interpolation [3], which generates a better approximation towards the original signal, therefore keeping more sharpness.

A new method to compensate chromatic aberration is using distortion meshes. We render the first mesh with only the blue channel of the texture. With a small offset in front of it we render the next mesh with only the green channel, setting the alpha value to transparent, while setting the blend mode to additive blending. We repeat the last step for red.

3.2 Improved Object Space Solutions

We present improved object space solutions against spatial and chromatic distortions. As vertex and tessellation shaders are not sufficient we propose instead to do an exact barrel-distorted sampling of the scene. This requires that the sampling pattern for rendering can be freely adjusted as in a software rasterizer, voxel ray caster or a ray tracer. We use the latter with a camera model that instead of shooting rays through the regular perspective grid will use a barrel-distorted grid (Figure 4).

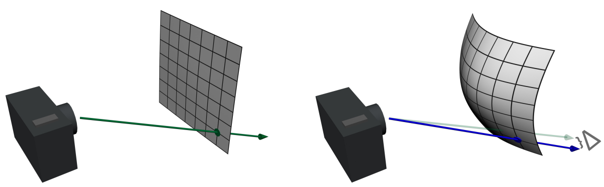


Figure 4: *Camera sampling a pixel. Left: Using regular grid (green ray). Right: Using barrel-distorted grid (blue ray).*

For minimizing chromatic aberration we propose keeping the approximation of considering three color channels instead of the whole visible light spectrum. Again we change from sampling at a fixed grid to a barrel-distorted one using a ray tracer. For every pixel a ray is cast each for red, green and blue according to the scaled radial length as described in Equation 2.

4. EXPERIMENTAL RESULTS

After describing the system we show images using different methods for distortion compensation for subjective comparison. We quantify the difference using the Scharr edge detector [8] and the Structural SIMilarity (SSIM) index [9]. Further we look at oversampling for image space solutions. Next we discuss invalid areas around pre-warped images. Finally we examine the performance.

4.1 System Description

We create side-by-side stereo images in a total resolution of 1280×800 pixels using an experimental in-house ray tracer. Post-processing shaders are applied using GLSL. We use the "island" level from the game *Enemy Territory: Quake Wars*[1]. We use the distortion parameters given by the SDK of the Oculus Rift Developer Kit 1 [1]. We use a Dual-CPU (Intel® Xeon® E5-2687W, 3.1 GHz, 8 physical cores) workstation with a NVidia® GeForce® 680 GTX.

4.2 Subjective Evaluation

We look at solutions regarding spatial distortions. We compare distortion meshes in different triangulations with image-warping using bilinear filtering. Next we compare warping using first bilinear and then bicubic filtering with our in-camera corrected approach. We compare the image and object space solution against chromatic aberration and show the effect of compensating it through the lens of an HMD.

We show the results from rendering onto distortion meshes with 200, $7,200$ and $125,000$ triangles (Figure 5). We compare those to image warping with bilinear filtering. For better visualization of potential artifacts we use a grid texture. We observe that with more triangles it converges towards the solution using image warping with bilinear filtering.

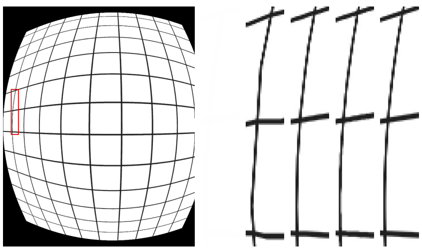


Figure 5: *Barrel-distorted grid using image space solutions. Left to right: Grid rendered on distortion mesh. Close-ups on distortion mesh with 200, 7,200, 125,000 triangles. Close-up using warping with bilinear filtering.*

In Figure 6 we present in (a) the left part of the barrel-distorted stereo image as reference (using supersampling with 32 rays per pixel to minimize aliasing artifacts). We zoom into the marked region to see how this looks photographed through the lens of the Oculus Rift HMD once for image warping using bilinear filtering (b), warping using bicubic filtering (c) and our in-camera correction using a ray tracer (d). We perceive slightly increased sharpness going from bilinear to bicubic-filtered warping. Stepping to our in-camera corrected version the image appears much sharper.

[1]id Software and Splash Damage

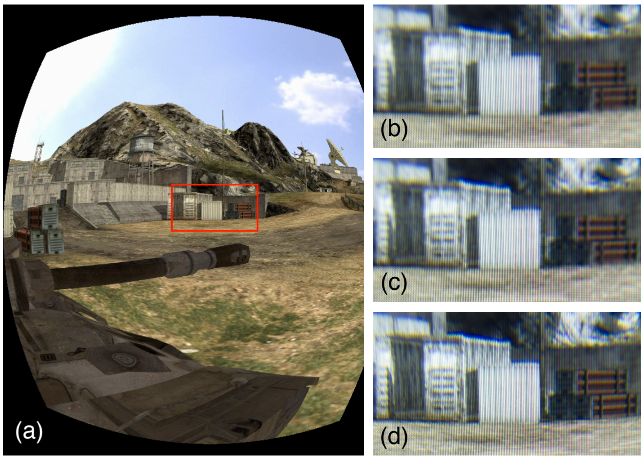


Figure 6: *The image quality of 3 different methods for spatial distortion correction are compared. The highlighted region of the reference image (a) is shown as seen through the lens of an Oculus Rift HMD, for bilinear resampling (b), bicubic resampling (c), and in-camera correction via ray tracing (d).*

We look at results of handling chromatic aberration. In Figure 7 we show the pre-warped image produced by today's default warping approach using bilinear filtering and our accurate in-camera correction. As with handling spatial distortions we see that image warping introduces a loss of sharpness.

Figure 7: *Chromatic distortion correction via image warping with bilinear filtering (left) and in-camera correction (right).*

Finally we show photos through the lens of the Oculus Rift HMD without and with chromatic aberration handling (Figure 8) using our in-camera corrected approach. When enabled, that distortion is less noticeable.

Figure 8: *Seen through the lens of the Oculus Rift. Left: no handling of chromatic aberration. Right: chromatic aberration minimization enabled.*

4.3 Quantitative Measurements Using Scharr

In order to quantize the sharpness we use a standard edge detector, the Scharr operator [8]. In the resulting gray-value image brighter intensities represent more intense edges. As the loss of sharpness is happening for both spatial and chromatic distortions we will focus on the spatial distortions. We are not considering the distortion mesh as it is highly dependent on the number of triangles and converges towards image warping with bilinear filtering. The average intensity values of the Scharr images are 69.79 (warping, bilinear filtering), 74.13 (warping, bicubic filtering) and 77.10 (our accurate in-camera correction). We see a bigger gain in sharper edges going to bicubic filtering than we would have expected from the subjective analysis. As expected our accurate in-camera corrected version has the most detail.

4.4 Quantitative Measurements Using SSIM Index

We quantify image quality using a metric that takes human visual perception under account. We use the Structural SIMilarity Index [9] which compares two images, assuming one of them is of perfect quality and the other has quality degrading properties. Instead of just comparing physical attributes of the image this metric is optimized to deliver values of the perceived image quality. This is based on the assumption that human vision is highly effective at extracting structural information from images. Assuming the value of 1.0 for our accurate in-camera corrected version the results for image warping are 0.94 (bilinear) and 0.95 (bicubic). Those values represent closely what we perceived during our subjective evaluation: the change from bilinear to bicubic is noticeable, but using our in-camera corrected method delivers a much bigger gain in quality.

4.5 Oversampling

A way of improving results of image space solutions is to use oversampling. Instead of rendering the scene at the display resolution a larger framebuffer is used to have more sample points available during warping the image. Rendering at 1920×1200 instead of 1280×800 pixels and applying image warping we get average Scharr values of 74.11 (bilinear) and 77.11 (bicubic). At 2560×1600 pixels the values are 74.76 and 78.07. The Scharr average values of the bicubic versions using oversampling surpass the average value of our accurate in-camera correction at original resolution (77.10), meaning they contain sharper edges. However, resampling into the smaller framebuffer uses linear interpolation methods, while the solutions using a barrel distortion are modeled in a non-linear way.

4.6 Invalid Areas

We discuss the unused areas (15.2%) around the pre-warped image (Figure 6(a)). This limitation happens only for image space solutions because the areas around the pre-warped image would need data that has not been rendered using the undistorted image. Depending on how much the user can see through the HMD the used area can be extended using our accurate in-camera corrected version.

4.7 Performance

We present performance values for compensating distortions. We compare the time to apply spatial distortion compensation after rendering. We include the framebuffer upload as texture. Using distortion meshes consisting up to $7,200$ triangles takes 2.03 ms. Using $125,000$ triangles takes 2.87 ms. Image warping in a pixel shader consumes 2.04 ms (bilinear) and 2.45 ms (bicubic). Handling chromatic aberration using our method with three distortion meshes consisting of $7,200$ triangles per color channel takes 3.21 ms. Using each $125,000$ triangles takes 4.28 ms. Image warping consumes 2.10 ms (bilinear) and 3.04 ms (bicubic).

For the next comparisons we look at the total time of rendering. Using a ray tracer with a regular perspective camera we get 18.68 fps. Applying image warping to the ray traced image results in 18.00 fps (bilinear) and 17.73 fps (bicubic). Using instead our in-camera distortion corrected approach (respecting the invalid areas) the frame rate increases to 20.04 fps, faster than using the image warping methods. In-camera correction of chromatic aberration takes roughly three times longer as there are three times more rays necessary. Our system is not optimized for rendering a ray per color channel.

5. CONCLUSION

In this paper, we have shown improvements towards handling spatial and chromatic aberration. We improved the image space methods to deliver sharper images. We introduced a new method of handling chromatic aberration using distortion meshes. In object space we introduce an accurate in-camera correction for spatial and chromatic distortions that results in sharp images. We compared various approaches in terms of quality and performance.

We suggest for systems that rely on image space correction using warping to include a high quality option using bicubic filtering as it has only minimal performance impact on high-end systems, but delivers sharper images. For systems that are not limited to rendering at a fixed grid we suggest to use accurate in-camera correction as it both increases the quality and performance. Looking ahead we expect that future consumer HMDs will increase the FOV while staying at low-cost which will introduce an even higher amount of distortion. Therefore we expect an increasing impact for the used methods to compensate them.

6. ACKNOWLEDGMENTS

We thank id Software and Splash Damage for their game content. Further thanks to Steffen Hein, Oliver Grau, Audrey Younkin, Philip Rideout and Manfred Ernst.

7. REFERENCES

[1] M. Antonov et al. Oculus sdk overview, 2013. `https://developer.oculusvr.com/`.

[2] E. Hecht. *Optics*. Addison-Wesley, 2002.

[3] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, 1981.

[4] B. Lang. Guide: Build your own oculus rift vr headset, 2013. `http://www.roadtovr.com/2013/03/12/build-your-own-diy-oculus-rift-guide-3927`.

[5] OculusRift, 2013. `http://www.oculusvr.com/`.

[6] J. Olson et al. A design for a smartphone-based head mounted display. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 233–234, 2011.

[7] P. Rideout, 2012. `http://github.prideout.net/barrel-distortion/`.

[8] H. Scharr. *Optimal operators in digital image processing*. PhD thesis, Ruprecht-Karls-University, Heidelberg, 2000.

[9] Z. Wang et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.