

Inset Boundary Smoothing for Foveated Rendering in Virtual Reality

Jiacheng (Gary) Liu Yuliang Fan Yujia Yan Yifei Yan
University of Illinois at Urbana-Champaign, Urbana, IL, USA 61801
`{jl25, yf3, yujia2, yifeiy5}@illinois.edu`



Figure 1: Inset for (fixed) foveated rendering. The inset region (dark blue) is rendered in high-res, and the periphery region (light blue) is rendered in low-res and then upsampled to the frame size.

1. Introduction

In virtual reality (VR) devices, rendering requires a lot of amount of graphics resources because it has to be done with high resolution and high frame rate. For example, the state-of-the-art Oculus Quest has a native rendering size of 2304x1280¹ and frame rate of 72Hz. In comparison, gaming PCs typically renders at around 2560x1440 and refreshes at 60Hz or 120Hz. However, with the trend to untether our VR experience, the Quest headsets are only equipped with an SoC (e.g. Qualcomm Snapdragon) and are thus not as privileged as gaming PCs with powerful GPU. Therefore, these VR devices face the challenge of quickly producing high-quality graphics within the hardware constraint.

Foveated rendering is one of the techniques that allows us to deliver high-fidelity graphics with low computation resources. It exploits the fact that human eyes are more sensitive to the central gaze region and less sensitive to the peripheral region. We can save computation by only rendering the peripheral region to a lower quality without user's notice. There are two types of foveated rendering: fixed

¹Although the Quest display size is 2880x1600, its native applications render at 0.8x display size.



(a) Frame rendered in high-res entirely.



(b) Frame rendered with fixed foveation.

Figure 2

foveated rendering (FFR) and dynamic foveated rendering (DFR). In FFR, a fixed window at the center of the frame is rendered at a higher resolution than the periphery. This is based on two observations: (1) people tend to gaze more at the center region of display (i.e. they would rotate their heads to put the attended region at the center); (2) periphery regions of the display are offset from the center, so in human eyes the periphery has a larger "effective resolution" if rendered at the same resolution as the inset. In DFR, gaze direction is estimated from eye tracking data and is used to retarget the high-res inset location.

One problem faced by foveated rendering is the discontinuity at the inset boundary. Although not very visible if



(a) Alpha blending kernel in (b) Result of naive alpha blending
Section 3.1

Figure 3: Naive alpha blending.

viewed on a regular monitor, such discontinuity is rather perceptible for VR users. Our work proposes and analyzes methods for smoothing such discontinuity. We further experiment on videos with both FFR and DFR to show the robustness and applicability of our methods.

2. Task Formulation

We take as input a foveated-rendered image and output the same image with smoothed inset boundary. The input image, illustrated in Figure 2, is composited from a high-res inset and a low-res periphery region. The inset is rendered at normal resolution, while the periphery is rendered at $1/4$ normal resolution and then upsampled by $4\times$. We assume that the inset is $1/4 \times 1/4$ the size of the entire frame. Our goal is to produce an output image that

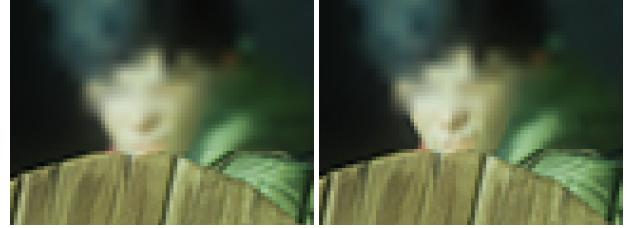
- mostly preserves the high-res and low-res regions
- smoothes out the inset boundary to make it imperceptible (especially for VR users)
- does not introduce significant computation overhead

3. Methods

3.1. Naive Alpha Blending

A simple method we propose to smooth out the inset boundary is to do alpha blending in the inner side of the boundary. The intuition is that we want a gradual transition from the low-resolution region to high-resolution region. For a boundary of the inset, we replace each pixel within an inner (high-resolution) strip of that boundary by a linear combination of the corresponding pixels from the low-resolution frame and the high-resolution frame. The combination coefficient depends on the pixel location, where the weight of the low-resolution pixel value decays as the pixel goes further from the boundary. We use a linear kernel for alpha blending shown in Figure 3a, and use a strip width $S = 8$.

Figure 3b shows a sample result of the naive alpha blending method. Compared to the original fixed foveation



(a) 2D Gaussian kernel (b) 1D Gaussian kernel

Figure 4: Gaussian smoothing.

frame, the horizontal boundary over the avatar's nose is not visible, while no significant artifact is introduced.

The naive alpha blending method is expected to run very fast in the graphics pipeline. However, one drawback of this method is that it has to know the entire low-resolution frame (because it has to know the low-resolution pixels in the inset region). Therefore, this method can only be placed rather early in the pipeline (i.e. before the high-res inset and the low-res periphery are composited together).

3.2. Gaussian Smoothing

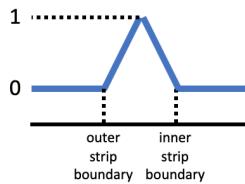
The inset boundary discontinuity can also be viewed as a high-frequency component of the gradients in the direction orthogonal to the boundary. We therefore propose another method, Gaussian smoothing, to address the discontinuity problem. For a boundary of the inset, we take a strip along the boundary (including pixels on both sides) and convolve it with a Gaussian kernel.

We experimented with 2D Gaussian kernels and 1D Gaussian kernels orthogonal to the boundary. We choose $\sigma = 1.0$, a kernel size of 9 and a strip width of 16. Figure 4 shows the result of the Gaussian smoothing method. As we can see, applying Gaussian kernels to the strip introduces significant discontinuity artifact on the strip boundaries, which is not desired. Meanwhile, the strip applied with 1D Gaussian looks sharper than the one applied with 2D Gaussian.

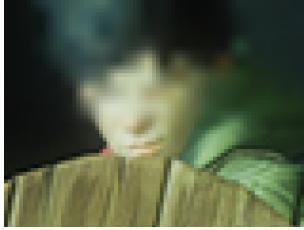
3.3. Combining Gaussian Smoothing and Alpha Blending

To counter the undesired artifact in Gaussian smoothing, we add an alpha blending step after applying the Gaussian kernel. This step composites the strip region from both the original foveated-rendered image and the Gaussian-smoothed image. The kernel for this alpha blending is shown in Figure 5a. Pixels on the inset boundary come from the Gaussian-smoothed image, while pixels on the strip boundary come from the original foveated-rendered image. Pixels in between are linearly interpolated from corresponding pixels in the two images.

We kept the same parameters as in Gaussian smoothing.



(a) Alpha blending kernel in Section 3.3.



(b) 2D Gaussian kernel



(c) 1D Gaussian kernel

Figure 5: Gaussian smoothing + alpha blending.

Figure 5 shows the result of this method. We can see that there is no significant discontinuity on the inset boundary or the strip boundary. Using 2D Gaussian kernel produces slightly better smoothing result at the inset boundary.

3.4. Laplacian Pyramid Blending

TODO: images

We tried to apply Laplacian Pyramid directly to smooth the gap of resolution discontinuity. First we get the high resolution image, then we down sample the high resolution image by a factor of 4 and then upsample by a factor of 4 to get the low resolution. Now we apply Laplacian pyramid on the original image and the sampled image, and reconstruct the new image. However, we found the result is not as good as the alpha blending method. The effects are as follows:

The first image is the resized image(as the low resolution image), the second is the result of using alpha blending, the third one is the result of using laplacian pyramid with five step steps.

There are a few issues in the Laplacian Pyramid blending:

- First, Laplacian pyramid changes the frequency of the high resolution area to low resolution during the reconstruction step. Therefore, the high resolution image at the center will be not as clear as the original image.
- Similar reason as the low resolution part; due to the reconstruction approximation, the resolution will be even lower.
- The image looks brighter than the one using alpha blending. We think that this is due to the addition of

the low-pass result and the high-pass result. The summation is slightly larger than the original image.

4. Metrics and Quantitative Results

After various methods are applied to the images, we need some metrics to characterize the smoothing effect of these methods, and we believe that gradient intensity is a good metric, because after the image is smoothed, the gradient intensity in the same area will decrease. To calculate the gradient intensity at each pixel in the image, we first converted the image from BGR to grayscale, and then applied a Sobel filter with the size of 3×3 , and multiplied the result by 100 just to make the boundary between high and low resolution area more obvious, as shown in Figure 6.



Figure 6: Gradient intensity (100X) in x direction of the resampled image

To quantitatively prove that the metric is effective and valid, we calculated the average gradient intensity in both high and low resolution area, as shown in Figure 1. For the resampled image, we found that the ratio of average gradient intensity between high and low resolution area is 1.78 and 1.74 in x and y direction, respectively. In contrast, the ratio for the original image without high/low resolution area is 1.34 and 1.24, indicating that average gradient intensity is a good metric. If we looked at the values of average gradient intensity, the change of ratio is due to the decrease of average gradient intensity in the surrounding low resolution area.

Our goal is to bridge the discontinuity at the boundary of the high and low resolution area. Now, we can know that compared with the original image, the ratio of average gradient intensity is higher in the resampled image, and the change of gradient intensity is sharp, because the boundary of high resolution area is very clear in the resampled image. Ideally, the change of gradient intensity should be smoother across the boundary, but we are not able to characterize the smoothness of the change only with the ratio of the average gradient intensity of the high and low resolution area.

Since we applied various methods to process the im-

age, and filters are applied only around the boundary of the high and low resolution area, we decided to further section the image into four areas, as shown in Figure 7 where *Area1* and *Area2* are inside the boundary, while *Area3* and *Area4* are outside. We calculated the average gradient intensity in all four areas, so that we can also know how average gradient is changing across the boundary.

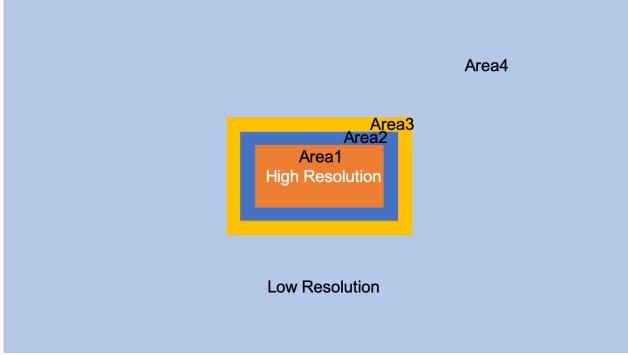


Figure 7: Illustration of further sectioned image. *Area1* and *Area2* are high resolution area, while *Area3* and *Area4* are low resolution area.

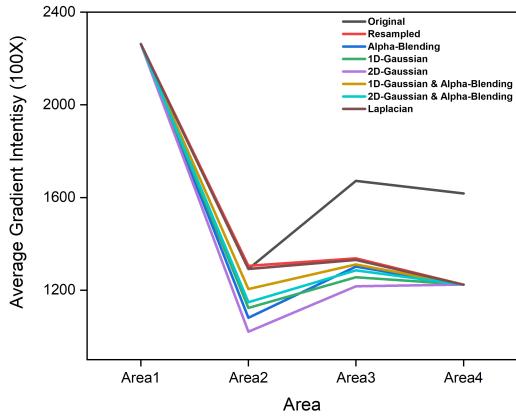


Figure 8: Average Gradient Intensity (100X) vs. Area

We plotted the data points obtained with various methods of smoothing in Figure 8, and for all processed images, the average gradient intensity in *Area1* and *Area4* are the same. For *Area1*, the average gradient intensity is the same as the original image, because this area has the highest resolution. For *Area4*, the average gradient intensity is lower than the original image, because the resolution is lower and *Area4* is not affected by the filters since it's far away from the boundary.

Figure 9A shows the zoomed area of the original image where the resolution is high in the whole image. By look-

ing at Figure 9B of the man's face in the resampled image, we can see that the boundary is very clear, which can be explained by the plot, because the average gradient intensity in *Area1* and *Area2* is the same as the original image, while that of *Area3* and *Area4* is much lower. This is the same for Figure 9D, the image applied with Laplacian pyramid, which has almost the same average gradient intensity in all areas as the resampled image.

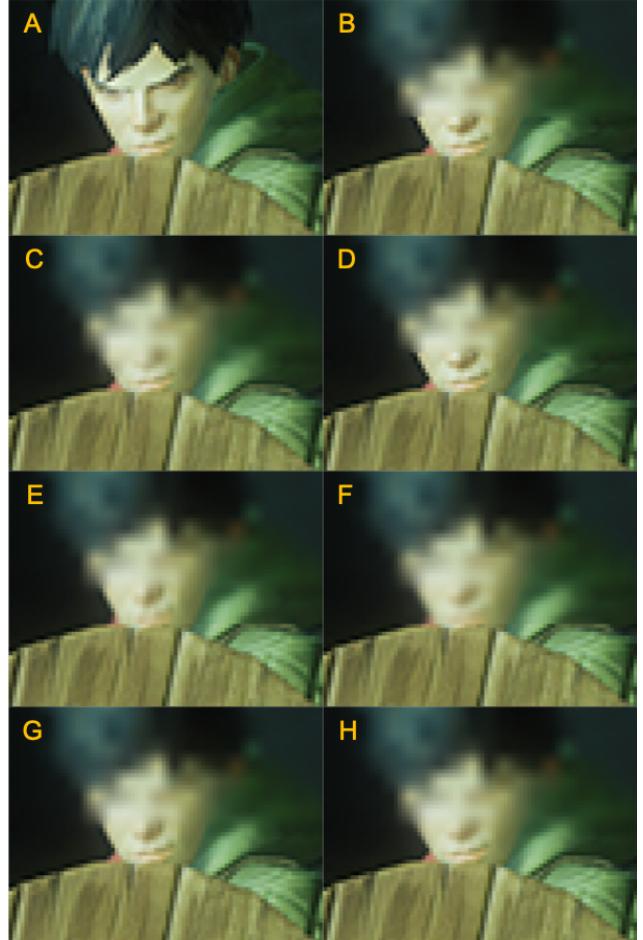


Figure 9: Zoomed image of boundary area processed with various methods. (A) Original image; (B) Resampled image; (C) Alpha-Blending; (D) Laplacian; (E) 1D-Gaussian; (F) 2D-Gaussian; (G) 1D-Gaussian Alpha-Blending; (H) 2D-Gaussian Alpha-Blending

For images processed with all other methods, the difference is in the average gradient intensity in *Area2* and *Area3*. By comparing the zoomed image of the man's face, we found that images applied with only 1D or 2D Gaussian filter (Figure 9E&F) have an obvious edge right under the man's lips, which is probably due to the Gaussian filter and its parameters like σ and kernel size. The resolution of *Area2* is also relatively low compared with images

processed with the combination of alpha blending and 1D or 2D Gaussian filter (`gaussian1d_alpha gaussian2d_alpha`), which can be explained by the average gradient intensity.

Comparing the images applied with Alpha-Blending (Figure 9C), 1D-Gaussian & Alpha-Blending (Figure 9G), and 2D-Gaussian & Alpha-Blending (Figure 9H), we found no obvious difference in resolution in *Area3*, while there is some difference in *Area2*. The image applied with 1D-Gaussian & Alpha-Blending has the highest resolution, which is consistent with the relatively high average gradient intensity. However, the change of average gradient intensity is also relatively sharp across the boundary. For the image applied with Alpha-Blending, it's difficult to see the boundary, but the resolution is also the lowest among the three, as explained by the average gradient intensity. The image applied with 2D-Gaussian & Alpha-Blending is in between the other two images analyzed above, with medium resolution and boundary. Therefore, we conclude that there's a trade-off between resolution and boundary smoothing.

By comparing the average gradient intensity of the image applied with Alpha-Blending and that applied with 1D-Gaussian, we know that these two images have similar average gradient intensity in *Area2* and *Area3*, and the average gradient intensity for each image is higher in one area and lower in the other. Although the resolution in *Area2* is similar for these two images, the boundary smoothing effect of Alpha-Blending is obviously better than 1D-Gaussian. This can be explained by the slope of the line between the data point for *Area2* and *Area3*. Between these two images, the slope of the one applied with Alpha-Blending is closer to the slope of original image, which means that the resolution is lowered to similar extent in *Area2* and *Area3* with Alpha-Blending, which is why the boundary is smoothed better. Therefore, from this comparison, we can learn that in order to make the boundary less sharp, we should make the slope similar to that in the original image.

5. Extending to Videos and Dynamic Foveation

To extend our methods to FFR videos, we simply apply smoothing to individual frames.

For DFR videos, we generate these videos based on a synthetic inset center trajectory instead of real gaze tracking data. For adjacent frames, the inset center moves at a velocity that follows a 2D Gaussian distribution. The size of the inset remains constant over frames. When smoothing the boundary, we assume that we know the true inset center for each frame (which is a safe assumption), and apply the methods to appropriate regions near the inset boundary.

We generated some results for two spotlight VR titles, *Beat Saber* and *Sports Scramble*, using the Gaussian smoothing + alpha blending method. The results are on this YouTube playlist: https://www.youtube.com/playlist?list=PLX4bIC6e61M5ew10X_fhZMxKQjQ1SBsY0

6. Conclusion and Future Work

In conclusion, among all smoothing methods that we tried, Alpha-Blending, 1D-Gaussian Alpha-Blending, and 2D-Gaussian Alpha-Blending can yield the best results. We also found the trade-off between resolution and boundary smoothing. To make the boundary smoother, the difference in average gradient intensity between *Area2* and *Area3* should be similar to that of the original image.

There are certain limitations that we hope to address in future work. First, Our peripheral low-res region is down-sampled from the corresponding region in the full high-res frame. However, in real applications, the low-res region is directly rendered by the engine. It has been shown in VR that an image rendered in high-res and then downsampled has better quality than that directly rendered in low-res, so our algorithm may behave differently in real scenarios.

Second, our synthetic inset center trajectory does not reflect real gaze tracking data. We assumed that the inset center moves with a random 2D Gaussian velocity between frames. Human eye movements are far more complex, such as smooth pursuit, saccades and VOR.

Third, we do not have a comprehensive dataset to evaluate our methods. Our experiments are based on two spotlight VR titles and there might be issues migrating to other applications.

Lastly, we do not have the capacity to do user study to analyze our methods. To faithfully evaluate how these algorithms work requires deep integration into the graphics pipeline of the VR device, which is proprietary code.

7. Individual Contributions

TODO