Head Movement Based Temporal Antialiasing for VR HMDs

Jung-Bum Kim Soo-Ryum Choi Joon-Hyun Choi Sang-Jun Ahn Chan-Min Park

Samsung Electronics
{jb83.kim, sooryum.choi, jh53.choi, sjun.ahn, chanmin.park}@samsung.com

ABSTRACT

Inherent properties of VR HMDs cause degradation of visual quality which disrupts immersive VR experience. We identify a new temporal aliasing problem caused by unintended tiny head movement of VR HMD users. The images that users see slightly change, even in the case that the users intend to hold and concentrate on a certain part of VR content. The slight change is more perceivable, because the images are magnified by lenses of VR HMDs. We propose the head movement based temporal antialiasing approach which blends colors that users see in the middle of head movement. In our approach, the way to determine locations and weights of colors to be blended is based on head movement and time stamp. Speed of head movement also determines proportions of colors in the past and at present in blending. The experimental result shows that our approach is effective to reduce the temporal aliasing caused by unintended head movement in real-time performance.

Keywords

Temporal antialiasing, head movement, virtual reality, head mounted displays

1 INTRODUCTION

VR, Virtual Reality, has been recently gaining enormous attention, since the advent of advanced VR HMD, Head Mounted Display, devices such as Oculus Rift [Ocu16a], Vive [Viv16a], and Gear VR [Gea16a]. The devices significantly enhance immersiveness of VR experience by displaying images full of users' field of view, which promptly reflect movement of users' head posture [Ear14a]. That is, the VR HMDs provide a part of VR content at which users look at a real-time frame rate. However, in terms of visual quality, improvement is required because of inherent properties of the recent VR HMDs. VR HMDs are equipped with optics systems which magnify display panels showing images of VR content. In order to manufacture lightweight and affordable hardware, the optics systems in the recent VR HMDs are relatively uncomplicated, which cause problems with visual quality including spherical and chromatic aberrations [Hen97a]. Although modern displays, in terms of resolution, are dense enough so that users are not able to recognize individual pixels in a panel, they are insufficient for VR HMDs which magnify displays using their lenses. As a result, screen-door effect, which is a problem of a grid of fine lines between pixels is observable, appears. Furthermore, insignificant visual artifacts such as aliasing become noticeable, although they are not serious defects in typical smartphone and desktop environment.

This paper concentrates on identifying and solving a visual quality degradation problem caused by inherent characteristics of VR HMDs. In general, users hold their heads when appreciating a certain part of VR content. However, it is unavoidable for users to make tiny movement which sensors in VR HMDs are able to detect. In response to the tiny head movement, VR HMDs slightly change the images that users see, even in the case that they intend to hold their heads. This slight change in the images is perceived as a temporal aliasing which disturbs comfortable VR experience. In this paper, we define the temporal aliasing which is caused by unintended tiny head movement as *head jittered aliasing*. Although many researches have investigated to resolve aliasing problems, constraints of VR HMDs have not been their concerns. Most of the previous researches are not suitable for eliminating head jittered aliasing. In addition, in VR environment, real-time performance is critical for immersive and long-lasting experience, since users feel motion sickness if high frame rate is not supported [Lav00a] . To preserve real time performance, an antialiasing technique for VR HMDs should be very fast as well as not burdensome. Therefore, a new antialiasing which takes into account VR HMDs is necessary.

In this paper, we propose a head movement based temporal antialiasing which blends the colors that a user sees in the middle of head movement. In terms of performance, the approach is executed at a real time frame rate on a modern mobile VR HMD. Distinctive features of our approach in blending colors are 1) the location of the colors to be blended is determined by partially inverting head movement, 2) the way to derive the weight of the colors is based on speed of head movement and time stamp, 3) blending is localized based on the amount of temporal change of colors. To evaluate our approach, we define a measurement which computes the amount of temporal change of colors in the images. The measurement accounts for how effective an antialiasing is to reduce change in temporally consecutive images. As a value of the measurement lowers, it becomes more effective to reduce head jittered aliasing. The experimental result indicates that our approach outperforms other candidate approaches in reducing head jittered aliasing, and is accomplished at a real-time frame rate.

2 RELATED WORK

2.1 Spatial antialiasing techniques

Supersample Anti-Aliasing(SSAA) and Multisample Anti-Aliasing(MSAA) are basic antialiasing techniques. They have been used to reduce spatial aliasing as generic methods for the past years.

SSAA reduces aliasing artifacts by the following 3 steps; generating the image at a higher resolution, filtering multiple samples for each pixel and then downsampling to the final resolution. Since SSAA is performed for the whole pixel in the image, it has the highest-quality results. However, it is the most expensive method in terms of its processing and memory bandwidth requirements [Jim11a]. Recently most of graphical processors have stopped supporting SSAA to avoid performance degradation [Jia14a].

MSAA is a special case of SSAA that is only performed for pixels at the edges of polygons. By reducing number of samples, it becomes less expensive and faster than SSAA, but it could not improve aliasing artifacts inside geometries and textures. It is supported by all of the latest graphics processors and application programming interfaces(APIs) [Jim11a].

Morphological Anti-Aliasing(MLAA), developed by Intel Labs, blends colors around silhouettes which are detected with certain patterns [Res09a]. These patterns (Z-shapes, U-shapes and L-shapes) are used to search for color discontinuities and determine blending weights. It has advantages in terms of quality and implementation. MLAA provides the quality comparable to 4X SSAA. It also allows for better processor utilization, since it is independent from the rendering pipeline and parallel with rendering threads. However, MLAA might produce temporal artifacts between frames, because it uses only image data for reconstruction. In addition, it cannot identify pixel-size features which very small or thin geometries and unfiltered textures have. Thus, it could be resulted in moire pattern with these input.

Fast approXimate Anti-Aliasing(FXAA), developed by NVIDIA [Lot09a], reduces edge aliasing in a similar way to MLAA. However, it is simpler and faster. It detects edges by checking for a significant change in average luminance, and filters directions of sub-pixel on the edge perpendicularly. It can be easily implemented as one per-pixel filter. In addition, it is extremely fast, averaging just 0.11 ms per million pixels on NVIDIA GTX 480 [Jim11a]. It can handle edge alias even inside textures by processing with all the pixels on the screen. However, FXAA also cannot solve the temporal artifacts.

These spatial antialiasing techniques introduced in this section are not enough to solve temporal aliasing artifacts between consecutive frames, because they only uses a current frame image [Sch12a].

2.2 Temporal antialiasing techniques

Temporal aliasing is caused due to incoherence between continuous frames. This artifact is shown as flickering or crawling pixels temporally during camera and object motions. There are some approaches to reduce temporal aliasing.

A temporal antialiasing method for CryENGINE 3.0 has been popularly used in video games [Jim11a], because it is a simple method which is also known as Motion Blur. It is performed in real-time by using two images of previous and current frame and a velocity vector between them.

Amortized supersampling by Yang et al.[Yan09a] proposed an adaptive temporal antialiasing with supersampling, reusing shading samples from previous frames. It controls the tradeoff between blurring and aliasing with the smoothing factor calculated in a recursive temporal filter. However, it cannot properly handle temporal artifacts resulted in fast changes which cannot be predicted by reprojection.

Recently, Karis [Kar14a] presented high-quality temporal supersampling as a temporal antialiasing technique for Unreal Engine [Epi16a]. It generates super samples by jittering the camera projection, and then takes samples with a pattern such as Halton [Hal60a] Sequence. It accumulates the previous moving average of samples and uses it as the smoothing factor to reduce temporal alias.

Some approaches with supersampling, such as Yang's and Karis's methods, produce high quality results, but they have a limitation in terms of the performance with high-resolution images. In addition, temporal reprojection can cause ghosting artifacts, since it cannot accurately reproject when the images are significantly changed between consecutive frames.

3 HEAD JITTERED ALIASING

A majority of advanced VR HMDs consist of sensors, displays, and an optics system. Sensors in a VR HMD detect movement of users' head posture at a very high frequency. As users' head posture change, VR HMDs render images of VR content in the direction of users' field of view to the displays. The images in the displays are magnified by the optics system to fully occupy users' field of view. By instantly displaying images full of users' field of view in response to head movement, VR HMDs provide users with immersive experience. The magnification of the images influences density of pixels in the displays to be decreased, and makes it more vulnerable for insignificant visual artifacts to be noticed. We identify a visual quality problem that users with a VR HMD experience, when they intend to hold and concentrate on a certain part of VR content. It is unavoidable for users to make tiny head movement, even in the case they attempt to hold. The tiny head movement of users is detected by the sensors, then the images displayed to users are slightly changed. Figure 1 illustrates the slight change in the images in response to the tiny head movement. Since VR HMDs update images at real-time frame rate, the slight change is supposed to be noticed as temporal aliasing artifact. While it is not critically noticeable at typical devices such as smartphones, users with VR HMDs are able to easily perceive the temporal aliasing because of the magnification of the optics system. In Figure 1, it is difficult to observe the difference between the two images. However, in the magnified regions in the images, the change in colors of the images is more perceivable. Therefore, this is a problem that arises due to inherent properties of VR HMDs. The temporal aliasing problem occurs in various cases including computer generated-geometries, texts, and images.

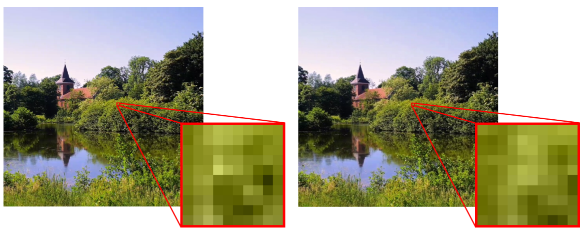

Figure 1: Slight change of temporally consecutive images in the case that users concentrate on a certain part of VR content.

In this paper, we define the temporal aliasing caused by unintended tiny head movement as *head jittered aliasing*, as it occurs because of head jittering of users. The previous antialiasing techniques, introduced in section 2, are not appropriate to remove this artifact. The spatial antialiasing techniques which aim to solve aliasing problems in a spatial manner are not effective for eliminating temporal aliasing. The temporal antialiasing techniques are not feasible for mobile VR HMDs. They are designed for desktop GPUs such as NVIDIA GeForce, which indicates that supporting real-time performance is not achievable. We propose a temporal antialiasing approach which is based on head movement to solve the problem. Our approach is suitable for mobile VR HMDs in terms of performance and effectiveness.

4 HEAD MOVEMENT BASED TEMPORAL ANTIALIASING

Head jittered aliasing is basically caused due to abrupt change of colors in images during tiny head movement. Blending colors with organized weights is a common technique to compensate abrupt change of colors. Basically, our temporal antialiasing approach blends colors that users see in the middle of head movement. Our approach is head movement based temporal antialiasing, which indicates selection of colors and deriving weights to be blended are based on head movement.

4.1 Interpolated reprojection

In order to select a color that a user sees during head movement, we partially invert head movement. In this paper, we assume that the type of head movement detected by VR HMDs is rotation. It is possible to extend our approach to 6 DoF head movement. To achieve a partial inverse of head movement, we introduce *interpolated reprojection* which transforms a sample at which a user is currently looking to the past locations in the middle of head movement. A location of a color that interpolated reprojection returns is a two-dimensional coordinate in an image space. We call a location of a color in an image space a sample. Interpolated reprojection is represented by the following function.

$$s = P \cdot slerp(V_{n-1}, V_n, d) \cdot V_n^{-1} \cdot v_p \qquad (1)$$

s is a past sample that is acquired after applying a partial inverse of head movement. d is a degree of inverting head movement, which is equivalent to closeness to the current head posture. d ranges from 0 to 1, and the value of 0 indicates a full inversion of head movement. v_p is a three-dimensional coordinate in $R^3 \in [-1, 1]$, which is also known as a clip space. v_p represents a sample in the current image with a depth information and is obtained after applying a projection. V_n is a view matrix that denotes the current head posture. V_{n-1} is a view matrix that denotes the head posture in the previous frame. $slerp$ is a spherical linear interpolation function that calculates a matrix in the middle of the two view matrices V_n and V_{n-1}. d is used as a parameter that determines closeness to V_{n-1}. P is a projection matrix applying camera parameters. Interpolated reprojection is a process that finds a sample to be blended. By using multiple different values of d, it is possible to variously control the number of samples to be blended. By substituting the *slerp* function to a function that returns an intermediate transform between two transforms, Interpolated reprojection is extended to support 6 DoF head movement.

4.2 Determination of blending weight

In our approach, determination of a weight of individual color to be blended is based on both time stamp and speed of head movement. Basically, we assign a greater or equal weight to a more recent sample. We compute c_{past}, accumulated colors of past samples, using the following equation.

$$c_{past} = \sum_{0}^{n-1} W(d_i)C_k(s_i) \quad (2)$$

n is the number of past samples. s_i is a past sample which is an output of Equation (1). d is a closeness to the current head posture. d_i is used to compute a sample s_i. That is, a past sample with a value of d closer to 1 is more recent one. C_k is a function that returns a color of a sample from an image displayed in the k^{th} frame. W is a monotone function that returns a weight of a sample based on value d. Given the number of samples n, the total sum of the values returned by the function W is 1. According to the function W, a more recent sample has a greater or equal weight.

Head jittered aliasing becomes serious when users attempt to hold their head movement. In addition, blending with past colors is possible to cause an excessive blur which deteriorates quality of images. Therefore, we decrease strength of blending, as speed of head movement gets faster. In order to find c_k - a color in the k^{th} frame, we blend the current color with the accumulated past color through the following equation.

$$c_k = (1 - A(h)) \sum_{0}^{n-1} W(d_i)C_{k-1}(s_i) + A(h)C_k(s_c) \quad (3)$$

The first term of the equation comes from Equation (2). As the function C_{k-1} is used in the first term, colors of past samples are obtained from the $k-1^{th}$ frame. Accordingly, s_c is a sample in the k^{th} frame. Since a color c_k is derived from colors in the k^{th} and $k-1^{th}$ frames, blending in our approaches recursively accumulates colors in a period of frames. A is a function that determines a weight of a color of s_c from h, speed of head movement, as an input. The function A should be monotonously increasing for speed of head movement, and has a curve similar to a ease-in and ease-out curve. Figure 2 illustrates a graph of the function A, which satisfies the conditions. The graph has different forms depending on a parameter specifying a minimum weight, w_{min} in Figure 2.

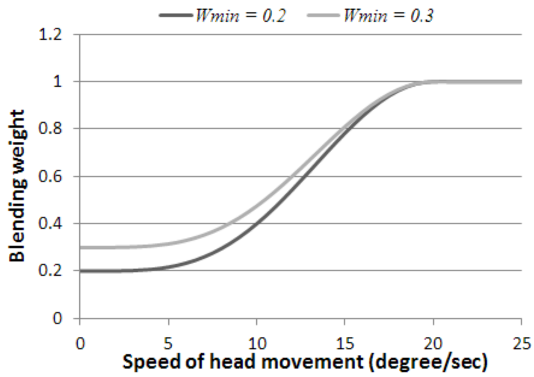

Figure 2: A graph representing the function A in Equation (3)

From experiments, we conclude that an appropriate function for A is as follows.

$$w = (1 - w_{min})\{-\frac{1}{2} * (\cos(\pi \frac{h}{h_{max}}) - 1)\}^2 + w_{min} \ , \quad w = 1 \quad if \quad h > h_{max} \quad (4)$$

w is a result weight value. h is speed of head movement, and h_{max} is a maximum value of speed of head movement. w_{min} is a minimum value of a weight. Using the function A, a weight of an accumulated past color gets smaller, as speed of head movement becomes faster.

4.3 Localization of blending weight

The blending function in Equation (3) assigns a constant weight to the entire colors in an image. However, the amount of change of each color in an image during head movement is diversified. And we observe that temporal aliasing is more noticeable in areas that have larger color change. In order to enhance effectiveness of our temporal antialiasing, we locally assign a weight depending on a temporal difference of individual colors. A sample with a larger color difference has a smaller weight. w_{min} in Equation(4) is substituted to w'_{min} to achieve localization of blending weight as follows.

$$w'_{min} = w_{min} - (w_{min} - w_{lb}) * c_{diff} \quad (5)$$

c_{diff} is a temporal difference of a color. w_{lb} is a lower bound of a minimum weight, which indicates that the largest value of c_{diff} has a weight value of w_{lb}.

Our localized weight determination is devised to be suitable for parallel processing on GPUs. VR HMDs normally produce images using GPU for better performance. Calculation of each color in images is parallelly executed on shaders of GPUs. However, computation of functions with high complexity such as Equation (4) for all the colors in a image is burdensome, which leads performance degradation. To secure high performance, our approach minimizes the amount of weight computation on shaders of GPUs, by separating a complex part of the computation - Equation (4) in this case

- that is globally applied to all the colors in an image. The complex part is operated on CPU, and the result of computation is delivered to shaders. As a result, in our approach, the relatively simple function - Equation (5) - is performed on GPUs for weight computation.

4.4 Compatibility with dynamic scenes

Blending of temporally consecutive images possibly causes a motion blur problem. To avoid this problem, one of common methods is to analyze a velocity of individual colors, and to selectively apply blending. However, in VR HMD environment, high performance is a top priority for users not to experience motion sickness. Therefore, complicated analysis requiring enormous computation is not feasible. Our approach is designed to be compatible with a map which is a form of a 2D image specifying dynamic areas. By referencing the map, we selectively apply blending to static regions in images. For a performance reason, we take advantage of scene information. Our approach rasterizes a region on which static objects are projected on the map, which is a process marking static regions on the map. Since rasterization of the map, with less cost, is accompanied with rendering of a scene with the help of a functionality of GPUs, producing the map in our approach is able to preserve performance.

5 EVALUATION

5.1 Mean Temporal Color Difference

To quantitatively evaluate effectiveness of temporal antialiasing approaches, we define a new measurement - MTCD, Mean Temporal Color Difference, which computes the average amount of change in colors from temporally consecutive images. Some researches employ PSNR, Peak signal-to-noise ratio, for evaluation of antialiasing. However, PSNR to an optimal image is not appropriate for measuring effectiveness of temporal antialiasing. It is even possible that a sequence of temporally consecutive images having considerable temporal aliasing is able to achieve low PSNR to corresponding optimal images. Suppose that all the images in a temporal sequence of images has a specific PSNR value ε to corresponding images in an optimal sequence of images. All the difference between pixels in k^{th} images is negative and that of $k-1^{th}$ images is positive. In this case, temporal change of colors in images is possibly regarded larger than PSNR indicates. Therefore, we need a new measurement that takes into account temporal coherence of image sequences. MTCD of an image sequence is defined as Equation (6).

$$MTCD(I,t) = \frac{1}{nm(t-1)} \sum_{k=1}^{t-1} \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} d_{ijk},$$

$$d_{ijk} = \sqrt{(r_{ij(k-1)}-r_{ijk})^2+(g_{ij(k-1)}-g_{ijk})^2+(b_{ij(k-1)}-b_{ijk})^2} \quad (6)$$

I is a sequence of temporally consecutive images in a period of t frames. A width and a height of an image in I are n and m. i and j denote x and y coordinates in an image, and k represents the k^{th} image in a sequence of images. r,g,b represent color components for red, green, and blue respectively. An optimal value of MTCD is definitely zero. As MTCD gets smaller, a sequence of images is more robust to temporal aliasing.

5.2 Experimental result

For evaluation of efficiency and effectiveness of temporal antialiasing, we build a platform that consists of a mobile VR HMD (Gear VR), a smartphone (Galaxy S7), and an image viewer. The resolution of the VR HMD in our experiments is 1024x1024 for each left and right eye. The device is equipped with Exynos 8890 processor which includes a 2.3GHz Quad-core and 1.6 GHz Quad-core CPU, and a Muli-T880 MP12 GPU. The dataset in the experiments includes 11 images. With the dataset, we perform experiments measuring performance using frame rate, and effectiveness of reducing temporal aliasing using MTCD.

5.2.1 Performance

Real-time performance is essential for immersive and long-lasting VR experience. Temporal antialiasing is an additional process that requires more execution time. Therefore, performance requirement is very intensive to preserve real-time performance of applied applications. In the experiments, we measure additional execution time after applying our approach. The additional execution time of our approach is approximately 2 *msec* in average. As antialiasing is applied for each left and right eye in VR environment, the measured execution time contains the amount of time for applying our approach twice for both eyes. For quality experience, we observe that the entire execution time for an application to render one frame should be less than 32 *msec*. Because the additional execution time of our approach is approximately less than 10% of 32 *msec*, we conclude that our approach performs at reasonable performance for immersive VR experience.

5.2.2 Effectiveness

We utilize MTCD measurement to quantitatively measure effectiveness of our approach for the purpose of reducing head jittered aliasing. To measure MTCD, we set up experiments simulating an image viewing application. In the experiment, participants are requested to hold and concentrate on a certain part of an input image. Then, we compute MTCD from a sequence of images displayed to users while the participants attempt to hold. A sequence of images in experiments contains 10 images.

For comparison, we choose MSAA [Jim11a] which is the most common antialiasing in mobile environments because of its high performance. Other recent antialiasing techniques are possibly considered as candidates. The techniques, however, are not feasible for adopting mobile environment such as Gear VR for a performance reason, since they are intended to operate on desktops or consoles. In experiments, three variations - no antialiasing, MSAA, and our approach - are compared. The comparison result of the approaches is plotted in Figure 3. The parameter values used are as follows. d in Equation(3) is $2/3$. One past sample is used in the experiments, so n in Equation(3) is 1. h_{max} in Equation (4) is 25. For Equation (5), values of w_{min} and w_{lb} are 0.3 and 0.1 respectively. For measuring MTCD values, the number of images in a sequence - t - is 10.

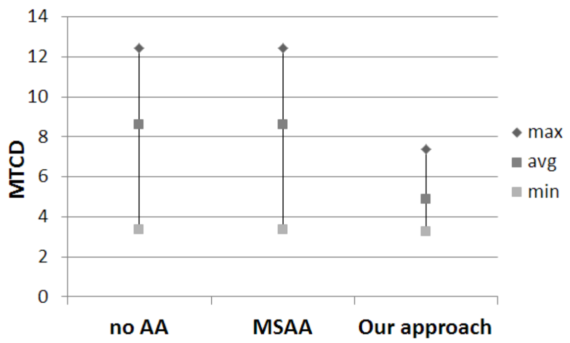

Figure 3: A comparison of MTCD results

The experimental result shows maximum, minimum, and average MTCD values of the approaches for the dataset. Our approach achieves the lowest MTCD in average, which implies it is the most effective to reduce temporal aliasing. The average MTCD of our approach is approximately 56% of MSAA. The results of no antialiasing and MSAA are almost identical, because antialiasing is applied to edges in case of MSAA and our dataset mostly consists of textures. Minimum values of all the approach are almost same, although a minimum MTCD value of our approach is slightly lower than other two approaches. One of the images in the dataset has almost same color in the entire image, and its spatial color change is insignificant. This image contributes to the result that minimum values of MTCDs are almost indistinguishable.

It is possible for our approach to be applied in combination with MSAA. Combined with MSAA, our approach is expected to perform most effectively.

Figure 4 illustrates a comparison of result images. In Figure 4(a), the result images of MSAA are represented. And $k-1^{th}$ (left) and k^{th} (center) images in a temporal sequence of images are depicted. The smaller images with red borders on the right side of each image magnify red rectangular regions on the corresponding images. The difference between the two images on left and center is shown on the rightmost side. The images in Figure 4(b) are the result of our approach. The difference images on the rightmost side describe the amount of change of colors in temporally consecutive images. Darker regions represent larger difference. The difference images indicate that the result of our approach is more effective for reducing temporal aliasing.

6 CONCLUSION

In this paper, we define head jittered aliasing which is a new temporal aliasing problem identified due to properties of VR HMDs. To alleviate head jittered aliasing, we propose head movement based temporal antialiasing which blends colors that users see in the middle of head movement. Our approach determines weights for blending based on head movement, time stamp, and speed of head movement. In addition, the derived weight is localized based on the amount of temporal color difference. For quantitative evaluation of effectiveness, we define a new metric - MTCD - which measures the average amount of change in colors from temporally consecutive images. In the experimental results, our approach has the lowest MTCD among other competitive antialiasing approaches, which implies that our approach is the most effective for reducing head jittered aliasing. In terms of performance, the additional execution time after applying our approach is 2.5 msec in average, which is reasonable for quality VR experience.

7 FUTURE WORK

Our approach takes advantage of a map specifying dynamic regions to be compatible with dynamic scenes. For an easier and a more portable application of our approach, we plan to develop a method identifying dynamic regions independent upon scenes at high performance.

Also, we expect that reducing head jittered aliasing is effective to alleviate visual fatigue which is one of the serious problems of VR HMDs. To validate the expectation, we plan to conduct qualitative analysis on effectiveness of our approach for relieving visual fatigue, and figure out correlation between MTCD and the qualitative measurement.

8 REFERENCES

[Ocu16a] Oculus Rift website: http://www.oculus.com/rift/, 2016.

[Viv16a] HTC Vive website: https://www.vive.com/, 2016.

[Gea16a] Samsung Gear VR website: http://www.samsung.com/global/galaxy/gear-vr/, 2016.

[Ear14a] Earnshaw, Rae A., ed. Virtual reality systems. Academic press, 2014.

[Hen97a] Hendee, William R., and Peter NT Wells. The perception of visual information. Springer Science & Business Media, 1997.

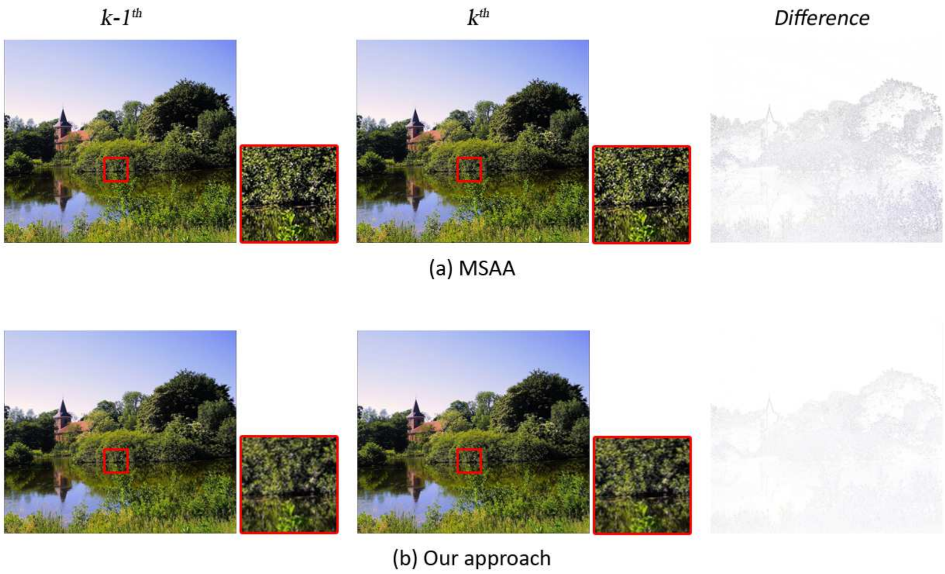

Figure 4: A comparison of result images. Two temporally consecutive images and their difference of the cases of no antialiasing (a) and our approach (b)

[Lav00a] LaViola Jr, Joseph J. A discussion of cybersickness in virtual environments. ACM SIGCHI Bulletin 32.1: 47-56, 2000.

[Jim11a] Jimenez, Jorge, et al. Filtering approaches for real-time anti-aliasing. ACM SIGGRAPH Courses 2.3: 4, 2011.

[Jia14a] Jiang, X. D., Sheng, B., Lin, W. Y., Lu, W., Ma, L. Z. Image anti-aliasing techniques for Internet visual media processing: a review. Journal of Zhejiang University SCIENCE C, 15(9), 717-728, 2014

[Res09a] Reshetov, Alexander. Morphological antialiasing. Proceedings of the Conference on High Performance Graphics, ACM, 2009.

[Lot09a] Lottes, T. FXAA-Whitepaper. Tech. rep., NVIDIA, 2011. 2, 2009.

[Sch12a] Scherzer, Daniel, et al. Temporal Coherence Methods in Real-Time Rendering. Computer Graphics Forum. Vol. 31. No. 8. Blackwell Publishing Ltd, 2012.

[Yan09a] Yang, Lei, et al. Amortized supersampling. ACM Transactions on Graphics (TOG). Vol. 28. No. 5. ACM, 2009.

[Kar14a] Karis, B. High-quality temporal supersampling. Advances in Real-Time Rendering in Games, SIGGRAPH Courses 1, 2014.

[Epi16a] EPIC GAMES: Unreal Engine 4. https://www.unrealengine.com/, 2016.

[Hal60a] Halton, J. H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik, 2(1), 84-90, 1960.