EG UK Computer Graphics & Visual Computing (2017)
Tao Ruan Wan and Franck Vidal (Editors)

Efficient remote rendering using equirectangular projection

J. McNamee,[1] K. Debattista[1] and A. Chalmers[1]

[1]University of Warwick, UK

Abstract

Presenting high quality Virtual Reality (VR) experiences on head-mounted displays (HMDs) requires significant computational requirements. To ensure a high-fidelity experience, the displayed images must be highly accurate, detailed and respond with a very low latency. In order to achieve high-fidelity realistic experiences, advantage needs to be taken of remote high performance computing resources. This paper presents a novel method of streaming high-fidelity graphics content from a remote physically accurate renderer to an HMD. In particular, an equirectangular projection is transmitted from the cloud to a client, so that latency-free 360° observations can be made within a viewpoint.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Graphics Systems—Distributed/network graphics

1. Introduction

Virtual Reality requires the ability to generate visual environments with a high degree of accuracy, if sufficient authenticity is to be reached. For serious applications providing convincing virtual experiences to users, the high-fidelity graphics rendering requires the simulation of light transport using physically-based algorithms such as path tracing [Kaj86]. Being able to obtain these graphics in a timely manner, however, currently requires large, fixed computing resources. This is necessary if a head-mounted display (HMD) is to be used comfortably [HGAB08]. When these are not available locally, the solution is the use of remote rendering services, including those available in the cloud. Remote rendering permits resources like supercomputers or high-performance computing clusters to be used anywhere there is an internet connection and a client capable of receiving and displaying the data.

This accessibility however brings its own limitations. Efficient compression is required to transmit the rendered graphics from the remote rendering service to the user in a reasonable amount of time. A user wearing an HMD expects a minimal amount of latency between a movement and a response in the image, to avoid motion sickness [DL92]. In addition, a high level of quality for the content must be maintained to retain the verisimilitude of the rendered image; there is little point in using vast resources to display an image which is heavily compressed. These data requirements rapidly increase as stereo imagery to supply to either eye is added.

Workstations, even relatively low-powered ones, today contain powerful local hardware which can be utilised for efficient division of work. Remote rendering always has a minimum cost to the client in terms of resources, even if it is only the cost of operating a hardware video decoder such as are commonly found in smart-phones. Client devices nowadays possess resources in excess of this which can be exploited to maximise the performance, in quality or latency, of a remote rendering system. A workstation, for example, will typically have a GPU capable of basic graphical rendering tasks. The application of this is not a straightforward problem, as the division needs to be simple enough that computing the combination of the two efforts does not negate the benefit gained compared to simply computing the whole render at the server.

This paper explores methods of efficiently streaming graphics rendered in high quality so as to supply an HMD or other immediate display with minimal latency. Specifically, this paper presents novel results related to the use of equirectangular projections for virtual, rendered experiences, based on reprojection from calculated motion flow.

This is accomplished, in part, by dividing the work between client and server with the (reduced) bulk of the rendering computed on the high performance resource and the client accounting for reprojection costs with availability of depth data computed locally.

These techniques for presenting additional views based on a single full-quality transmitted frame and varying quantities of additional data permit a local, low-latency response. As the client has both a high-quality 360° image to pan and tilt within, as well as access to depth information with which to reproject that image to simulate spatial motion, applications can be kept responsive over increased server latency.

2. Related Work

Levoy [Lev95] proposed a system for mixing server and client rendering based on the ability of a server to render high quality textures

DOI: 10.2312/cgvc.20171278

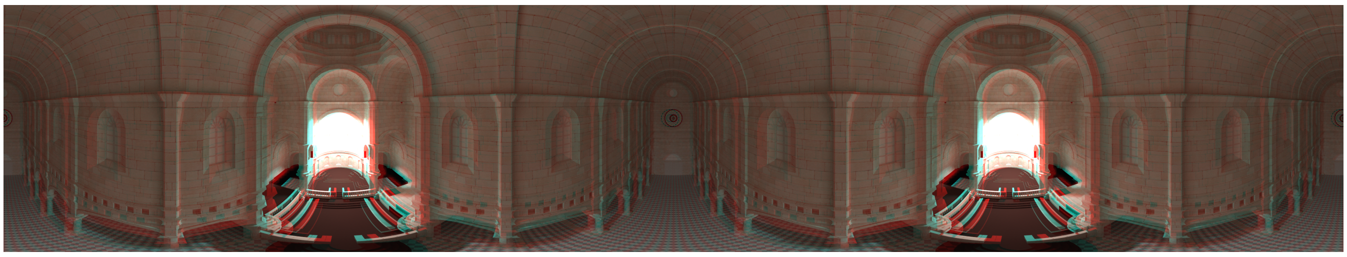


Figure 1: *An anaglyph stereo representation of an equirectangular projection of the Sibenik scene.*

and complex shading, while relying on a lower-powered client to render information such as smooth shading and sharp edges, on the basis that such information performs poorly in typical video encoders. This depended on calculating both a high-quality and low-quality rendering on the server end and then encoding and streaming with a video encoder the difference between the two. This approach relied on the fact that the difference added to the low-quality render produced simultaneously on the client would mathematically produce the original high-quality render. Video codecs are not well suited to encoding difference information however, and combined with their inherently lossy nature, the output will not match the original image exactly.

Efficient modern video codecs use motion vectors to predict the movement of parts of the image across the screen, allowing a rate of less than one bit per pixel to be achieved. Motion vectors can be calculated much more easily in a rendering engine than by analysis of video alone. In McMillan et al. the cylindrical projection used to generate imagery makes the production of a motion flow straightforward, as all motion in the scene becomes motion across the surface of the cylinder, unlike in a more standard perspective view where motion is necessarily warped [MB95].

Wallach et al. described a method of passing motion vectors straight from a renderer to the then state-of-the-art MPEG video encoder. They showed significant efficiency improvements over the use of exhaustive search methods [WKC94].

Pece et al. proposed a custom schema for encoding a continuous stream of depth buffer information using regular video codecs [PKW11]. Working on a 16-bit unsigned integer buffer, the depth information is packed into the luminance and two chrominance channels of a regular video stream (i.e. H.264). The luminance channel is a straightforward linear packing of the buffer. The chrominance channels are provided by a piecewise linear triangle wave each, sufficiently fast-changing to be represented meaningfully in a potentially subsampled chroma stream.

Metadata buffers present unique challenges and opportunities for streaming data [PKW11]. They are visibly simpler than full realistic video in that they tend towards flat colours or smooth gradient and well-defined edges, and at the same time are inefficiently placed in a regular video codec; a high bit depth one-dimensional depth buffer has only 8 bits for its payload in a regular codec, and yet a further 4 bits of chrominance data is left empty.

Didyk et al. [DRE*10] described a method of adaptively scaling block sizes for reprojection based on disparity to provide a stereo view for a perspective camera in a virtual environment.

Methods of rendering virtual environments with 360° views were examined by Ardouin et al. [ALMM14] who discussed various projections and their utility for a user interacting directly with a 360° view as their interface.

3. Overview

In this paper we compare an existing stereo streaming system which utilises reprojection to create a stereo image from a single rendered viewpoint with an innovative system which streams 360° projections to allow free changes of viewpoint within a single frame. In this way the cost of tracking the head movement of a user in a HMD is removed, lowering the overall latency of the system.

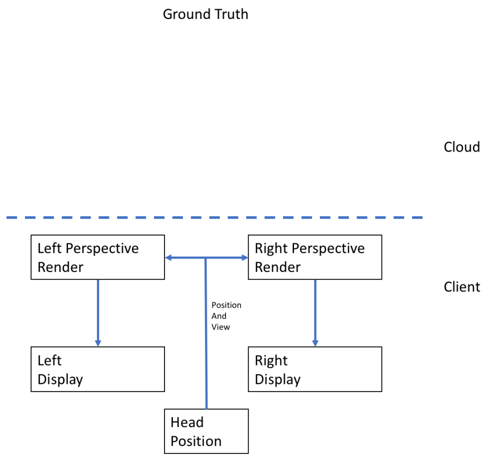


Figure 2: *The ground truth process for rendering stereo: two perspective images, both generated locally.*

3.1. Equirectangular projection

There are many potential projections of a globe onto a flat surface, and indeed the problem predates computing entirely, originally forming a problem within cartography. We have opted to use the equirectangular projection within this paper, both for its suitability to graphics hardware and its adoption as a de facto standard for the streaming of 360° video by online video services and commercial cameras. The task of adapting a ray tracer to generating an equirectangular projection is fairly straightforward: Rather than the flat, screen-shaped camera plane used in traditional ray tracing, instead a set of rays are generated based on spherical coordinates to map out a full sphere. These rays are then evaluated in the typical fashion and the resulting colour values (and any relevant metadata) can be mapped back into a regular image. The x axis in the image becomes panning rotation around the sphere's y axis and the y axis in the image serves as vertical tilt, applied to the camera direction C_{dir}. In equation 1, i and j are normalised image coordinates.

$$Ray(i,j) = C_{pos} + R_x\left(-\frac{\pi}{4} + \frac{\pi}{2} \cdot j\right) \cdot R_y(\pi \cdot i) \cdot C_{dir} \qquad (1)$$

In order to display a regular perspective image to the user, the client must be capable of generating a sphere and mapping the equirectangular projection to this sphere. With this performed, any pure rotation of the HMD can be simulated as rotation of the view of the sphere. In addition, a small amount of realism can be gained by having displacement represented as movement within the sphere – e.g. having leaning forward bringing the user's perspective closer to the surface of the sphere. This is a very limited effect however, as the inability of the projection to reveal occluded locations will quickly become apparent.

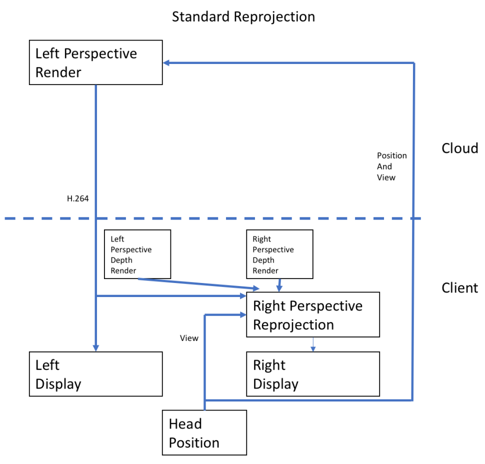


Figure 3: *An existing reprojection-based remote rendering stereo solution. a single image is rendered remotely, encoded with H.264, and transmitted to the client where it is reprojected to provide a stereo perspective.*

3.2. Reprojection

A standard approach to considering the acquisition of more than one view from a single frame is reprojection, where the image values of a previous image temporally or spatially are projected back to a new camera position. This provides a screen-space transformation from one image to the other, with only disoccluded regions - gaps - remaining unaccounted for.

As our images are generated from a renderer rather than a camera, we have access to accurate information about the image which is unavailable in other circumstances. We can quickly and cheaply generate metadata about the scene from its geometry, such as depth, and from depth, screen-space motion vectors. For the goal of generating stereo imagery, the left eye position can be projected to the right eye position.

3.3. Reprojecting equirectangular images

Reprojection using equirectangular images works similarly to reprojecting a perspective image, in that the projection of the camera is inverted to produce a line back to the initial camera position. An inherent advantage with a spherical camera is that as a full 360° sweep is included in each image, there are no gaps introduced around the edge of the image specifically due to the frame of the image moving.

3.4. Generating depth images

Where traditional methods of reprojection have relied on using a depth channel sent alongside a rendered video, this paper evaluates an innovative method of generating depth maps in real time using OpenGL and a copy of the local geometry data. In this way, the client can respond to increases in latency from the server by continuing to form reprojections of the current and previous images received, to allow seamless continuity of view movement for a user until the server recovers. These depth maps are used for both the standard perspective method as well as an OpenGL equirectangular projection for the equirectangular method.

In scenarios where the geometry of the scene is too large or too resource-intensive for the client to responsively process, a H.264 or H.265 stream of the depth buffer can be transmitted from the server alongside the processed images, at a reduced quality [PKW11].

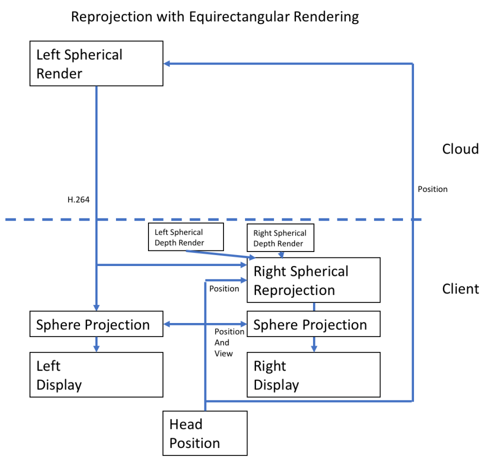


Figure 4: *The proposed system of remote rendering, in which a single equirectangular projection is rendered remotely, encoded with H.264, then transmitted to the client, where it is transformed to provide two separate perspective views.*

4. Methodology

Five sequences of frames were rendered using an unbiased path tracer at 1,000 samples per pixel for 150 frames, at 1080p resolution. The same resolution was used for both perspective renderings and equirectangular renderings, in order to place the same demand on the server for both image sets. This necessarily puts the equirectangular images at a quality disadvantage, as a high percentage of pixels rendered aren't displayed in any one projection onto a sphere. Further work could be done to establish the point of quality parity between a spherical projection and an original perspective render, but that is outside of the scope of this paper. Each sequence was rendered using a path tracing renderer at both stereo viewpoints as well as rendering dual 360° equirectangular projections using a spherical camera, one at each stereo eye position.

4.1. Metrics

To compare the image quality of the outputs, two standard image metrics, PSNR and SSIM, were used [HTG08] [WBSS04]. Both metrics were used to compare outputted images to the 'ground truth' provided by the original renders for the left and right stereo perspectives. PSNR measures the loss of fine detail in the images, while SSIM measures the structural similarity. PSNR is expressed in dB, with higher values representing greater fidelity to the original image, while SSIM is expressed as an average index over the image, with values closer to 1 representing high fidelity.

4.2. Filling gaps

Gaps introduced in reprojection can be handled with a number of techniques, including interpolation or filling with a simple raster representation of the scene calculated on the client. Where the reprojection technique introduced gaps into the image here, these were filled using a 16x smaller render of the relevant stereo image encoded in a second H.264 stream alongside the main render. Reprojection methods typically struggle with reflection and refraction due to the complicated way reflections and refractions move along with camera motion. Regions where this causes significant disparity between ground truth and the reprojections can be treated as gaps.

4.3. Scenes

Five scenes were tested, Sponza, San Miguel, Rungholt, Kitchen and Sibenik [McG11]. These are shown in Table 1. The scenes were chosen to represent a range of environments including indoor and outdoor scenes as well as more and less visually complex scenes.

4.4. Encoder

The rendered frame sequences were encoded using H.264, a commonly used video encoder for streaming video. The videos were encoded at two quality levels, high quality CRF 16 and medium quality CRF 24. H.265, a newer encoding standard, could be used to achieve higher quality results or a reduced bandwidth, but H.264 was chosen for this experiment due to the stability of the encoders available.

4.5. Hardware

The server-side path tracing was preprocessed using a renderfarm and the ground truth input to the methods was provided using full-range OpenEXR files. The client side reprojections were performed using a Macbook Pro with a 2.6ghz Intel Core i7 and a Geforce 750M GPU. The code was implemented for the GPU using a combination of OpenGL and OpenCL.

4.6. Pipeline

To obtain the results, the rendered frame sequences were encoded and decoded into EXR files, then reprojected and/or projected onto a sphere, using the original camera path to maintain direction of view. In this way, pixel-accurate representations of the same view were obtained for both the reprojections and the equirectangular projections, which could be compared using quality metrics to the ground truth.

5. Results

Results are presented in Figures 7 and 8 for quality metrics on each of the four sequences. The measured quality of the images tends to drop as more transformations are made from the original rendered output to the final presented image. For scenes featuring simpler texturing, such as the Kitchen scene, the effect is less pronounced. The drop is also less pronounced in the CRF 24 images, indicating that the loss of quality is to do with high frequency information being lost by reprojection, which functions as a form of resampling.

The San Miguel sequence suffered in particular for all reprojections, as in the data set the left eye is briefly obstructed several times by different leaves in the scenes, providing little to no valid data to reproject into the right eye position. This holds even for the 360° view, as the leaf obstructs the region of the image in view.

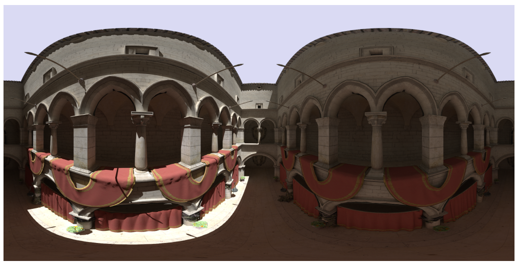
In Figure 6 it can be seen that the processing times for both methods are similar for three of the five scenes, but in the two high-polygon outdoor scenes (San Miguel and Rungholt) the required client-side processing is greater. This is probably due to the increase in OpenGL draw calls when calculating the equirectangular depth maps (from one to seven). It is possible that through more aggressive model culling these figures could be brought back to the level of the rest of the scenes.

Table 2 and Figure 5 show the filesize for each of the processed H.264 video files. For the majority of files, the use of the CRF maintains a similar filesize for perspective renderings as with equirectangular ones. There is a slight trend towards a reduction in bitrate for the equirectangular files.

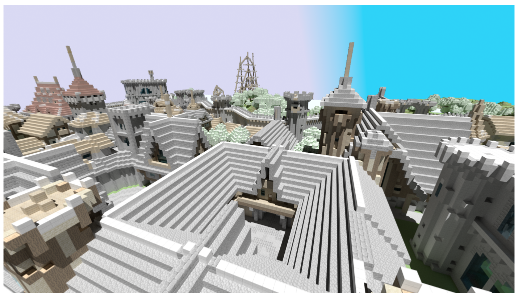
6. Conclusion and further work

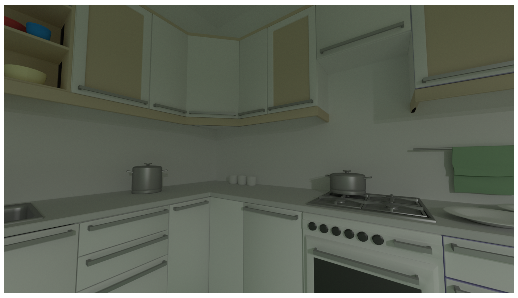
In conclusion, there is potential for equirectangular projection combined with reprojection to provide significant benefits to remote rendering systems aimed at HMDs. Images with a quality approaching that of the standard techniques can be achieved while also obtaining a significant increase in flexibility. A cost allocation must be made, however, to offset the drop in image quality caused

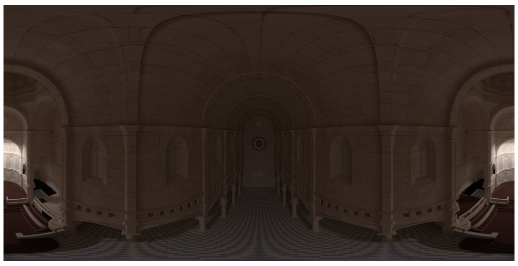
Table 1: *Scenes: (a) Sponza, (b) San Miguel, (c) Rungholt, (d) Kitchen, (e) Sibenik.*


(a)

(b)


(c)


(d)


(e)

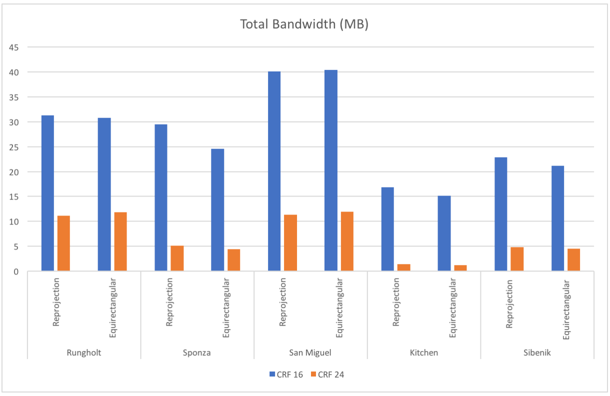


Figure 5: *Graph of the total bandwidth used for each method.*

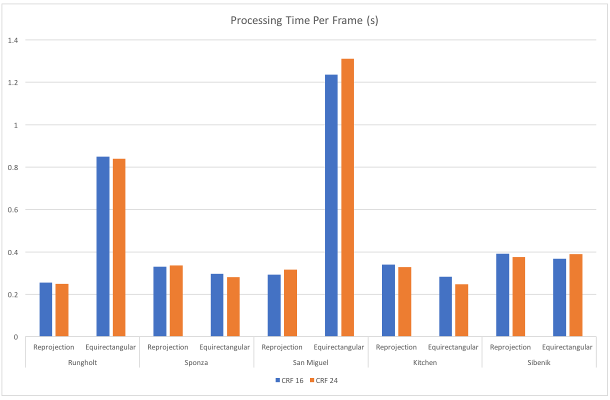


Figure 6: *Graph comparing processing time per frame by method.*

by packing more data into the same pixel space, and inefficiency in the projection chosen.

Further research into this area could serve to establish a direct relation between the resolution and/or bitrate required to attain parity of quality between a 360° projection and a regular perspective view. With this established, 360° rendering and reprojection could be compared directly with the processing cost and latency of rendering two full stereo images.

References

[ALMM14] ARDOUIN J., LÉCUYER A., MARCHAL M., MARCHAND E.: Stereoscopic rendering of virtual environments with wide field-of-views up to 360. In *2014 IEEE Virtual Reality (VR)* (March 2014), pp. 3–8. 2

[DL92] DIZIO P., LACKNER J. R.: Spatial orientation, adaptation, and motion sickness in real and virtual environments. *Presence: Teleoperators and Virtual Environments 1*, 3 (2017/03/13 1992), 319–328. URL: http://dx.doi.org/10.1162/pres.1992.1.3.319. 1

[DRE*10] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: Adaptive image-space stereo view synthesis. In *Vision, Modeling and Visualization Workshop* (Siegen, Germany, 2010), pp. 299–306. 2

[HGAB08] HOFFMAN D. M., GIRSHICK A. R., AKELEY K., BANKS

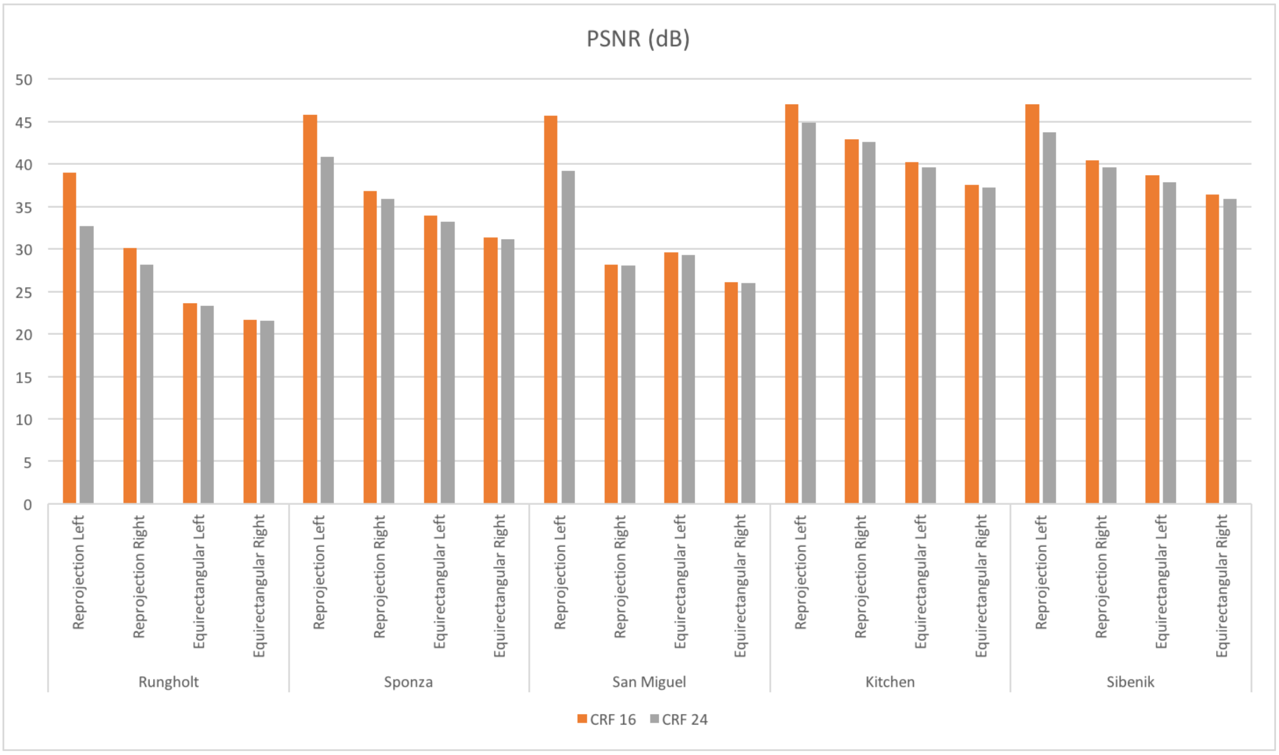


Figure 7: *PSNR results per method, separated into left (rendered) and right (reprojected).*

M. S.: Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision 8*, 3 (2008), 33. URL: +http://dx.doi.org/10.1167/8.3.33, arXiv:/data/journals/jov/932853/jov-8-3-33.pdf, doi:10.1167/8.3.33. 1

[HTG08] Huynh-Thu Q., Ghanbari M.: Scope of validity of psnr in image/video quality assessment. *Electronics Letters 44*, 13 (June 2008), 800–801. doi:10.1049/el:20080522. 4

[Kaj86] Kajiya J. T.: The rendering equation. In *ACM Siggraph Computer Graphics* (1986), vol. 20, ACM, pp. 143–150. 1

[Lev95] Levoy M.: Polygon-assisted jpeg and mpeg compression of synthetic images. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 21–28. URL: http://doi.acm.org/10.1145/218380.218392, doi:10.1145/218380.218392. 1

[MB95] McMillan L., Bishop G.: Plenoptic modeling: An image-based rendering system. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 39–46. URL: http://doi.acm.org/10.1145/218380.218398, doi:10.1145/218380.218398. 2

[McG11] McGuire M.: Computer graphics archive, August 2011. http://graphics.cs.williams.edu/data. URL: http://graphics.cs.williams.edu/data. 4

[PKW11] Pece F., Kautz J., Weyrich T.: Adapting standard video codecs for depth streaming. In *Proceedings of the 17th Eurographics Conference on Virtual Environments &#38; Third Joint Virtual Reality* (Aire-la-Ville, Switzerland, Switzerland, 2011), EGVE - JVRC'11, Eurographics Association, pp. 59–66. URL: http://dx.doi.org/10.2312/EGVE/JVRC11/059-066, doi:10.2312/EGVE/JVRC11/059-066. 2, 3

[WBSS04] Wang Z., Bovik A., Sheikh H., Simoncelli E.: Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on 13*, 4 (April 2004), 600–612. doi:10.1109/TIP.2003.819861. 4

[WKC94] Wallach D. S., Kunapalli S., Cohen M. F.: Accelerated mpeg compression of dynamic polygonal scenes. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 193–196. URL: http://doi.acm.org/10.1145/192161.192198, doi:10.1145/192161.192198. 2

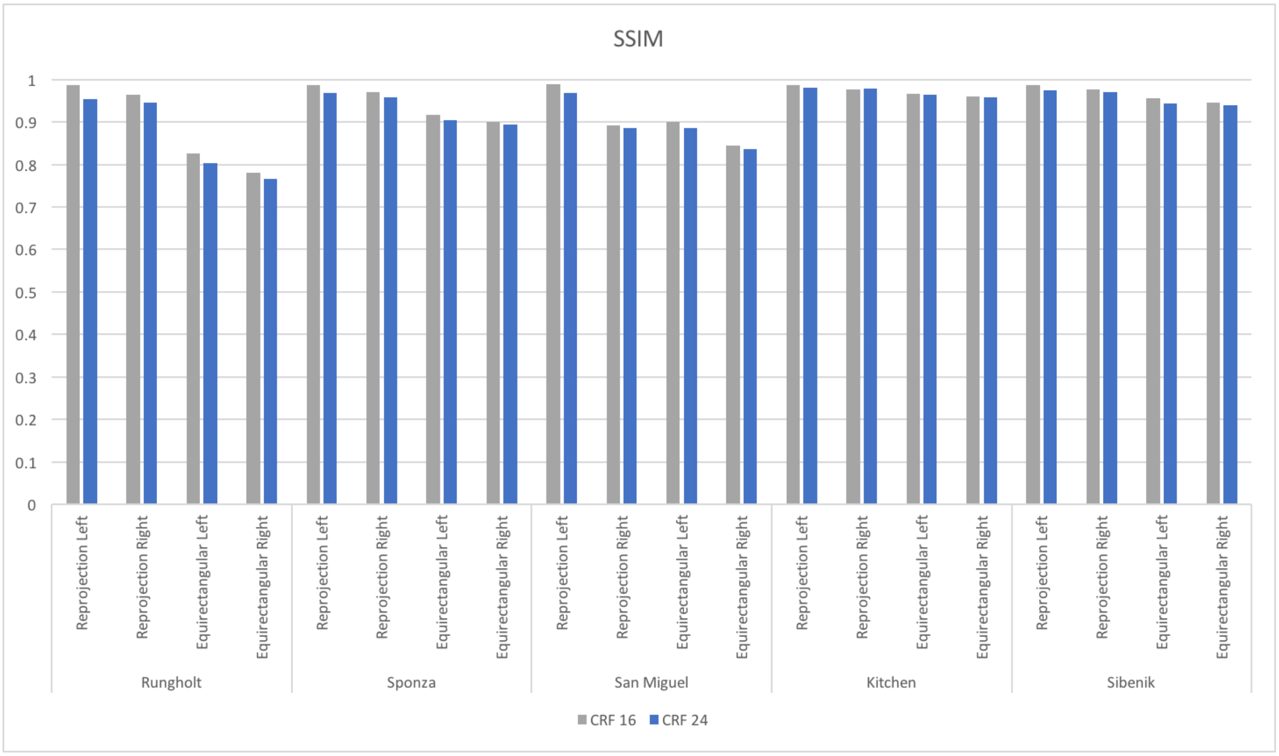


Figure 8: *SSIM results per method, separated into left (rendered) and right (reprojected).*

Table 2: *Cost in bandwidth for each method, broken down into the primary view and the supplementary scaled down render for filling in gaps. (MB per 150 frame sequence)*

		CRF 16	CRF 24
Rungholt	Reprojection Left	28.3	9.83
	Reprojection Right (small)	2.97	1.28
	Equirectangular Left	27.9	10.5
	Equirectangular Right (small)	2.85	1.33
Sponza	Reprojection Left	27.7	4.61
	Reprojection Right (small)	1.77	0.487
	Equirectangular Left	22.9	3.88
	Equirectangular Right (small)	1.69	0.502
San Miguel	Reprojection Left	37.1	10.3
	Reprojection Right (small)	2.94	1
	Equirectangular Left	37.4	10.9
	Equirectangular Right (small)	2.97	1.08
Kitchen	Reprojection Left	16.3	1.22
	Reprojection Right (small)	0.567	0.208
	Equirectangular Left	14.7	1.05
	Equirectangular Right (small)	0.438	0.151
Sibenik	Reprojection Left	21.1	4.4
	Reprojection Right (small)	1.79	0.458
	Equirectangular Left	19.2	4.05
	Equirectangular Right (small)	1.95	0.492