Virtual Reality Motion Parallax with the Facebook Surround-360

David Lindell and Jayant Thatte
EE 367
{lindell, jayantt}@stanford.edu

Abstract

Interest in acquiring and viewing natural scenes in full 360-degree immersion has led to the development of camera rigs which capture images for 360-degree stereo rendering. Most such platforms capture a scene for viewing from a single vantagepoint and do not incorporate motion parallax. Methods for scene acquisition which do support motion parallax, such as concentric mosaics or free viewpoint rendering, can be complicated to implement or result in a large data format that is difficult to compress or stream. We demonstrate motion parallax rendering using data acquired from the Facebook Surround-360 camera rig and depth-augmented stereo panoramas. Rendered views demonstrate motion parallax for horizontal head motion within a range of 24 cm. We also demonstrate real-time rendering of head-motion parallax from a synthetically generated depth-augmented stereo panorama.

1. Introduction

Growing interest in virtual reality has led to the development of camera rigs and smaller handheld cameras which capture images for rendering immersive 360-degree views of natural scenes. Ideally, the virtual reality viewing experience would incorporate depth cues such as binocular disparity, correct handling of occlusions, accomodation, and motion parallax. When depth cues are omitted or the virtual reality experience otherwise conflicts with what is naturally expected, immersion is reduced and discomfort can occur [1].

Most current systems for 360-degree capture of real-world scenes support rendering of stereo views from a single vantagepoint. That is, while the viewer can view an entire surrounding scene, the view is invariant to any lateral or vertical head motion. Typically such systems consist of a ring of outward facing cameras, such as the Google Jump [2], or the Facebook Surround-360[1]. Other designs using rotating cameras, and prisms or mirrors have also been devised [3–5], but similarly constrain the viewer to the single vantagepoint, and so do not incorporate motion parallax.

While techniques which support motion parallax rendering exist, scene acquisition is more complicated than using a camera rig, and the data format from which views are rendered can be large and difficult to stream or compress. Such techniques include rendering from images acquired from positioning a camera on concentric circles [6], or reconstructing a 3D geometry from which to synthesize views. In each approach, a large amount of data must be stored to render the views: many images or an entire 3D geometry.

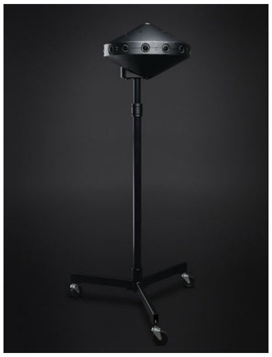

Figure 1. Image of the Facebook-360 camera rig[2].

In this paper we propose to render stereo views of a real-world scene with motion parallax using the Facebook Surround-360 camera rig and a depth-augmented stereo panorama (DASP), which consists of a texture and depth panorama for each eye [7, 8]. Although the Surround-360 conventionally supports rendering views from only a single vantagepoint, we show that processing on the captured data can produce a DASP, which supports rendering views for limited head-motion range. Thus motion parallax rendering can be achieved using a camera rig, and an easily compressible data format.

In particular, we make the following contributions

- We render views with head-motion parallax from real-world data captured with the Surround-360 camera rig.
- We create a real-time demo showing head-motion parallax with a head-mounted display using data from a synthetic DASP.
- We build a processing pipeline to render views with head-motion parallax using data from the Surround-360.

[1] https://facebook360.fb.com/facebook-surround-360/

[2] https://code.facebook.com/posts/1755691291326688/introducing-facebook-surround-360-an-open-high-quality-3d-360-video-capture-system/

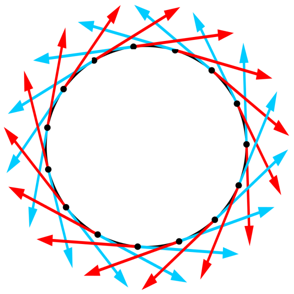

Figure 2. Illustration of ODS rays tangent to the viewing circle. The blue rays correspond to a right-eye view and the red rays correspond to a left-eye view.

2. Related Work

In many cases, the design of a camera rig for 360-degree scene capture is motivated by the intermediate format in which the data are to be stored before a view is rendered. A commonly used format is omnidirectional stereo (ODS) [9, 10] , in which left-eye and right-eye panoramas are constructed in an equirectangular projection and directly viewed by warping each to a viewing sphere. Views can also be rendered from concentric mosaics, or generated using novel view synthesis techniques. Finally, other methods construct a 3D geometry of a scene or use a depth-based representation to render views from a range of viewpoints.

2.1. Overview of Omnidirectional Stereo

Omnidirectional stereo is a compact format for stereo 360-degree viewing which consists of a left-eye and a right-eye texture panorama. The stereo image pairs are produced by casting rays from opposite sides of a circle whose diameter is the interpupillary distance. The direction of the rays lies tangent to the circle (referred to as the viewing circle), and each column in the panorama corresponds to a ray direction and thus a different point on the viewing circle. Scene points at a non-zero elevation angle are similarly projected, so the resulting panoramas approximate a cylindrical projection. The left and right eye rays are visualized in Fig. 2, where the tangential relationship is illustrated. ODS is a convenient and simple-to-use format for viewing 360-scenes because the content is already rendered into stereo panoramas and is readily viewable.

Several camera rigs have been designed to capture natural scenes for rendering in the ODS format. Among these are the Google Jump camera [2] which consists of 16 outward facing cameras and a stitching pipeline designed to produce 360-video in an ODS format. Facebook has also designed a similar camera rig, the Surround-360, which incorporates 14 outward facing cameras and upward and downward facing cameras to produce ODS panoramas with a full 180-degree vertical field of view[3]. Other methods have constructed stereo panoramas using a multi-camera system with mirrors [3], though the vertical field of view is only 60 degrees. Another systems uses a vertical stereo baseline with two cameras and parabolic mirrors to create stereo panoramas [5], and the system of Tanaka and Tachi uses a single camera and a rotating mirror and prism rig to create 360-degree stereo panoramas with a 60-degree vertical field of view. [4]. Views captured for the ODS format with these rigs do not directly support motion-parallax rendering, however.

2.2. Concentric Mosiacs

While the ODS format does not directly support motion parallax rendering, other scene representations or image acquisition techniques can be used to render 360-degree views with head-motion parallax. Views from multiple vantagepoints can be directly acquired, as with concentric mosaics [6], or indirectly acquired with novel view synthesis techniques.

Rendering with concentric mosaics uses images captured by positioning a camera around concentric circles. Views for positions within the region of concentric circles can be synthesized by stitching together slits from the captured images to approximate incoming rays corresponding to a desired viewpoint [6]. Alternatively, given a set of images from disparate viewpoints, novel views can be synthesized using deep networks [11], rather than directly acquired.

2.3. Depth-based Representations

Depth-based representations for rendering multiple views include those of structure-from-motion approaches, which rely on a moving camera, and free-viewpoint video, which uses multiple cameras to estimate 3D geometry. One approach for rendering 360-videos with 6 degree-of-freedom motion uses input monoscopic 360-degree images to estimate the 3D geometry of a scene and then synthesizes stereo views for a range of viewing positions [12]. Free-viewpoint videos use synchronized cameras to reconstruct 3D geometry and enable multiple-viewpoint viewing of a scene [13–16]. For such techniques, the data representation before rendering can have a large footprint, and so compression remains a challenge [15].

Another depth-based representation is that of DASP, which consists of a stereo pair of texture and depth panoramas. The stereo texture panoramas correspond to a conventional ODS representation, except that light rays at increasing elevation angles are mapped to eye positions in a shrinking viewing circle. This modified geometry enables points directly above the viewing circle to be represented in the panorama, making it more suitable for virtual reality than conventional ODS, where such points are not represented.

Head-motion parallax can be rendered using DASP [7] by creating the DASP representation using an extended stereo baseline, which is equivalent to increasing the diameter of the ODS viewing circle. The depth information and the extended baseline enable viewpoint rendering for eye positions within the extended viewing circle.

3. Methodology

To render views with motion parallax using the Surround-360, we process the acquired images into the DASP representation and then synthesize stereo views for a given head position.

[3] https://facebook360.fb.com/facebook-surround-360/

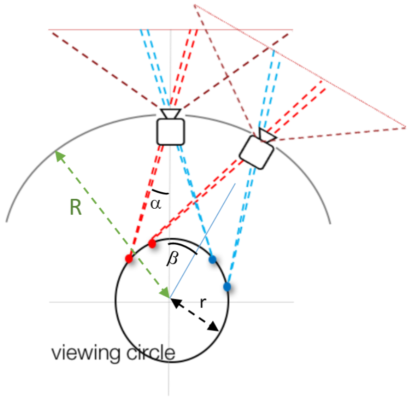

Figure 3. Illustration of the camera rig geometry from an overhead view. The rig radius is given by $R = 23$ cm, the viewing circle radius is r, the viewing radius angle is α, and the angle between the optical axes of adjacent cameras is β = 25.7 degrees.

To construct the DASP we modify the open-source Surround-360 processing pipeline[4] to use an extended viewing circle and to estimate the scene depth. Views are synthesized from the DASP by projecting the points into 3D space and then warping the point cloud onto the desired viewport.

3.1. Constructing the DASP

The Surround-360 camera rig is designed to render ODS panoramas with an interpupillary distance of approximately 6.5 cm. Due to the geometry of the camera rig, the positioning of the cameras, and the camera field of view, the viewing circle can be expanded to approximately 24 cm to enable head-motion parallax within a horizontal motion range of approximately 10 - 15 cm.

3.1.1 Expanding the Viewing Circle

The Surround-360 has $N = 14$ side-looking cameras, each with a $2\gamma = 77$-degree vertical and horizontal field of view and resolution of 2048 × 2048 pixels. An overhead view of the rig geometry is shown in Fig. 3, where the angle between adjacent camera optical axes is $\beta = \frac{360}{N} = 25.7$ degrees, the viewing radius angle is α, the viewing circle radius is r, and the camera rig radius is $R = 23$ cm. We also have that $\sin(\alpha) = \frac{r}{R}$.

Conceptually, a panorama is created by stitching together slits from images acquired from all points on the camera rig circle. Since there are only 14 points from which images are directly acquired, other camera views are synthesized by interpolating views between the nearest adjacent cameras using optical flow. Fig. 4 shows images acquired by adjacent cameras, the overlapping portion of the camera images, and the slits used for the right and left-eye panoramas. The distance between the slits depends on the viewing circle radius angle. The locations of the slits move horizontally as the angle of the optical axis of the synthesized virtual camera view shifts.

The images from adjacent cameras overlap by 2/3 and so the width of the overlapping image segments is $\frac{2}{3} \cdot 2\gamma = \frac{4\gamma}{3} = 51.3$ degrees. Since the location of the slits must lie within the field of view of the overlapping images, we have that $2\alpha + \beta < \frac{4\gamma}{3}$. Note that since we have 14 cameras with a 77 degree field of view, there is a three-fold overlap in the field of view between cameras, or $14 \times 77 \approx 3 \cdot 360$. So we also have the relationship that $2\gamma = 3\beta$. So we have that

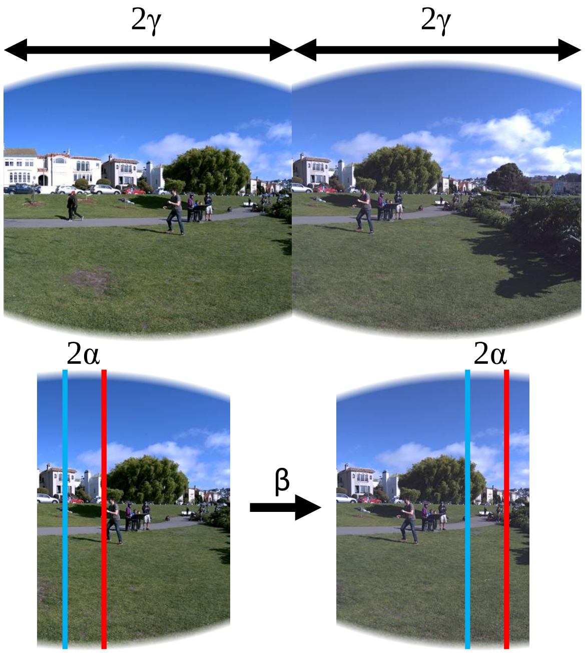

Figure 4. Images for a camera pair and slit geometry used to construct the stereo panorama. The top row shows images from adjacent side-looking cameras whose optical axes are separated by angle β. Each camera field of view is 2γ = 77 degrees. Left and right-eye panoramas are stitched together from the overlapping portions (bottom row) by assembling slits from the left image, interpolated images, and the right image, where the interpolated images represent camera views from positions between the original views. The angular distance between the right (blue line) and left-eye (red line) slits is given by 2α.

$$\alpha < \frac{\beta}{2} = 12.9 \text{ degrees.} \tag{1}$$

With the relationship $\sin(\alpha) < \frac{r}{R}$ and R = 23 cm, we have $r < 5.1$ cm. The typical interpupillary distance is 6.5 cm, so the maximum allowed motion within the viewing circle would be less than $2 \cdot 5.1 - 6.5 = 3.7$ cm. So using views from only adjacent camera pairs results in a very small range of motion parallax.

Alternatively, extra camera pairs can be used to synthesize the slit views. In that case, we use the images from the next adjacent cameras to the right and left and the so the overlapping field of view is essentially doubled to $\frac{8\gamma}{3}$. In this case, we have

$$\alpha < \frac{3}{2}\beta = 38.6 \text{ degrees.} \tag{2}$$

Using the same relationship, we are left with $r < 14.3$ cm and a more useful head motion range of approximately 15.6 cm. In practice, however, r will be smaller due to edge discontinuities and artifacts from view extrapolation with the extra

[4] https://github.com/facebook/Surround360

camera pairs. We find that $r = 12$ cm produces acceptable results.

3.1.2 Stitching Pipeline

To simplify the task of assembling the left and right eye panoramas from the acquired images, we use the open-source Facebook algorithm, making alterations as necessary to enable motion-parallax rendering. The algorithm, with our changes, is summarized as follows. Each image is warped into a spherical projection using the known focal length of the camera, and the optical flow between image pairs is computed. The camera images are warped using the optical flow to assemble image columns from the appropriate camera views into a panorama. We also pass the optical flow values themselves through this processing to construct a left-eye and right-eye flow panorama.

Some distortion results from stitching together columns from the images in the spherical projection because each image has the camera as the center of projection. Ideally, the spherical projection would use depth estimation to place the center of projection at the rig center, but the approximation is made to simplify the stitching algorithm. In practice, the distortion is small if the nearest objects to the camera rig are further than five feet away [17].

Warping with optical flow is accomplished by assuming that the angular distance between the virtual camera and real camera varies linearly with the motion of a pixel along its flow trajectory. If α_1 and α_2 are angular positions of adjacent cameras on the rig and $\xi \mid \alpha_1 < \xi < \alpha_2$ is the position of the virtual camera, a new optical flow vector is calculated for each pixel as

$$\psi' = \frac{\xi - \alpha_1}{\alpha_2 - \alpha_1}\psi, \quad (3)$$

where ψ is the original flow vector, and ψ' is the new flow vector used to warp pixels to compute the virtual camera view. The above equation corresponds to warping from the α_1 (left) camera position. Virtual camera views are also synthesized from the α_2 (right) camera position, and after the image slits are assembled for both left and right warping, they are merged using alpha blending.

Note that as the viewing radius grows larger, it is possible that the slits required to construct the panorama are located outside the field of view of the overlapping images from a camera pair, as depicted in Fig. 5. In this case, we use the next adjacent camera pair and corresponding flow to warp pixel value back to the virtual camera position between the initial camera pair. Here, we are extrapolating the camera views (not interpolating) because the virtual camera lies outside the camera pair used to compute the flow.

Slits from rendered camera positions corresponding to each camera pair are concatenated together to form a chunk of the final image panorama as illustrated in Fig. 6. For each panorama chunk, views synthesized by warping the left and the right camera images are merged together using alpha blending where slits rendered for a camera position further from the initial camera position contribute less to the combined image. Flow magnitude images are also stitched together in this fashion as shown in Fig. 6, and the flow magnitude values are used as an approximation for disparity in the image pair in order to compute scene depth.

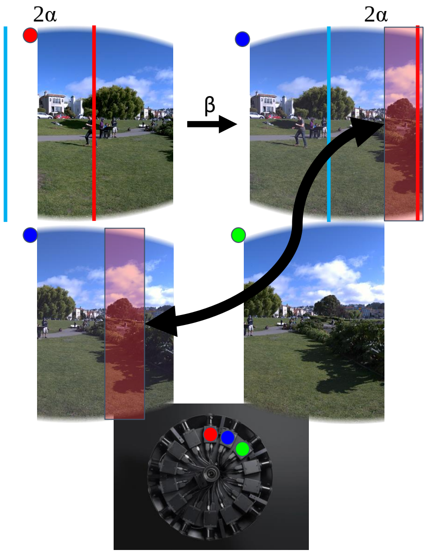

Figure 5. Illustration of panorama stitching procedure with overlapping image pairs shown for camera positions denoted by the red, blue, and green-colored circles shown in an overhead view of the Surround-360 in the bottom-most image. The slits for the left-eye (red vertical line) and right-eye (blue vertical line) panoramas move horizontally as the camera position for which views are synthesized shifts an angular distance of β. As the slit moves outside the initial field of view (into the red box), pixels from the blue-circle camera are warped using the optical flow from the blue-circle, green-circle camera pair (middle row of images) to synthesize slits from the appropriate camera views.

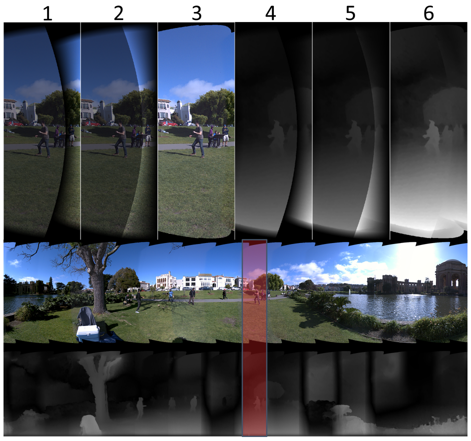

Figure 6. Illustration of how the rendered slits are combined into the final panorama. The top row of six images show concatenated slits from a left and right camera image pair where virtual camera views are synthesized from the left image (1), the right image (2), and the alpha-blended combined image (3). The corresponding flow images are also shown (4,5,6). The combined view (3) forms a chunk of the final left-eye panorama, indicated by the red box in the bottom images.

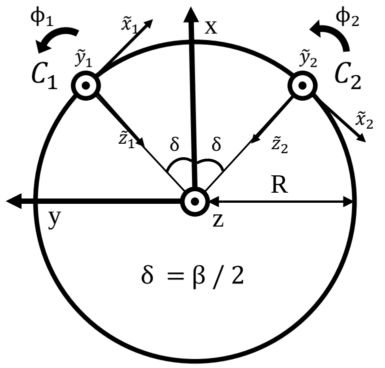

Figure 7. Coordinate system used to derive depth from disparity in a spherical projection. C_1 and C_2 are the locations of two cameras separated by an angular distance of β. A global coordinate system is defined with respect to the center of the camera rig, and the two local cartesian and spherical coordinate systems are defined.

3.1.3 Spherical Disparity to Depth

To construct the depth panorama, the disparity, given in spherical coordinates, must be converted to a depth value. We consider the problem in the coordinate system given by Fig. 7. We define two camera positions C_1 and C_2, with the local Cartesian and spherical coordinate systems given as in the figure. The global coordinate system is with respect to the center of the rig. We wish to determine x and z as a function of the elevation angle, θ, and azimuthal disparity $\Delta\phi = \phi_1 - \phi_2$ where $\phi_1 = 0$ and $\phi_2 = 0$ are defined to be in the x direction of the global coordinate system. If we assume that for each scene point location $x >> z$ and objects are relatively far away from the camera (at least five feet), we can derive that

$$\begin{aligned} x &= \frac{2R\sin\delta}{|\Delta\phi|} + \frac{R}{\cos\delta} \\ z &= \frac{\tan\theta}{\cos\delta}(x\cos\delta - R) \\ d &= \sqrt{x^2 + z^2}, \end{aligned}$$

where δ is $\frac{\beta}{2}$ and R is the rig radius. An extended derivation is provided in Appendix A.

3.2. Rendering with Motion Parallax

With the DASP representation, the left and right-eye panorama pixels can be projected into 3D space using their corresponding depth values, and a viewport can be rendered as described in [7]. A target image plane is defined perpendicular to the desired viewing direction for a given eye position, and the depth values are projected onto the image plane. The warped depth values are used to perform a lookup into the texture panoramas and the textures are also mapped to the image plane. While pixels with a smaller depth value are generally chosen as the rendered pixel value because objects with a greater depth value are occluded, such an approach results in flickering artifacts as frames are rendered for different head positions because of inaccuracies in the depth estimates. To mitigate artifacts in the image rendering due to differences in the estimated depth from the left and right eye panoramas, the final output image is calculated by smoothing out the boundary where depth values from one view transition to being greater than the other. 360-degree views with head motion can thus be rendered while mitigating frame-to-frame artifacts.

4. Results

Views from DASPs for a synthetic scene and for a scene captured with the Surround-360 are generated for varying head positions and are shown in Fig. 8 for image pairs (A)-(E). We also produce a real-time rendering of head-motion parallax from the DASP for the synthetic scene.

The synthetic scene is rendered from a DASP with a stereo baseline of 30 cm and is presented to illustrate performance where scene depth is known. The Surround-360 scene is obtained from the Surround-360 repository[5] and views are rendered from a 24-cm stereo baseline, which provides a good tradeoff between range of head motion and the amount of artifacts in the scene resulting from inaccuracies in the optical flow estimation. A quantitative evaluation of the performance of DASP for motion parallax rendering in synthetic scenes is performed in [7]. As we have no truth images for the natural scene, only a subjective analysis of the results is performed in this work.

For the synthetic scene the motion parallax effect is clearly visible from the relative motion of the vertical poles in (A), and the cones, toroids, and spheres in (B). The natural scene shows the same parallax effect, where near objects exhibit a larger horizontal shift relative to far objects for the horizontal viewpoint translation.

The natural scene also shows some artifacts from the rendering process. In (C) the specular reflections from the water surface are not accurately reproduced in the rendering because not all specular rays are captured by the camera rig. Additionally, some artifacts result near image boundaries where depth discontinuities exist due to the Surround-360 optical flow implementation, which incorporates a smoothness assumption. Such artifacts are especially noticeable near the boundary of the chair and tree in (D). Using epipolar geometry and disparity estimation rather than optical flow is a potential solution to improving the DASP depth estimates and removing or reducing such artifacts. Rendered videos are also included in the supplementary material.

The optical flow estimation in the Surround-360 rendering pipeline produces depth estimates that are inconsistent from camera pair to camera pair and which are smoothed out at sharp depth discontinuities. While careful warping of the texture panoramas can mitigate artifacts in the resulting rendered views, the extra processing makes implementation on a GPU for real-time rendering difficult. Instead, we render the synthetic scene in real-time after using the DASP to create a mesh from the scene. Each pixel in the DASP is projected into 3D space, creating a point cloud, and the point cloud is converted into a mesh format using a point cloud meshing software[6]. We

[5] https://github.com/facebook/Surround360
[6] http://sequoia.thinkboxsoftware.com/

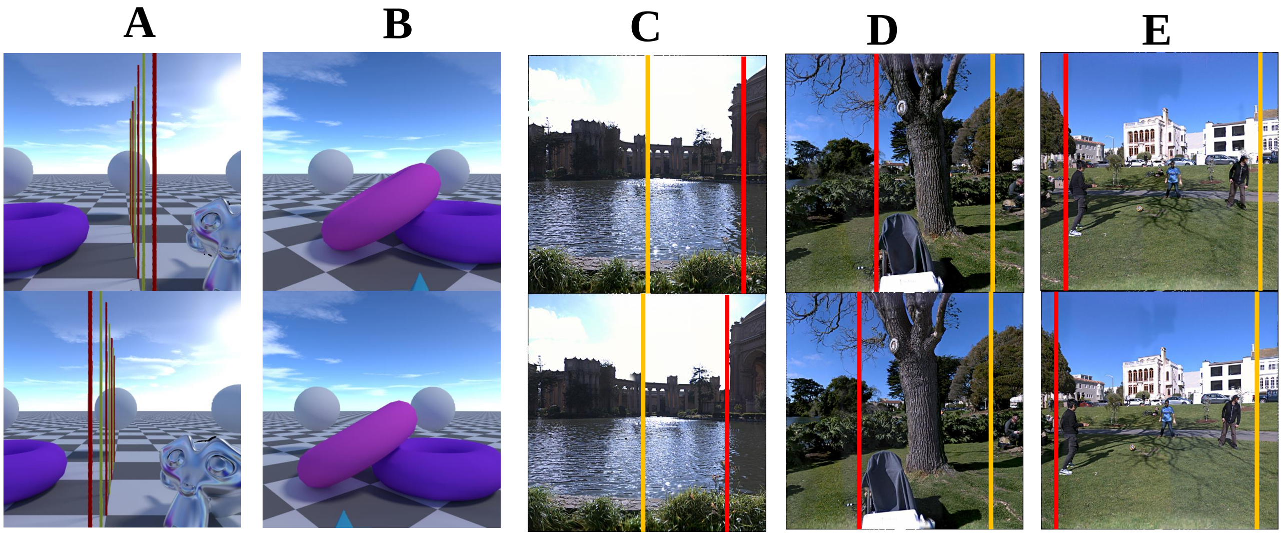

Figure 8. Monoscopic views rendered with motion parallax for head translation from left to right. Views for a synthetic scene are shown in (A) and (B), with 30 cm horizontal translation.(B) has a downward viewing angle of 25 degrees. (C), (D), and (E) show a scene from the Surround-360 with 24 cm horizontal translation. The red bars are aligned with a near object (shrub in (C), white cooler in (D), and bystander's shoes in (E)) and the yellow bars are aligned with a far object (column in (C), tree in (D) and building chimney in (E)) to illustrate the parallax effect.

then render the mesh in Unity[7] and view the scene using an Oculus Rift DK2[8] headset. We are able to achieve 90 frames per second running on an Intel Core i7 6700k with an Nvidia GTX-980 Ti graphics card. A video for a single eye view is included in the supplementary material.

5. Discussion

In summary, we demonstrate rendering with motion parallax for natural scenes acquired with a Surround-360 camera rig. The range of head-motion parallax that can be achieved depends on the rig geometry, including the camera field of view, and the radius of the circle on which the cameras are placed. From optical flow computed in a spherical projection used by the Surround-360 rendering pipeline, we compute per-pixel depth estimates and produce depth-augmented stereo panoramas, which enable rendering with motion parallax. Views are rendered from a synthetic DASP with known depth, and from a DASP generated using data from the Surround-360.

5.1. Limitations

For the proposed system, motion parallax can be rendered for head motion only along the horizontal plane. The quality of the renderings deteriorates outside the field of view of the side-looking cameras because areas near the poles do not have full stereo coverage. Additionally, the range of head motion is limited by the geometry of the camera rig, and is shown here for a range of 24 cm. Though motion parallax in this range of head motion provides a valuable depth cue, it is not suitable for wide-range scene exploration. Finally, synthesizing DASPs with a camera rig adds a layer of complexity beyond what is required for ODS panoramas in that depth estimation is required. In many cases, though, optical flow or disparity estimation is already performed for panorama stitching, so the task of depth estimation is simplified. The quality of the rendered views depends largely on the quality of the resulting depth map, however, and some artifacts are present in our results due to inaccurate optical flow.

5.2. Future Work

The results shown for the Surround-360 can be improved with a more accurate depth-estimation technique. In particular, knowledge of the rig geometry can be used for disparity estimation using the epipolar geometry. Additionally, further work can be done to quantitatively evaluate views rendered from the Surround-360 DASPs, where the ground truth data is known. For this purpose, a calibrated scene or a synthetic scene with a simulated Surround-360 acquisition process would be required. Additionally, efforts could be made to perform motion parallax rendering on videos rather than single frames.

6. Conclusion

While interest in capturing and viewing natural scenes in VR is growing, a completely convincing VR experience which incorporates all possible visual cues remains unrealized. Our proposed system is a step towards improving existing 360-degree camera rigs and the quality of the resulting VR experience. By incorporating motion parallax, the VR experience can become more comfortable and convincing.

References

[1] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks, "Vergence–accommodation conflicts hinder visual performance

[7] https://madewith.unity.com/
[8] https://www.oculus.com/rift/

and cause visual fatigue," *Journal of vision*, vol. 8, no. 3, pp. 33–33, 2008.

[2] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, "Jump: virtual reality video," *ACM Trans. Graph.*, vol. 35, no. 6, p. 198, 2016.

[3] C. Weissig, O. Schreer, P. Eisert, and P. Kauff, "The ultimate immersive experience: Panoramic 3d video acquisition," in *MMM*, pp. 671–681, Springer, 2012.

[4] K. Tanaka and S. Tachi, "Tornado: Omnistereo video imaging with rotating optics," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 6, pp. 614–625, 2005.

[5] J. Gluckman, S. K. Nayar, and K. J. Thoresz, "Real-time omnidirectional and panoramic stereo," in *Proc. of Image Understanding Workshop*, vol. 1, pp. 299–303, Citeseer, 1998.

[6] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics," in *SIGGRAPH*, pp. 299–306, ACM Press/Addison-Wesley Publishing Co., 1999.

[7] J. Thatte, J.-B. Boin, H. Lakshman, and B. Girod, "Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax," in *ICME*, pp. 1–6, IEEE, 2016.

[8] J. Thatte, J.-B. Boin, H. Lakshman, G. Wetzstein, and B. Girod, "Depth augmented stereo panorama for cinematic virtual reality with focus cues," in *ICIP*, pp. 1569–1573, IEEE, 2016.

[9] H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-directional stereo for making global map," in *ICCV*, pp. 540–547, IEEE, 1990.

[10] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: Panoramic stereo imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 279–290, 2001.

[11] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *IEEE CVPR*, pp. 5515–5524, 2016.

[12] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-DOF VR videos with a single 360-camera," in *CVPR*, 2017 (submitted).

[13] A. Smolic, "3d video and free viewpoint videofrom capture to display," *Pattern recognition*, vol. 44, no. 9, pp. 1958–1968, 2011.

[14] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Trans. Graph.*, vol. 23, pp. 600–608, ACM, 2004.

[15] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, no. 4, p. 69, 2015.

[16] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," in *ACM Trans. Graph.*, vol. 22, pp. 569–577, ACM, 2003.

[17] Facebook, *Surround-360 Manual*, 2016. `https://github.com/facebook/Surround360/blob/master/surround360_design/assembly_guide/Surround360_Manual.pdf`.

A. Depth from Spherical Disparity

Based on the geometry defined in Fig. 7, we have the following condensed equations, e.g. writing $\tilde{y}$ for $\tilde{y}_1$ and $\tilde{y}_2$, and for $\pm$ or $\mp$, the top sign indicates the C_1 coordinate system and the bottom sign indicates the C_2 coordinate system.

$$\begin{aligned}
\tilde{y} &= z \\
\tilde{x} &= \pm x \sin\delta - y\cos\delta \\
\tilde{z} &= -x\cos\delta \mp y\sin\delta + R \\
u &= \frac{\tilde{x}}{\tilde{z}} f \\
v &= \frac{\tilde{y}}{\tilde{z}} f \\
\tan\phi &= \frac{u}{f} \\
\tan\theta &= \frac{v}{\sqrt{u^2+f^2}},
\end{aligned}$$

where u and v are local pixel coordinates, ϕ is the local azimuth angle, θ is the local elevation angle, and f is the focal length. Now we can write

$$\begin{aligned}
&\tilde{x}f = u\tilde{z} \\
\Rightarrow\ & uR = (u\cos\delta \pm f\sin\delta)x + (-f\cos\delta \pm u\sin\delta)y \\
&\quad \text{assume } x >> y \\
\Rightarrow\ & uR = (u\cos\delta \pm f\sin\delta)x \\
\Rightarrow\ & u_1 = -u_2.
\end{aligned}$$

To make the problem more tractable, here we assume that the distance in x from the camera rig is much greater than the horizontal displacement in y, since objects within the camera field of view should generally follow this rule.

We can also write

$$\begin{aligned}
&v\tilde{z} = f\tilde{y} \\
\Rightarrow\ & vR = vx\cos\delta \pm vy\sin\delta + fz \\
&\quad \text{assume } x >> y \\
\Rightarrow\ & vR = vx\cos\delta + fz \\
\Rightarrow\ & v_1 = v_2 \\
\Rightarrow\ & z = \frac{v}{f}(R - x\cos\delta).
\end{aligned}$$

Now, subtracting for u_2R from u_1R gives

$$\begin{aligned}
(u_1-u_2)R &= (u_1-u_2)x\cos\delta + 2fx\sin\delta \\
\Rightarrow \Delta uR &= \Delta ux\cos\delta + 2fx\sin\delta \\
\Rightarrow x &= \frac{\frac{\Delta u}{f}R}{\frac{\Delta u}{f}\cos\delta + 2\sin\delta}.
\end{aligned}$$

At this point, we wish to find $\frac{\Delta u}{f} = \frac{u_1-u_2}{f}$. Recalling that $\phi_1 = 0$ and $\phi_2 = 0$ are defined to point in the $x = 0$ direction, we have $\frac{u_1-u_2}{f} = \tan(\phi_1 - \delta) - \tan(\phi_2 - \delta)$ which we approximate using the Taylor series representation.

$$\begin{aligned}
\frac{\Delta u}{f} &= \tan(\phi_1-\delta) - \tan(\phi_2-\delta) \\
&\approx -\tan\delta + \frac{\phi_1}{\cos^2\delta} - \tan\delta - \frac{\phi_2}{\cos^2\delta} \\
&= -2\tan\delta + \frac{\Delta\phi}{\cos^2\delta}
\end{aligned}$$

Substituting back into the expression for x yields

$$\begin{aligned}
x &= -\frac{2R\sin\delta}{\Delta\phi} + \frac{R}{\cos\delta} \\
&= \frac{2R\sin\delta}{|\Delta\phi|} + \frac{R}{\cos\delta}
\end{aligned}$$

Where the second step follows from the coordinate system geometry, where $\Delta\phi$ is always negative. The first term corresponds to a baseline divided by disparity, and the second term is an offset from the rig center along the x-axis.

From the above equations, we derived that

$$z = \frac{v}{f}(x\cos\delta - R).$$

We also had that

$$\begin{aligned}
\tan\theta &= \frac{v}{\sqrt{u^2+f^2}} \\
\tan\phi &= \frac{u}{f}.
\end{aligned}$$

Hence we also have

$$\begin{aligned}
\cos\phi &= \frac{f}{\sqrt{u^2+f^2}} \\
\Rightarrow \tan\theta &= \frac{v}{f}\cos\phi \\
\Rightarrow \frac{v}{f} &= \frac{\tan\theta}{\cos\phi}.
\end{aligned}$$

Here, each ϕ value depends on the orientation of each camera position used to synthesize the panoramas; however, if objects are relatively far away from the camera, we can assume that ϕ is on the order of $\pm\delta = \pm 12.9$ degrees, such that $\cos\phi \approx 1$. The depth is given by $\sqrt{x^2+z^2}$ and so for $x >> z$, this approximation does not affect the depth values significantly. So we approximate the expression for z as

$$z \approx \frac{\tan\theta}{\cos\delta}(x\cos\delta - R)$$