Affordable Cinematic VR Content Creation

Aniq Masood
Stanford University
amasood@stanford.edu

Neel Bedekar
Stanford University
neelb@stanford.edu

Abstract

This project aims to create a camera system that captures stereoscopic 360 degree panoramas of the real world, and a viewer to render this content in a headset, with accurate spatial sound .

1. Introduction and Motivation

Advances in head mounted displays and computer graphics have positioned virtual reality (VR) headsets to become a potential standard computing platform of the future. VR has taken a strong foothold in the gaming industry due to the relative ease of creating computer generated content. However, there is a great demand for generating immersive environments captured from the real world. Panoramic imaging is a well established field in computer vision, and many cameras and algorithms exist to capture 360 panoramas. However, a truly immersive experience in virtual reality necessitates stereoscopic 360 panoramas. Companies like Facebook and Google have created cameras such as the Surround 360 and Jump [1] that capture 360° stereoscopic video based on the omnidirectional stereo (ODS) projection model [2]. As virtual reality becomes a standard media platform, the need to generate real world content that is visually appealing and cost effective will be paramount.

This paper first outlines current VR camera systems that are used today, and how they implement the omni-directional stereo projection model. Then, a breakdown of the proposed processing pipeline and camera architecture is presented. Finally, an evaluation of the generated panoramas is presented, outlining the pros and cons of the capture method.

2. Related Work

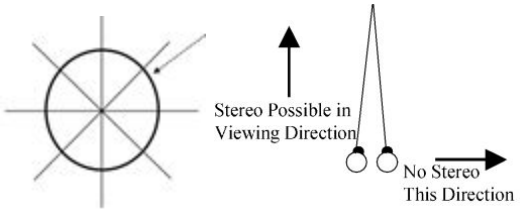
The ideally captured real world environment for virtual reality consists of two things: stereo vision, where each eye sees a viewpoint of a scene mimicking the human visual system, and a complete 360° view, where the user is able to look in any desired direction. Omni directional stereo (ODS), a projection model proposed by Peleg, Ben-Ezra, and Pritch satisfies these two criteria. In addition, they theorize many camera architectures that would possible capture scenes under this model [3], including spinning cameras and exotic arrangements of mirrors and lenses. This paper explores the application of the former, using two spinning cameras to capture a scene in full 360 degrees..

The omni directional stereo projection model has become the de-facto format for cinematic virtual reality video encoding. Current solutions capturing stereo panoramas implementing the ODS projection model fall into two categories: sequentially capturing images and the use of camera arrays. Sequentially capturing images provide the most accurate implementation of ODS, but is a slow process and prevents the ability to capture a scene at video frame rates. The use of camera arrays, such as Facebook's Surround 360 and Google's Jump camera, allows the capture of scenes at video framerates. However, these systems produce a massive amount of raw data and uses computationally expensive optical flow based stitching algorithms.

3. Methods

3.1. Omni Directional Stereo Imaging

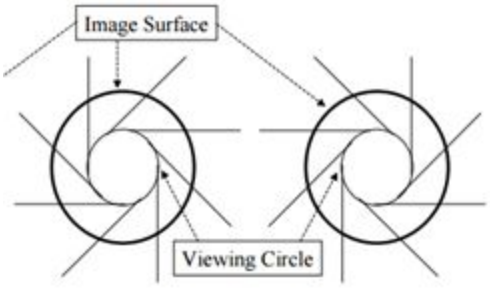
A panorama is defined as an image with a large field of view. It is constructed by capturing scene points in various directions from a single center of projection.



(a) (b)

Figure 1: (a). Projection lines perpendicular to the scene are used to construct single-viewpoint panoramas. (b). However, this model cannot be extended to stereo due to the inherent directionality of disparity.

The logical projection model for stereo panoramas would be to capture two panoramas in the same way from two different viewpoints. However, no arrangement of single viewpoint images can give stereo in all directions. This is because people perceive depth from stereo images if there is horizontal disparity between the two images when looking at the same scene point. As seen in Figure 1(b), if disparity is present in one viewpoint, it is nonexistent in the perpendicular direction. As shown in Figure 2, the ODS projection model captures disparity in all directions simultaneously by capturing scenes from multiple centers of projection. A viewing circle is defined in the middle of the scene, and scene points captured on projection lines tangent to this viewing circle are used to construct the panoramas from two different viewpoints.



(a) (b)

Figure 2: ODS Projection (a). Projections for left eye. (b). Projections for right eye.

In the context of virtual reality, the diameter of this viewing circle is the human interpupillary distance (IPD). The projections can be captured by placing two cameras on the diameter of this viewing circle and spinning them about the circle's center point, as shown in Figure 3. By capturing images sequentially and discarding the image columns not perpendicular to the viewing circle, the images can be stitched together to create the 360° panorama.

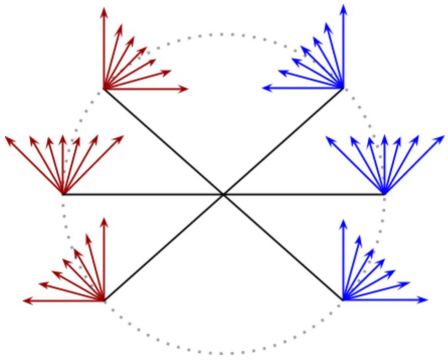


Figure 3: Capturing ODS panoramas using two rotating cameras.

3.2. Spatial Audio

To achieve the sense of presence as much as possible, one of the most crucial parts of a VR experience is that of sound. At a high level, the goal of most VR audio is to accurately represent sound information of a given scene, and allow the user to hear it in the same manner as they would in real life. Human ears have the ability to localize sound sources very precisely, and they do so with a variety of cues, namely using the direction of the sound source, and the time at which the source is heard. To accurately capture the direction, a traditional omnidirectional microphone will not suffice, as it detects sound sources with equal gain from all directions. Using direction and timing cues is important for the user to localize the sound, or to locate the position of a sound source.

One often-used method for attempting to simulate the ear's response to a sound source is through the HRTF(head-related transfer function) [4]. The HRTF is dependent on the shape of a person's head, pinna, and torso, and effectively models how the human will respond to a given sound source. HRTFs can be used to detect a binaural (both ears) sound and localize the direction the source is coming from, as well as its distance away. The impulse response given certain human cues is known as the HRIR(head-related impulse response).

While HRTFs and HRIRs are a fantastic way to simulate virtual sounds, they rely on 3 key parameters-azimuth and elevation, which signify the direction of the sound, and its distance from the viewer. Such parameters work well if simulating audio in a virtual environment, as they will be known ahead of time, but when attempting to capture real-world sound, they're much less effective. In addition, HRTFs are unique to each person and ear, which could lead to a considerable cost and barrier, if precision is key.

A different method to allow virtual reality users to accurately localize sound sources is through ambisonics. Ambisonics is a method of capturing audio in a full 360 degree range. Ambisonics represent a sound field in a special representation known as the B-format, which is entirely independent of the user's speaker setup and positioning [5].

B-format ambisonics in the first-order have 4 parameters: W, which corresponds to an omnidirectional source, and X, Y, and Z, each of which are directional sound components along each axis. Higher order ambisonics can be evaluated as well, as is shown in Figure 4 [5], but at a computational cost. Due to the additional computational overhead of computing higher order ambisonics, as well as the fact that first order ambisonics provide a good representation of sound along each axis, we opted for the first order approximation.

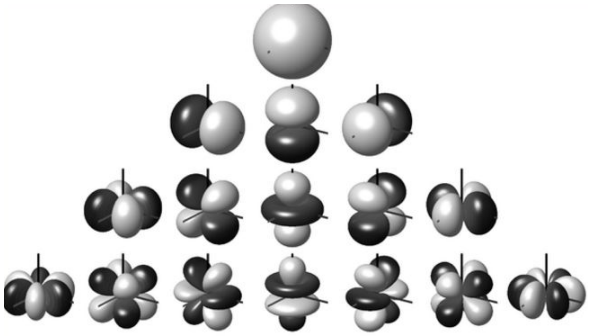


Figure 4: Zero to third order ambisonics.

There are 2 standard ways to represent a sound field in the B-format using the first order Ambisonic approximation. The first is with an ambisonic panner, or encoder, which given a source signal S, the azimuth angle θ, and the elevation angle Φ, and represents the 4 components in the following way:

$$W = S \cdot \frac{1}{\sqrt{2}}$$
$$X = S \cdot \cos\theta \cos\phi$$
$$Y = S \cdot \sin\theta \cos\phi$$
$$Z = S \cdot \sin\phi$$

Figure 5: Ambisonic encoder

The second method is to use an ambisonic microphone to capture sound sources in each of these 4 channels, but organically. We had access to an ambisonic microphone, and due to the accuracy with which it captures each channel, we opted to use it.

Given all 4 channels, an ambisonic decoder can then be used to specify an optimal setup for sound sources. We used Google Chrome's Omnitone library for Spatial Audio on the Web; Figure 6 represents the pipeline that Omnitone uses; in our case, we used a 4-channel spatial audio file, which we then applied a rotation transformation matrix to. The rotation matrix was the inverse of the rotation matrix garnered by the quaternion corresponding to the user's orientation; as a result, the rotator used the new rotation matrix to detect the new sound orientation, as well as an 8-speaker setup, to finally deliver the 2-channel audio output, one in each ear.

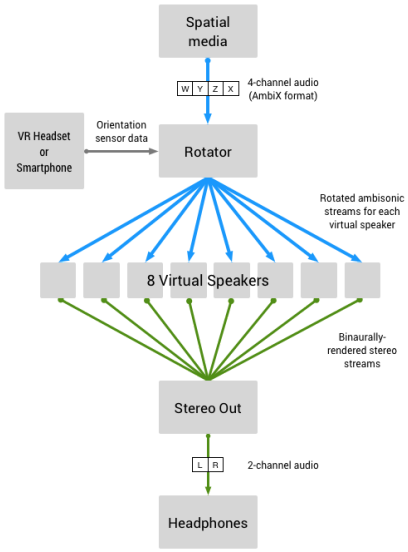


Figure 6: Omnitone Pipeline

4. Evaluation

4.1. Camera Rig

The camera rig used is shown in Figure 7. It uses two cameras connected to Raspberry Pi computers. The Raspberry Pi is a credit card sized computer that was developed in 2006 to teach children how computers work. Although its primary function is to be an educational platform to teach programming to children, hobbyists and researchers and adopted it to build small electronics projects due to its functionality and price. The Raspberry Pi 3 Model B, released in 2016, has a 1.2GHz 64-bit quad-core ARMv8 CPU running Debian Linux. It has 1GB of onboard RAM, 40 GPIO pins, a Camera Serial Interface (CSI) connector, 4 USB ports, and an Ethernet port.

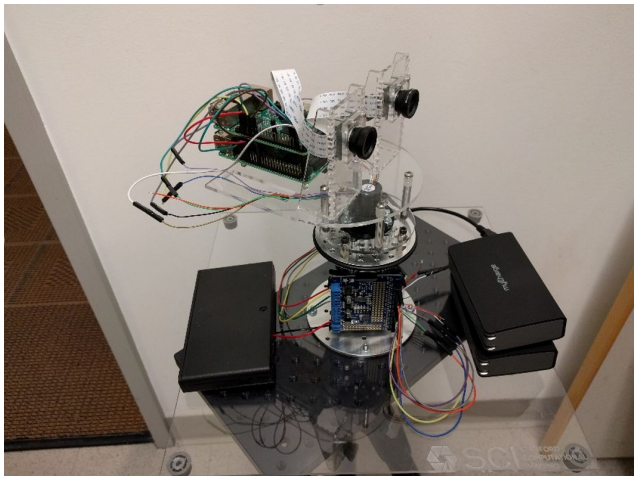


Figure 7: Camera rig used to capture panoramas.

The cameras use a Sony IMX219 8-megapixel sensor, with a square pixel size of 1.12 µm. It has an onboard lens with fixed focus, and a maximum picture resolution of 3280 x 2464 pixels. The cameras are rotated 90 degrees such that the maximum field of view and pixel resolution are in the vertical direction. Additionally, wide FOV lenses are added to capture as much of the scene vertically as possible. They are separated by 6.4 cm corresponding to the average human interpupillary distance. They are also toed in such that the zero-disparity point is roughly 1 meter away from the camera.

The electronics are mounted on a rotating platform, controlled by a stepper motor. The motor step size per successive frame can be changed to construct a more or less dense set of input images that contribute to the panorama.

4.2. Microphone

Figure 8: Zoom H2n spatial audio microphone.

The microphone used to capture spatial audio is the Zoom H2n. It is a portable, handheld microphone that is capable of capturing four channel surround sound audio. The first order ambisonics audio of the scene is captured in one WAV file that includes omni, left/right, and front/back tracks. It is the microphone used on the Google Jump camera platform and is recommended for content creators who wish to showcase their work using Youtube's VR rendering.

4.3. Headset and Viewer

To render the stereo projection onto each eye, we decided to construct a sphere, serving as a photosphere, onto which we pasted our panoramic texture, which was padded using a set of parameters from the ODS model, which was important to prevent distortion. As for how to update the image based on the user's orientation, there were two options: to rotate the cameras, or to rotate the sphere that contained the image texture. We decided to opt for the latter approach, because based on the ODS model, each eye is at an offset, of the interpupillary distance divided by 2, from the center of the sphere. This means that rotating the cameras based on an orientation would require rotating them around the origin, which is computationally far more expensive than keeping the cameras motionless and simply rotating the mesh that holds the sphere.

Every Three.JS object has a property named "matrix", and conveniently, setting this can update the translation, rotation, or scale of any object. Because translation and scale of the spheres could remain constant, all that was required was to create a rotation matrix based on the quaternion, stipulating the orientation, and then take the transpose of the resulting matrix.

Once the view was discerned, it just had to be displayed in a simple stereo rendering, with half of the screen representing each eye. Figure 9 shows a screenshot of the rendered VR view.

Figure 9: Rendered view, split screen for each eye

5. Discussion

The proposed method is effective in capturing static scenes for cinematic virtual reality. It captures

Figure 10: Output Panoramas. Top: left eye. Bottom: right eye.

stereoscopic panoramas with spatial audio at a fraction of the price and minimal post processing compared to current implementations. Some captured panoramas using the camera rig are shown in Figure 10. The actual audio at the time of the image capture from the first scene is rendered with the scene, and moves around the user when they rotate their head when using the viewer.

Although this method is effective, there are some limitations to our approach. Since the panorama is captured using images captured sequentially, any changes that occur over time such as movement in the scene or differences in illumination will present themselves as stitching artifacts in the panorama, taking away from the level of immersion the user experiences in the scene. These effects can be minimized by taking a video and extracting the panorama from the resulting frames, allowing the cameras to rotate at a faster speed. However, this method would not be able to record stereo panorama video, as video frame rates would require the cameras to spin fast enough to cover 360 degrees while capturing 30 frames per second.

In addition, the vertical field of view of the panoramas is limited by the lenses used. When rendered in the viewer, areas of the scene not captured by the cameras are padded with black pixels to fill the space. The full 180 degree vertical FOV can be captured using fisheye lenses. If this is done, the images would be noticeably warped at the vertical extents, since the cameras are offset from the center of rotation and would trace out a circle at the zenith and nadir. However, this could easily be rectified in post processing.

When viewing the scenes in the viewer, there is a visible pincushion distortion near the edges due to the lenses in the headset. Future work would include adding a barrel distortion correction to the viewer to rectify this.

Finally, for a truly compelling audio experience, spatial audio with higher order ambisonics are needed to more accurately localize sound. First order ambisonics provides the necessary effect when the user rotates their head, but having a higher fidelity of localization would vastly improve the user's experience.

References

[1] Anderson, R., Gallup, D., Barron, J.T., Kontkanen, J., Snavely, N., Hernandez, C., Agarwal, S., and Seitz. "Jump: Virtual Reality Video". *ACM Trans.Graph(SIGGRAPH Asia)*, 35, 6, 198:1-198:13

[2] S. Peleg, M. Ben-Ezra, and Y.Pritch. "Omnistereo: Panoramic Stereo Imaging", *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 23, No. 3, pp. 279-290, March, 2001.

[3] S. Peleg, Y.Pritch, and M. Ben-Ezra. "Cameras for Panoramic Stereo Imaging", *n Proc. IEEE Computer Vision and Pattern Recognition (CVPR),* Hilton Head Island, South Carolina, I:208-214, June 2000

[4] Algazi, V. Ralph, et al. "Approximating the head-related transfer function using simple geometric models of the head and torso." *The Journal of the Acoustical Society of America* 112.5 (2002): 2053-2064.

[5] Moreau, Sébastien, Jérôme Daniel, and Stéphanie Bertet. "3D sound field recording with higher order ambisonics–Objective measurements and validation of a 4th order spherical microphone." *120th Convention of the AES*. 2006.