# SpinVR: Towards Live-Streaming 3D Virtual Reality Video

ROBERT KONRAD*, Stanford University
DONALD G. DANSEREAU*, Stanford University
ANIQ MASOOD, Stanford University
GORDON WETZSTEIN, Stanford University

Fig. 1. We present Vortex, an architecture for live-streaming 3D virtual reality video. Vortex uses two fast line sensors combined with wide-angle lenses, spinning at up to 300 rpm, to directly capture stereoscopic 360° virtual reality video in the widely-used omni-directional stereo (ODS) format. In contrast to existing VR capture systems, no expensive post-processing or complex calibration are required, enabling live streaming of high quality 3D VR content. We capture a variety of example videos showing indoor and outdoor scenes and analyze system design tradeoffs in detail.

Streaming of 360° content is gaining attention as an immersive way to remotely experience live events. However live capture is presently limited to 2D content due to the prohibitive computational cost associated with multi-camera rigs. In this work we present a system that directly captures streaming 3D virtual reality content. Our approach does not suffer from spatial or temporal seams and natively handles phenomena that are challenging for existing systems, including refraction, reflection, transparency and speculars. Vortex natively captures in the omni-directional stereo (ODS) format, which is widely supported by VR displays and streaming pipelines. We identify an important source of distortion inherent to the ODS format, and demonstrate a simple means of correcting it. We include a detailed analysis of the design space, including tradeoffs between noise, frame rate, resolution, and hardware complexity. Processing is minimal, enabling live transmission of immersive, 3D, 360° content. We construct a prototype and demonstrate capture of 360° scenes at up to 8192 × 4096 pixels at 5 fps, and establish the viability of operation up to 32 fps.

CCS Concepts: • **Computing methodologies** → *Computational photography*; *Image processing*; *Virtual reality*; • **Hardware** → *Displays and imagers*; *Electro-mechanical devices*;

Additional Key Words and Phrases: Virtual Reality, Omnidirectional Stereo, Computational Photography, Real-Time

**ACM Reference Format:**
Robert Konrad, Donald G. Dansereau, Aniq Masood, and Gordon Wetzstein. 2017. SpinVR: Towards Live-Streaming 3D Virtual Reality Video. *ACM Trans. Graph.* 36, 6, Article 209 (November 2017), 12 pages. https://doi.org/10.1145/3130800.3130836

## 1 INTRODUCTION AND MOTIVATION

There is a recent trend toward high-quality VR content creation using 3D panoramic VR cameras. These cameras offer substantial benefits in terms of realism and immersion, and are increasingly accessible with multiple commercial options available from GoPro, Google [Anderson et al. 2016], Jaunt[1], Facebook[2] and others. Live streaming of 360° video is also gaining attention, with recent live broadcasts including an orchestral performance[3] and a NASA rocket launch[4]. With future applications in sports, theatre, telemedicine, and telecommunication in general we envision live-streaming cinematic VR being critical to adoption of VR by the general community.

*Equal contributions



[1] http://jauntvr.com
[2] http://facebook360.fb.com/facebook-surround-360/
[3] https://www.youtube.com/watch?v=SsKMYu0Z868
[4] http://www.ulalaunch.com/360.aspx

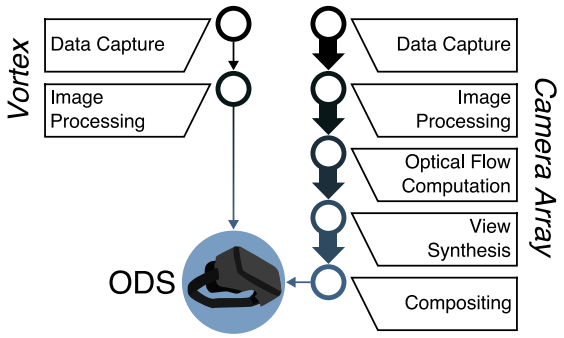


Fig. 2. Comparison of processing pipelines for Vortex and camera array based systems. Current-generation virtual reality (VR) camera rigs (right) record high-resolution videos from multiple cameras, requiring extensive processing. This includes optical flow and view synthesis, ultimately discarding much of the raw data, as indicated by the reduction in arrow width, and finally converting into the omni-directional stereo (ODS) format. In contrast, Vortex (left) captures only the rays of light required by the ODS format, enabling a low computational-cost processing pipeline.

There is, however, an important gap in quality between live streaming and recorded VR content. Multi-camera rigs require expensive optical flow or depth estimation for seamless stitching and rendering, and this can be prohibitively costly for live applications. For example, content capture by Google's Jump VR camera is reported to require 75 seconds of processing per frame for VR content [Anderson et al. 2016]. Live broadcast is therefore mostly restricted to non-stereo 2D panoramic content, lacking depth information, and sometimes suffering from seams due to the depth dependence inherent to the stitching problem. A few live streaming 3D cameras have recently become available, but these require careful calibration that can drift over time, and their real-time stitching algorithms can break in the presence of close-up, reflective and transparent objects.

We present Vortex, a live streaming VR system that directly records in the computationally and bandwidth-efficient ODS format. ODS is an established format that is widely supported by VR displays and streaming pipelines [Ishiguro et al. 1990; Peleg et al. 2001], and by directly capturing in this format, as depicted in Figure 1, our system is capable of streaming live content in high quality over a 360° field of view (FOV) without seams and in 3D.

Building on early ideas of spinning slit cameras [Bourke 2010; Peleg et al. 2001], we present an architecture capable of efficiently streaming VR videos. The proposed system requires no expensive processing and natively handles scenes that challenge camera rigs, including nearby objects, thin and repetitive structures, occlusions, transparent and translucent surfaces, and speculars and refractive objects [Anderson et al. 2016]. A depiction of the computational simplification afforded by our approach is shown in Figure 2.

We implement a prototype using two spinning line scan cameras interfaced via a slip ring to a host computer. This is, to our knowledge, the most computationally efficient architecture reported to date for streaming live 3D VR videos. We show live capture of 360° horizontal by 175° vertical scenes at up to $8192 \times 4096$ pixels at 5 fps, and establish the viability of operation up to 32 fps.

Key contributions:

- We describe a computationally-efficient live-streaming ODS architecture
- We demonstrate high-quality capture and seamless rendering free from the visual artifacts typical of multi-camera rigs
- We identify a new form of distortion inherent to the ODS format, and demonstrate an efficient way of correcting it
- We explore mechanisms for exploiting perceptual saliency to optimize user experience under camera and network bandwidth constraints
- We include a detailed design space analysis including tradeoffs between noise performance, resolution, and frame rate

*Limitations.* Vortex requires mechanically moving parts and appropriate safety measures. As in spinning LiDAR systems, we believe the benefits of the proposed approach justify the challenges associated with its mechanics. Spinning line sensors present a tradeoff between horizontal image resolution, video frame rate, and exposure time (i.e. noise), which we analyze in detail in Section 3.3. As with most VR cameras, our system provides only horizontal parallax. Our prototype is bulky compared to some multi-camera VR camera rigs, and we discuss strategies for system miniaturization in Section 5.

## 2 RELATED WORK

*Panoramic and ODS Imaging.* Panoramic stitching via sequential image capture and alignment is a well-explored area in computer vision [Brown and Lowe 2007; Szeliski 2010]. The challenge for VR panoramas is that stereoscopic depth cues should be supported for all possible viewing directions, which is not possible with conventional panoramas. Light field capture offers both depth cues and 6-degree-of-freedom movement, and recent work employing spherical lenses has demonstrated wide-FOV light field capture through a single lens [Dansereau et al. 2017]. However, the effective range of motion offered by such a compact device is severely limited, requiring multiple cameras to support any substantial virtual camera motion. Concentric mosaics [Shum and He 1999] consider a 3D slice of the plenoptic function that constrains camera motion to a plane but supports accurate stereo views in all directions. Native capture in this format corresponds to rotating multiple cameras through concentric circular paths, allowing substantial virtual camera motion.

ODS is a special case of concentric mosaics that uses a pair of multi-perspective panoramic images to encode all viewing directions for two eyes [Ishiguro et al. 1990; Peleg et al. 2001]. Any local viewing window of an ODS panorama conveys a perceptually convincing depiction of the captured 3D scene [Anderson et al. 2016]. In their seminal paper, Peleg et al. [2001] outlined several exotic ideas for recording ODS panoramas as well as a practical configuration that rotates a single camera around a fixed point, extracts two columns from each photograph, and sorts these into the ODS panorama pair. The algorithmic processing for this acquisition scheme was further refined in MegaStereo [Richardt et al. 2013], including methods for correcting ODS input data for hand-held cameras as well as an optical flow implementation for upsampling angular input resolution.

Huang and Hung [1998] proposed a setup comprising a slowly rotating camera pair. The ODS format was not used in that work and

image warping techniques introduced artifacts for synthesized viewpoints that did not coincide with the captured locations. The synchronized multi-camera systems presented by Weissig et al. [2012] and Chapdelaine-Couture and Roy [2013] experience the same problems. Similar to our system, Bourke [2010] used a continuously rotating pair of cameras to record an ODS panorama. However, conventional 2D sensor logic requires the entire image to be read out, locally stored, and in most cases also transmitted to the host computer. Thus, the communication link between sensor and host computer becomes the bottleneck and places a fundamental limit on sensor frame rates. In contrast to Bourke's approach, we use 1D line sensors that optimize the readout rate of the sensor. Conceptually, within the time it takes to read a conventional 2D sensor image, our line sensors can sweep through the entire 360° panorama. Note that all systems discussed so far are only capable of capturing static scenes.

*Dynamic Omni-Directional Stereo Imaging.* Much effort has gone into developing ODS capture systems for dynamic scenes due to the increased sense of immersion that these provide. The system described by Tanaka and Tachi [2005] is capable of achieving video rates by rotating optics (prism sheets, polarizers, and a hyperboloidal mirror) at high speed and capturing directly into the ODS format. However, the image quality of this system was low. Single shot, single camera ODS panorama capture with conventional 2D sensors were described by Peleg et al. [2001] and implemented by Aggarwal et al. [2016]. The combination of complicated mirrors and/or lenses has made it either impossible to fabricate these systems thus far or, as is the case for Aggarwal's system, significantly impacts image quality.

Over the last few years, synchronized multi-camera array systems have been advertised and adopted by many consumer electronics companies. Systems comprising of two spherical panorama cameras have been proposed [Matzen et al. 2017], taking advantage of readily available consumer-grade electronics like the Ricoh Theta S or Samsung Gear 360, but do not offer true stereo views in all directions. Most ODS capture systems place cameras radially on a rig (16 in the case of the Google Jump [Anderson et al. 2016]) and use view interpolation to provide intermediate views between adjacent cameras. The biggest challenge for these systems is the massive amount of captured data and impractical processing requirements. For example, Facebook's surround 360 uses 17 high-resolution machine vision cameras recording at 30 frames per second. This system generates 17 Gb/s of raw data that is streamed, via fiber link, to a large hard drive array for storage and offline processing. The most expensive processing step is optical flow between every pair of adjacent cameras, and over multiple temporally adjacent frames to allow smooth view interpolation. Anderson et al. [2016] report a compute time of 75 seconds for each VR video frame on a single machine using their highly-optimized algorithm, or 75 compute days for 1 hour of video. Even using cluster computing and hardware acceleration, live streaming is presently out of reach with these systems.

Live-streaming VR video camera rigs have also begun to emerge, including Intel's True VR offering hemispherical 3D video, and the Z-cam V1 Pro supporting 360-degree 3D viewing using NVIDIA's real-time stitching API[5]. These systems offer live streaming by leveraging careful camera calibration and GPU-accelerated stereo image stitching [Adam et al. 2009]. However, the performance of such systems can suffer in the presence of close-up, reflective and transparent objects, and calibration can drift over time and adversely affect performance.

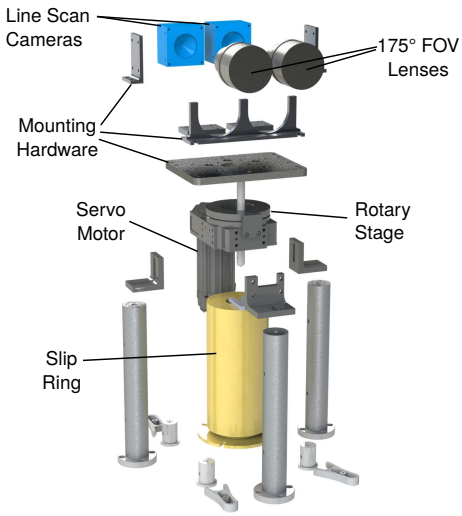


Fig. 3. The Vortex system comprises two line scan cameras, wide field of view lenses, a servomotor-driven rotary stage, and a slip ring for electrically coupling to the rotating components.

To overcome these limitations, Vortex uses a mechanically moving design that does not suffer from the sensitivity to fabrication tolerances associated with exotic optical systems, requires no computationally expensive interpolation or stitching, is robust to challenging scenarios, and does not require significant calibration.

*Spinning Cameras and Displays.* Rotating camera systems have been popular [Peleg et al. 2001; Shum and He 1999] and first attempts to capture dynamic scenes were presented by Tanaka and Tachi [2005]. Line sensors are commonly used in machine vision and have also been proposed for simplifying the stereo correspondence problem in 3D scene reconstruction [Benosman et al. 1996; Murray 1995]. We are the first to build a VR video camera using spinning line sensors.

Finally, light field displays using spinning parts have been very popular for 3D image presentation [Batchko 1994; Cossairt et al. 2007; Jones et al. 2007; Tanaka et al. 2004]. These displays and the success of spinning LiDAR sensors commonly used by autonomous vehicles make us confident that mechanically moving parts are a viable direction for domain-specific imaging and display systems.

## 3 SYSTEM DESIGN

### 3.1 Hardware Considerations

Vortex, depicted in Figure 3, comprises two line scan cameras mounted on a rotating platform. A key motivation for this spinning camera

[5]https://developer.nvidia.com/vrworks/vrworks-360video

design is to measure only the information required for ODS, eliminating extraneous data and processing. At every exposure of the line sensor we capture the rays of light tangential to the capture circle, and store them directly into one column of the output ODS panorama. The resulting algorithmic simplification is depicted in Figure 2, contrasting the processing required for a conventional camera array (right) with that required for Vortex (left). This level of simplification is what allows the Vortex architecture to deliver live streaming VR video.

The proposed architecture is essentially as described by Shum and He [1999], but there are important design considerations that must be overcome to make the idea practical.

*Data Offlink.* A key design consideration is offloading data from spinning cameras. With typical line scan cameras offering line rates of 45 kHz and $4096 \times 2 \times 12$-bit pixels for each line, there is sufficient data to saturate multiple GigE channels. We partially mitigate this by employing a dedicated GigE channel for each camera, but must nevertheless strike a balance between resolution, frame rate, bit depth and network bandwidth. We analyze these tradeoff in detail in Section 3.3. For electrical connection to the spinning cameras we employ a slip ring capable of hosting multiple GigE channels. In future we anticipate adopting an optical off-link, using the slip ring only for delivering power.

*Baseline and Vergence.* Two related design considerations are the baseline and vergence of the cameras, illustrated in Figure 4. Baseline $b$ is the distance between cameras' entrance pupils and determines the native interpupilary distance (IPD) of the captured imagery. Having a larger IPD gives more depth information, but too large a departure from the user's natural eye spacing may feel unnatural and result in difficulty fusing. This should not be confused for the display's IPD, which should always closely match the user's physiology.

Vergence is the distance $d$ at which the cameras' principal rays intersect. This controls which parts of the scene show zero parallax between left and right eye views, and is generally selected to maximize comfort and emphasize scene content. As seen in Figure 4 (left) parallel cameras verge at $d = \infty$, while (center) toeing in the cameras yields a closer vergence.

Post-capture, we can adjust the effective vergence by rotating the two ODS panoramas relative to each other. Figure 4 (right) shows this process for imagery captured by a toed-in camera verged to a distance $d$, being adjusted to verge at $d' = \infty$. To do this, the right eye panorama is rotated to the left to emulate a camera looking further to the right in the scene. This shifts the vergence distance further from the viewer, while rotating in the opposite direction does the converse. Note that this post-capture change in vergence also induces a small change in baseline, reducing it to $b' < b$ as shown in the figure. This effect is weak: in a typical scenario with baseline $b = 80$ mm, shifting from $d = \infty$ to 25 cm results in only a 1.25% change in baseline.

*Camera/Rotary Stage Synchronization.* The rotary stage and cameras can be synchronized in hardware or in software. In hardware solutions, a shaft encoder captures the phase of the motor, and this drives the cameras. Many line scan cameras directly accept rotary encoder signals, and camera line or frame triggers can be driven using a phase-locked loop synchronized with a magnetic switch or shaft encoder.

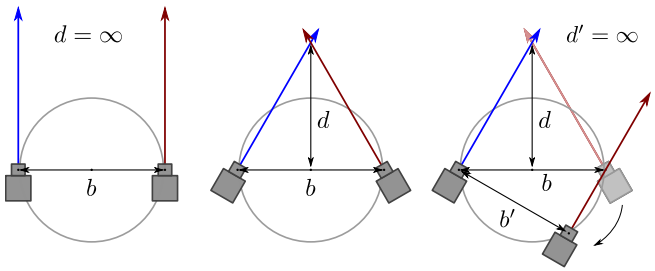


Fig. 4. Baseline $b$ and vergence $d$. (left) Parallel cameras verge at infinity. (center) Toeing in cameras results in closer vergence. (right) Vergence can also be adjusted post-capture: here imagery captured with toed-in cameras is adjusted by rotating the right-eye panorama to the left. The resulting effective shift in viewing direction is to the right, and in this example yields a vergence at infinity. Arbitrary vergence is achievable by controlling the extent of panorama rotation. This process has a a weak impact on baseline, such that $b' < b$, but only by a small amount.

In software solutions a rotary encoder signal is recorded along with free-running camera signals, and the imagery is aligned post-capture. In the absence of a rotary encoder, rotation can be estimated directly from the imagery, by feature tracking or using a simple 1D correlation method. For our hardware prototype we adopt the latter strategy.

## 3.2 Live Processing and Rendering

*Camera Calibration.* Here we introduce the various forms of distortion present in the capture system, then explain why they can mostly be ignored, allowing very simple camera calibration.

The wide-FOV lenses suffer from chromatic aberration near the extents of the FOV, and radial geometric distortion, which can be calibrated on a traditional 2D camera. Because we are using line scan cameras, these manifest chiefly as distortions near the vertical extents of the captured imagery, while radial distortion is reduced to the vertical image dimension, and does not vary horizontally. An acrylic safety enclosure introduces further optical distortion, again chiefly in the vertical direction and near the vertical extents of the image.

Further vertical distortions become apparent when the capture circle becomes larger than the viewing circle, a typical concern with camera arrays [Anderson et al. 2016]. Variations in the motor rotation rate introduce local horizontal bulging and squeezing of scene content while desynchronization between cameras and motor yield horizontal drift in the scene if not corrected for using one of the methods described earlier. Finally, errors in mechanical alignment between the cameras and the rotation platform cause further geometric error.

These sources of error tend to be very noticeable in conventional camera arrays, generally appearing as seams at the extents of each camera's FOV, as in Figure 5. This necessitates extensive depth-dependent processing to effect adjustment of the collected imagery. However, in the case of Vortex, the continuous capture and smooth

Fig. 5. A frame from a recent live broadcast 360° video, showing ghosting at the seam between camera FOVs. Stitching imagery from multiple cameras is inherently depth-dependent, making it difficult to perform well in real-time.

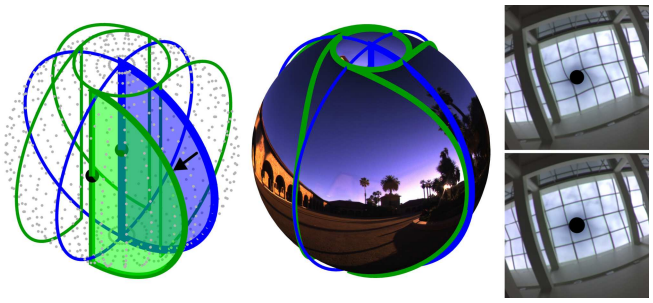

Fig. 6. Distortion near the poles: (left) a linescan camera rotating about its nodal point (blue) captures a spherical panorama, with the zenith capturing a single point for all rotations. (green) A camera offset from its center of rotation captures a distorted set of rays, with the zenith tracing out a circle. (top-right) The resulting distortion appears as a circulation about the pole. (center) We model this distortion and reverse it, with a falloff that maintains stereo information throughout the scene. (bottom-right) The dewarped image computed using an efficient 2D interpolation that is easy to carry out in real-time – the ceiling is accurately dewarped despite its 3D tented shape.

camera motion mean that no discontinuities arise, and consequently most of the sources of geometric error go unnoticed.

The processing required for Vortex is therefore very simple, and employs only a few calibrated parameters. These are the vertical field of view of the cameras and the vertical offset between left and right eye images. An additional horizontal offset can be introduced to adjust vergence, as discussed in Section 3.1. Additional parameters arise when addressing distortion near the vertical extents of the imagery, as addressed below. In all cases, we have found manual adjustment of the parameters to be straightforward and sufficient.

*Distortion Near the Poles.* ODS best approximates stereo vision near the horizontal viewing plane. At the vertical extents of the imagery, near the zenith and nadir, a noticeable warping becomes evident, an example of which is depicted in the top-right of Figure 6. This apparent circulation in the imagery follows the camera's motion about its axis of rotation.

The geometry of this distortion is depicted on the left in Figure 6. In blue the FOV of an ideal camera is depicted, covering 180° vertically and rotating about the nodal point of the lens. The zenith and nadir rays remain fixed and vertical for all camera rotations. In Vortex, the camera is offset from its center of rotation as depicted in green. This is the source of the parallax that allows stereo viewing, but it also results in an undesired shift in perspective near the poles, as the zenith and nadir rays now trace out a circle as the camera rotates.

The extent of distortion observed depends on the distance to the scene. However, because the camera is typically stationary beneath a ceiling or sky at a fixed distance, we have found that only two distances need be estimated to describe the warping, and one of them remains fixed across scenes as the camera's vertical distance to its platform does not change.

Expressions describing the distortion are derived in Appendix A. These show the distortion depends only on the vertical pixel location in the ODS images, and is independent of rotation. As such dewarping can can be carried out as two 1D interpolation, or a single 2D interpolation, all with precomputed interpolation coordinates.

Although the warping is undesirable at the poles, it is the source of depth parallax in the ODS and must therefore be maintained near the horizontal plane. As such we introduce an adjustable falloff in the correction, an example of which is depicted in the center of Figure 6. An example warped and dewarped image are shown on the right of the figure.

*Seamless Rendering.* Vortex captures data over a 360° FOV by rotating rapidly. A naive approach to rendering is to buffer up 360° of scanlines into a frame, and stream the resulting frames to a standard display. This scenario is depicted in Figure 7. At time $t = 0$ the camera is at angle $\theta = 0$, and one frame's worth of scanlines, shown in green, has been buffered up. When rendering the indicated FOV, there will be a seam near the center of the view, caused by the discontinuity between the newest ($\theta = 0$) and oldest ($\theta = -2\pi$) scanlines.

Seamless rendering can be accomplished by buffering up older scanlines beyond a single revolution, as shown in red in Figure 7. Now the requested FOV can be rendered without a seam by using the red scanlines leading up to $\theta = -2\pi$. The additonal scanlines should cover up to the maximum FOV of the rendered viewport, allowing seamless rendering for any requested view. An example of naive and seamless render for data captured using Vortex is shown in Figure 8.

While standard viewers like YouTube do not currently support the proposed method for seamless rendering, it is simple, efficient, and easy to implement. The approach imposes a maximum of one frame of display latency, and an average latency that depends on the FOV and is less than 1/2 frame for FOV < 180°. Note that seamless rendering occurs at the viewing device and so naturally supports multiple independent simultaneous viewers.

An additional form of distortion fundamental to the rotating camera design is the horizontal stretching and compression of fast-moving objects. This is similar to rolling-shutter artifacts present in most mobile phone cameras, and diminishes with increasing frame rate.

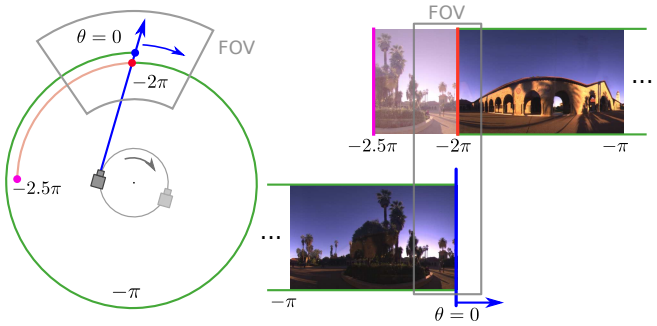


Fig. 7. Seamless rendering: (left) a rotating camera shown at time $t = 0$ and angle $\theta = 0$ (blue). The green trace shows the previous full rotation, tracing back in time and angle to $\theta = -2\pi$. The same scenario is depicted at right, with $\theta = 0$ at bottom right, and previous scanlines to the left and above. A naive rendering of the indicated FOV considers only those scanlines indicated in green, and shows a temporal seam between $\theta = 0$ and $\theta = -2\pi$. A seamless render is possible by storing older scanlines (red), up to a maximum of one complete FOV width. Seamless rendering for any viewing angle is possible by always selecting the newest contiguous stretch of stored scanlines.

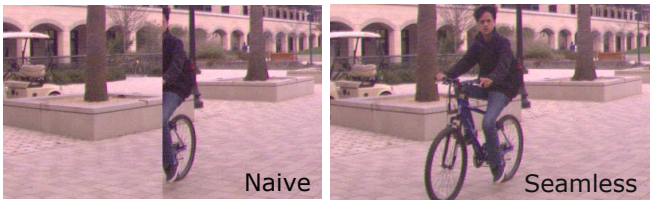


Fig. 8. Example of a naive render showing pronounced temporal seam, and seamless render employing the method depicted in Figure 7.

### 3.3 Design Tradeoffs

There is a fundamental tradeoff between spin rate, i.e. video frame rate, horizontal image resolution, and scanline exposure time. For a fixed exposure time, higher frame rates necessitate lower resolution. For fixed resolution, faster frame rates necessitate a shorter exposure time. An important system limit is therefore the ability to capture sufficient signal over short exposure durations.

To analyze the design tradeoffs in more detail, we estimate signal-to-noise ratio (SNR) for different operating conditions following the approach outlined by Cossairt et al. [2013]. We model the number of photoelectrons $\lambda$ detected by the sensor as

$$\lambda = 10^{15} \cdot (F/\#)^2 \cdot q \cdot R \cdot t \cdot \Delta^2 \cdot I, \tag{1}$$

where $F/\#$ is the f-number of the camera lens, $q$ is the quantum efficiency of the sensor, $R$ is the average reflectance of the scene, $t$ is the exposure time, $\Delta$ is the pixel size in meters, and $I$ is the illumination level in lux.

As in [Cossairt et al. 2013], we adopt an affine noise model combining signal-dependent and signal-independent components. Signal-independent read noise is Gaussian-distributed with zero mean and variance $\sigma^2_{read}$. Signal-dependent photon noise is Poisson-distributed, which we approximate as a Gaussian with mean and variance $\lambda$ – this is a good approximation when $\lambda > 10$ electrons.

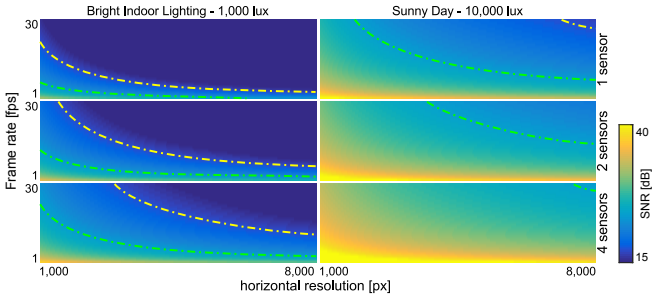


Fig. 9. Predicting SNR for varying illumination and operating modes. Thresholds for excellent and acceptable image quality, 32 and 26 dB respectively, are shown as green and yellow dotted lines. For indoor illumination (left), excellent image quality is only achieved for low frame rates or reduced horizontal resolution. Using 2 or 4 sensors significantly ameliorates the situation. For outdoor illumination (right), many modes yield excellent image quality (lower-left of the dashed green line), and nearly every mode yields acceptable quality.

The combined sensor noise variance $\sigma^2$ and the SNR in dB, are

$$\sigma^2 = \lambda + \sigma^2_{read}, \qquad \text{SNR} = 20 \log_{10} \frac{\lambda}{\sigma}. \tag{2}$$

We can now predict the SNR for given system parameters, illumination level and operating mode. We adopt parameters reflective of the prototype presented in Section 4: pixel size 7.04 µm, quantum efficiency $q = 0.7$, f-number $F/1.4$, and illumination typical of indoor and outdoor scenarios with average scene reflectance $R = 0.5$. Exposure time is dictated by horizontal resolution and frame rate. Figure 9 plots the resulting SNR estimate for configurations employing one, two and four line sensor per eye. For a given video frame rate, the multi-sensor setups allow for slower spin rates, thus longer exposure times and higher SNR.

According to the ISO standard[6], an SNR of 32.04 dB corresponds to excellent image quality, and 26 dB to acceptable image quality. These values are indicated in Figure 9 as green and yellow dotted lines, respectively. From the figure, excellent image quality is easily attainable for outdoor scenarios – all configurations to the lower left will provide an SNR that is better or equal to 32 dB. In indoor scenarios, tradeoffs in either frame rate, resolution, or sensor count must be made to achieve acceptable image quality. We evaluate these tradeoffs experimentally in Section 4.

### 3.4 Perceptually-Driven Nonuniform Sampling

Perceptual saliency provides a means to optimize user experience under camera and network bandwidth constraints. For many live streaming applications a region of interest (ROI) can be defined a priori, for example the playing field for sports events or the stage for live performance. In general applications dynamic ROIs may be employed, leveraging existing image and video saliency measures, or following specific actors. Finally, user studies have established that in general VR scenarios, gaze direction shows a strong bias towards the horizontal viewing plane [Sitzmann et al. 2016].

[6]ISO 12232:1997 Photography - Electronic Still Picture Cameras

We identify three mechanisms for exploiting saliency: camera triggering, software subsampling, and optics. Each achieves nonuniform sampling in space or time to deliver increased fidelity within an ROI, trading off lower fidelity outside the ROI to achieve higher perceptual quality for a fixed bandwidth.

Camera triggering can drive horizontal spatial and temporal nonuniform sampling. For spatial sampling the line trigger is manipulated to pack lines more densely in the ROI and less densely outside. This is easily achieved through the introduction of a simple microcontroller to drive the camera line triggers. For the same overall camera bandwidth, this results in higher fidelity in the ROI. For temporal sampling, line triggers are manipulated to skip frames for parts of the scene. Selective vertical temporal sampling is accomplished by modifying the camera's built-in ROI setting between frames, capturing a whole vertical frame only every other rotation, for example (note that not all cameras support this feature). If the baseline framerate is appropriately increased, each of these approaches increases quality within the ROI for a fixed camera bandwidth.

Software-driven approaches do not impact camera bandwidth – the link from camera to a local computer – but make optimized use of the communication channel to the viewers. Temporal and spatial subsampling are trivial to implement by selectively spatially or temporally filtering and downsampling. Further communication bandwidth optimization is possible through compression of the ODS streams, e.g. using low-latency video encoding.

Finally, optical nonuniform sampling is possible in the vertical direction through appropriate lens selection and software dewarping. The ideal sampling density, based on perceptual user studies [Sitzmann et al. 2016], is shown in Figure 10 in red. Also shown in the figure are an ideal thin lens showing an undesirable sampling density shape, and the fisheye lenses employed in our prototype coming closer to ideal. Specifically engineered lenses may more closely approach the ideal shape. For a fixed camera and communication bandwidth, such a lens delivers higher spatial resolution near the vertical center of the frame by appropriately dewarping the recorded imagery at the viewer. A simulated example of the imagery resulting from this process is depicted in Figure 11 for the sampling density shown in red in Figure 10. Note that elements near the horizontal viewing plane (green) improve in clarity compared with the uniformly sampled scene, while elements near the vertical extents (red) show decreased clarity. For this figure identical bandwidths were used for the uniform and optimized cases.

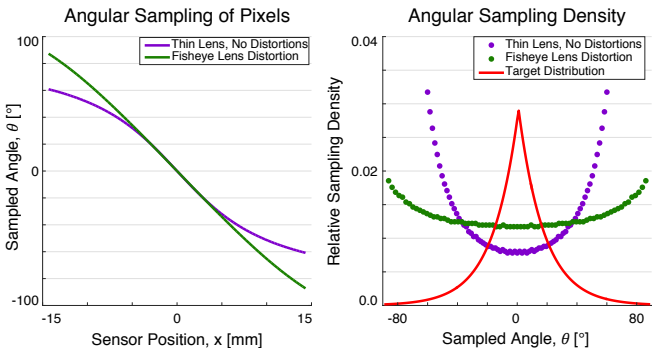


Fig. 10. Optical nonuniform sampling: perceptual studies have shown a strong perceptual bias towards the horizontal viewing plane, inspiring the ideal sampling density shown in red – note that sampling density approaches but does not reach zero near the vertical viewing directions. An ideal thin lens (purple), taken as representative of typical real-world lenses, does a poor job of approximating this, but some fisheye lenses are closer to ideal, as seen by the increased sampling rate near the horizon. A lens closer to the desired profile could be engineered for optimal results. Figure 11 shows an example of optical nonuniform sampling for increased fidelity in salient regions.

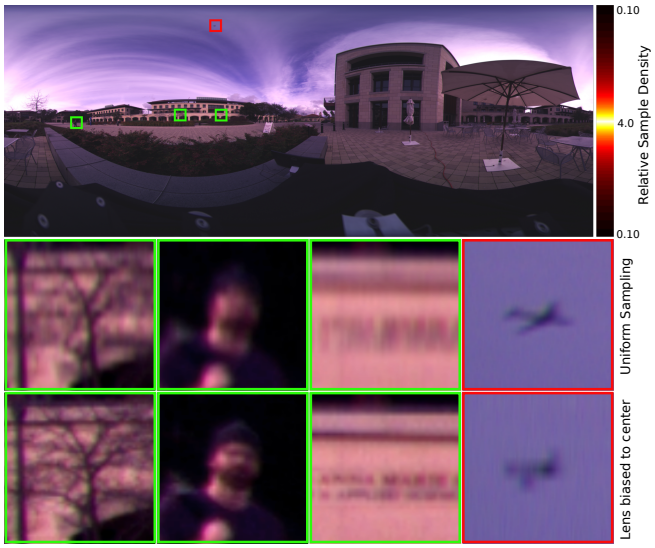


Fig. 11. Simulated optical nonuniform sampling: (top) the scene after undistortion for a lens following the ideal profile shown in Figure 10, resulting in a quadrupling of sample density near the vertical center of the frame. (bottom) Comparing uniform and optimized lenses with identical sensors, we see that nonuniform sampling increases fidelity near the vertical center of the image, trading off for a loss in clarity near the vertical extents.

## 4 RESULTS

### 4.1 Hardware Prototype

We constructed a prototype Vortex system capable of live streaming 3D content captured directly in the ODS format, with a 360° horizontal and 175° vertical FOV. The system's components are summarized in Table 1.

For experimental validation, a computer is used to initialized the cameras and motor, and record the raw data streams to hard drive. In live streaming applications these streams would be passed to a distribution system for live broadcast, and no controlling computer is necessary. For nonuniform sampling, a small microcontroller board is introduced to control triggering of the cameras, with a hall effect switch providing rotation information to the board.

Vortex's line scan cameras incorporate a 2-column Bayer color mask with alternating red and green on one column, and full green coverage on the other. Each camera continuously captures on both

Table 1. Hardware Specifications

| | |
|---|---|
| Cameras | 2 × Teledyne Linea Color GigE line scan |
| Pixel size | 7.04 µm |
| Pixel count | 2 × 4096-pixel columns |
| Line rate | 45 kHz internal, 13 kHz out |
| Lenses | Rokinon 8 mm 175° F/3.5 fisheye |
| Rotary stage | Bell-Everman SBR-50-31-234CP-DN |
| Motor | Teknic ClearPath CPM-MCPV-2341P-RLN |
| Max payload | 10 kg |
| Slipring | Moflon ME2382-P0410-S16 4x GigE |
| Max RPM | Rotary stage: 1000 rpm; slip ring: 300 rpm |
| Baseline | 80 mm |
| Toeing | None: parallel principal rays |
| Enclosure | 12 mm acrylic |

columns while rotating, and so demosaicing is simpler and yields twice the resolution across all color channels compared to a traditional Bayer pattern.

Note that though the prototype's slip ring is rated for operation up to 1000 rpm, we observed substantial signal degradation over 300 rpm resulting in dropped packets. The slip ring therefore limits our maximum frame rate to 5 fps, and we expect that by replacing this with a higher-quality slip ring or an optical off-link higher frame rates will be possible. In Section 3.3 and in the validation below we confirm the cameras can deliver up to 16 fps, and by adding two additional cameras we could deliver up to 32 fps.

We take advantage of a few important features to increase camera performance: the 12-bit pixels are quantized to 8 bits on-camera, applying adjustable digital gain and optional vertical binning. This allows a higher effective SNR and an optional halving in vertical resolution / bandwidth utilization. The cameras also offer a compression mode that improves bandwidth utilization for scenes with self-similar regions. With next-generation technologies including 10 GigE we expect the Vortex architecture to offer increasing visual fidelity and frame rates.

## 4.2 Experimental Validation

We evaluate Vortex over the operating modes summarized in Table 2. The three representative settings are "quality", "balanced", and "video", striking different balances between SNR, bandwidth and resolution. These serve as experimental validation of the analysis in Section 3.3.

Figure 12 demonstrates photographs with these settings in the respective rows. As predicted by our analysis, the illumination conditions in the outdoor scene are sufficient for high-quality video recording. Even the exposure times corresponding to a frame rate of 16.67 fps allow for an acceptable image quality (Figure 12, third row). Slight color artifacts on the metal chair for the lowest exposure are due to aliasing: we did not actually spin the sensor at $16.67 \times 60$ rpm

Table 2. Operating modes

| Setting | Quality | Balanced | Video |
|---|---|---|---|
| Horz pixels | 8192 | 4096 | 2048 |
| Vert pixels | 4096 | 4096 | 4096 |
| Line rate (line/s) | 318.6 | 4551 | 34133 |
| Exposure (µs) | 3000 | 200 | 29.3 |
| FPS | 1/25.7 | 1.11 | 16.67 |

but recorded the scene with an equivalent exposure time, thus horizontally under-sampled the scene. Aliasing would not be observed when spinning at high rates. For dim indoor lighting conditions (Figure 12, right column), only very low frame rates achieve a high image quality. Better sensors or lenses with an f-number higher than F/3.5 would improve this quality, but the current prototype is best poised to deliver high quality content for outdoor scenes and studio lighting.

We recorded a variety of other indoor and outdoor scenes. Some of these are shown in Figure 13. We provide these and additional VR scenes on a supplemental YouTube VR channel[7], best viewed using Google Cardboard. Temporal seams are apparent in the YouTube viewer because YouTube does not support the seamless rendering scheme described in Section 3.2.

The ODS videos show noticeable local bulging and compression in the horizontal direction. This is due to our use of software-only camera-to-motor synchronization, as discussed in Section 3. Software and hardware solutions to this problem exist, and exploration of their relative merits is left as future work.

To demonstrate that Vortex is capable of recording scenes that are very challenging for existing VR cameras, including refractive objects, caustics, reflections and specularities, fine details, and repetitive scene structures (see [Anderson et al. 2016] for more details), we include closeups of these phenomena captured with Vortex in Figure 14. In all cases Vortex, by virtue of natively capturing the ODS format, accurately captures and conveys the challenging content.

We directly compared Vortex with the output of the Google Cardboard Camera App in Figure 15. The Cardboard Camera App is run on a Nexus 6p phone, which is stabilized and mounted on a manual rotation stage. This result demonstrates the extreme vertical distortion exhibited by the Cardboard Camera App as well as all VR video camera arrays that use the same design, due to the capture circle being larger than the viewing circle as described in [Anderson et al. 2016]. Our system mitigates this distortion by allowing the capture circle to be much closer to the viewing circle, and, with smaller components, could be the exact size of the viewing circle.

We also implemented a version of the stitching pipeline used for current-generation camera rigs and show how the optical flow fails for objects close to the cameras in the supplemental video. Overall, the Vortex architecture directly captures the ODS format and not only mitigates the extreme requirements on data acquisition and processing but also reduces common artifacts of existing VR cameras.

[7] https://goo.gl/hzhU9e

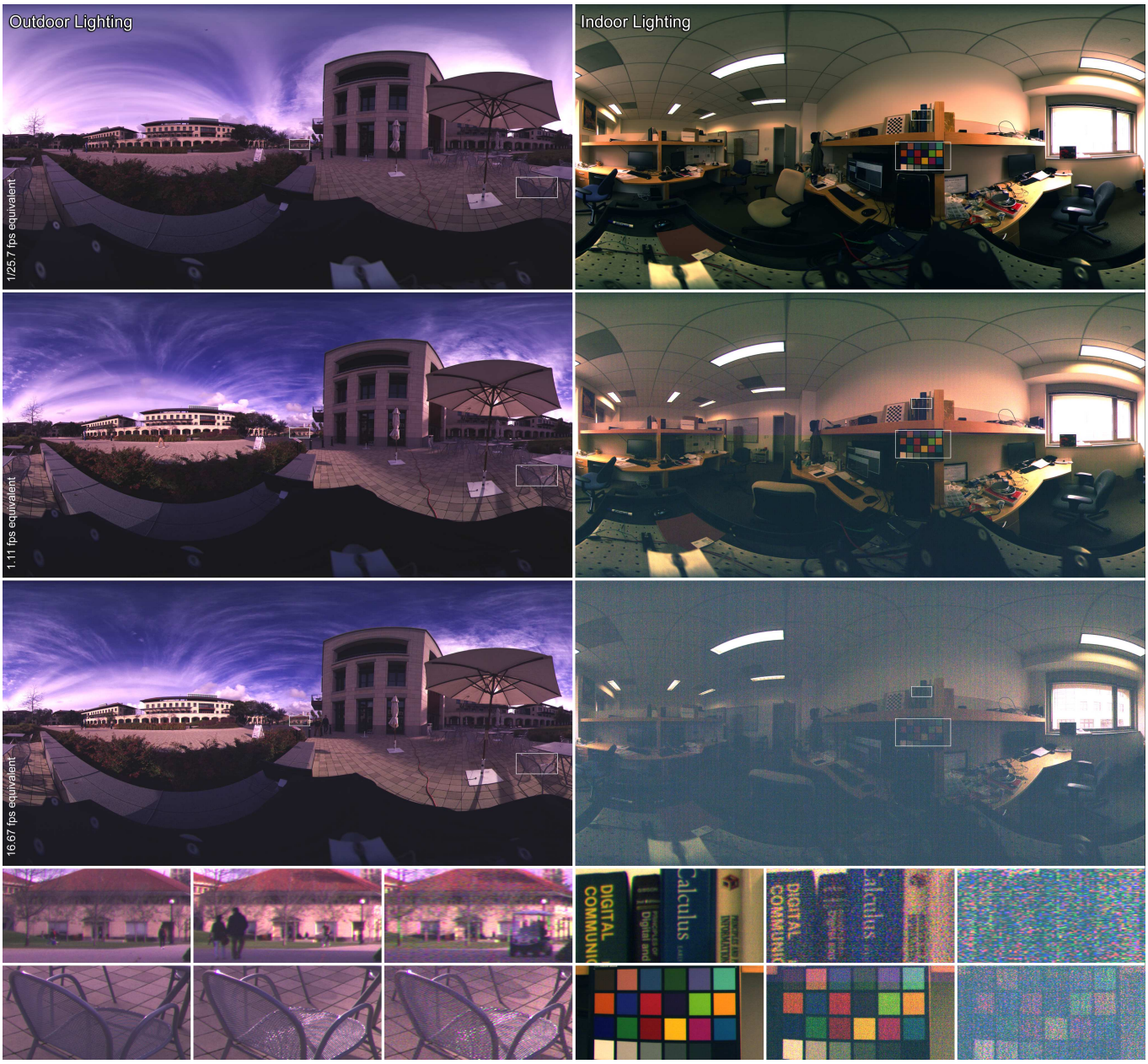


Fig. 12. Comparing three operating modes: "quality" (top row), "balanced" (second row), and "video" (third row). Outdoor lighting conditions (left) allow for high image quality when recording with exposure times that are equivalent to 16.67 fps. Dim indoor scenes can only be captured at a high quality with sufficiently long exposures, which places a limit on the frame rates.

Finally, we implemented the perceptually-driven nonuniform sampling described in Section 3.4. Figure 16 demonstrates horizontal spatial saliency, sampling the ROI 8 times more finely than outside the ROI, resulting in an increase in perceptual quality for a fixed camera and system bandwidth. We also simulated optical nonuniform sampling as seen in Figure 11, increasing effective resolution near the horizontal viewing plane, and again showing increased perceptual fidelity for identical camera and system bandwidth.

## 5 DISCUSSION

In addition to the design considerations discussed in Section 3, we discuss several other issues in the following that are relevant for future implementations of the proposed system.

*System Miniaturization.* To maximize light collection, we selected line sensors with a 7.04 µm pixel size, which is comparable to that of full-frame sensors. Modern, back-illuminated sensors, such as Sony's

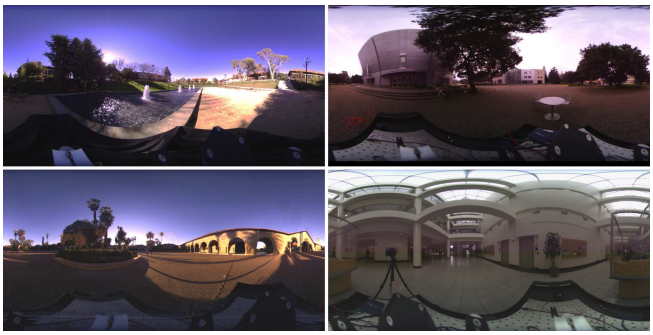

Fig. 13. Examples of indoor and outdoor scenes captured using Vortex. We provide video clips of these and other scenes on the supplementary YouTube VR channel, best viewed with Google Cardboard.

Exmor R technology, offer substantially better performance in low light conditions than the sensors used in our system. Switching to pixel sizes comparable to 3.45 µm is advantageous, because it would allow for machine vision-type cameras to be used, which not only offer smaller device form factors but also use significantly smaller lenses than full-frame sensors. The total weight and size of the device could be significantly reduced with such cameras. Line sensors with this technology are currently not available and to successfully implement fast line readout with modern 2D sensors, fast region-of-interest (ROI) readout would have to be supported by the sensor logic and the driver. At the time of submission, no such sensor was available to the authors.

Eventually, it would be ideal to use cellphone camera modules with a 1.1–1.4 µm pixel pitch. The small device form factor offered by these modules would be ideal, but light collection may be insufficient. To overcome this limitation, more than two of these tiny camera modules could be used simultaneously, which would relax the requirements on rotation speed of the system and allow for longer exposure times. Synchronized readout of many cellphone camera modules would be necessary for such a setup, which could

Fig. 14. Closeups of objects that pose a challenge to the optical flow algorithms used by existing VR cameras: reflections, refraction, caustics, fine details, and repetitive structures.


Fig. 15. Comparison between Google Cardboard Camera App and Vortex. The Cardboard approach exhibits strong vertical distortion for nearby objects and suffers from failure cases common to optical flow algorithms. Cardboard also requires the camera to be rotated on a much larger radius than Vortex, resulting in blurring of nearby content, e.g. as seen in the white teapot.

be engineered with the appropriate resources. Slip rings and all other system components are readily available at small sizes.

*Avoiding Rotating Electronics.* One of the bottlenecks of the current system is the slip ring, which requires maintenance and limits the types of camera interfaces that can be used at the moment. Removing the need for a slip ring by spinning only passive mechanical parts, such as mirrors or lenses, would thus be ideal. Dove prisms, custom mirrors, or other passive optical elements could help remove the need to actuate the detector in future implementations of this system. However, optical image quality and fabrication tolerances will have to be considered for practical versions of this idea.

*Advanced Denoising.* The system's neccesity for fast exposure times yields relatively poor low-light performance. We envision future implementations of Vortex to utilize additional wide-angle monoscopic cameras to cover the full 360° panorama, including the extreme latitudes (i.e. top and bottom). As discussed in Section 3.4, most people have a strong "equator bias", meaning they rarely look up or down. This fact is also exploited by Google's Jump system, which does not record data in these image parts, and Facebook's surround 360, which only captures it with a monoscopic wide-angle camera. The ideal setup for Vortex would thus also use a non-rotating fisheye camera to cover the extreme latitudes of the panorama. This image would have a high SNR and record many of the same image features as the spinning sensors. Therefore, a version of self-similarity denoising, such as non-local means [Buades and Morel 2005] or BM3D [Dabov et al. 2007], could be ideal for our setup, where small image patches in the noisy line sensor panoramas are denoised by similar patches in the clean, monoscopic images. Custom implementation to exploit spatial and temporal redundancy in the ODS structure would be required to allow real-time operation.

*Spatial Sound.* Commercial microphone systems capturing ambisonic audio are now widely available. Usually, these devices integrate several microphones and capture an omnidirectional sound component as well as three directional components. This is basically a first-order spherical harmonic representation of the incident sound field. YouTube VR and other VR players directly support the rendering of four-channel first-order ambisonic audio. Such a microphone could be easily integrated into our system, but we leave this effort for future work.

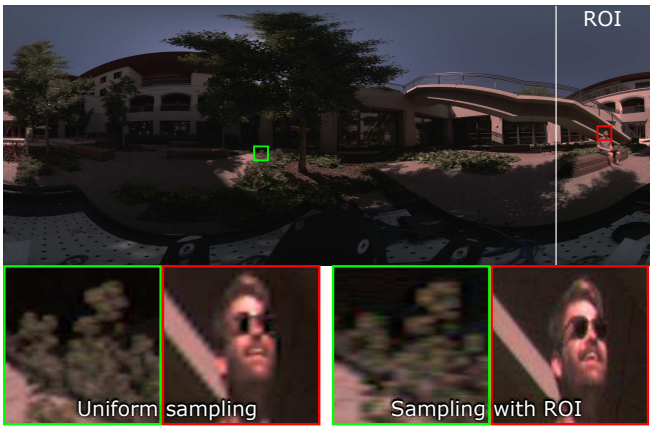


Fig. 16. Spatial nonuniform sampling: the ROI is sampled at an increased rate, and the rest of the scene at a decreased rate, yielding higher perceptual quality for the same total bandwidth. Here the ROI is sampled 8 times more densely than the rest of the scene.

## 6 CONCLUSION

Cinematic virtual reality is one of the most promising applications of emerging VR systems, and live-streaming 360° video is gaining attention as a distinct and important medium. However, the massive amount of data captured by existing VR cameras and associated processing requirements make live streaming of stereoscopic VR impossible.

In this paper we demonstrated an architecture capable of live-streaming stereoscopic virtual reality. We showed that direct ODS video capture is feasible, enabling live streaming of VR content with minimal computational burden. We demonstrated a prototype capturing ODS panoramas over a 360° horizontal by 175° vertical FOV, having up to 8192×4096 pixels, at 5 fps. We further established the viability of operation at up to 16 fps with an upgraded data offlink, and 32 fps with additional line sensors. With applications in sports, theatre, music, telemedicine and telecommunication in general, the proposed architecture opens a wide range of possibilities and future avenues of research.

## A DERIVATION OF UNWARPING

Here we derive expressions for correcting warping near the poles of native ODS cameras, as depicted in Figure 6. We begin by assuming an image covering the full viewing sphere, corresponding to horizontal and vertical ray directions $-\pi \le \theta \le \pi$ and $-\pi/2 \le \phi \le \pi/2$, respectively. For a scene at distance $r$, a spherical-to-Cartesian conversion yields a coordinate $(x, y, z)$ for each ray $(\theta, \phi)$. To these we apply an offset based on the radius of rotation of the camera $R$

$$x' = r\cos\phi\cos\theta - R\sin\theta, \tag{3}$$

$$y' = r\cos\phi\sin\theta + R\cos\theta, \tag{4}$$

$$z = r\sin\phi. \tag{5}$$

Converting back to ray directions $(\theta', \phi')$ and finding the shifts $\Delta\theta = \theta' - \theta$, $\Delta\phi = \phi' - \phi$, yields

$$\Delta\phi = \tan^{-1}\left(\frac{\sin\phi}{\sqrt{(R/r)^2 + \cos^2\phi}}\right) - \phi, \tag{6}$$

$$\Delta\theta = \tan^{-1}\left(\frac{R/r}{\cos\phi}\right). \tag{7}$$

Note that both shifts are symmetric about the axis of rotation of the camera, depending only on the vertical dimension $\phi$. The change in ray direction depends on the ratio of the camera rotation radius to the scene distance $R/r$.

## ACKNOWLEDGMENTS

This work was generously supported by the NSF/Intel Partnership on Visual and Experiential Computing (Intel #1539120, NSF #IIS-1539120). R.K. was supported by an NVIDIA Graduate Fellowship. G.W. was supported by a Terman Faculty Fellowship and an NSF CAREER Award (IIS 1553333). We would like to thank Ian McDowall, Surya Singh, Brian Cabral, and Steve Mann for their insights and advice.

## REFERENCES

Michael Adam, Christoph Jung, Stefan Roth, and Guido Brunnett. 2009. Real-time Stereo-Image Stitching using GPU-based Belief Propagation. In *Vision, Modeling, and Visualization Workshop (VMV)*. 215–224.

Rajat Aggarwal, Amrisha Vohra, and Anoop M. Namboodiri. 2016. Panoramic Stereo Videos With a Single Camera. In *Proc. IEEE CVPR*.

Robert Anderson, David Gallup, Jonathan T. Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M. Seitz. 2016. Jump: Virtual Reality Video. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6 (2016), 198:1–198:13.

Robert G. Batchko. 1994. Three-hundred-sixty degree electroholographic stereogram and volumetric display system. In *Proc. SPIE*, Vol. 2176. 30–41.

R. Benosman, T. Maniere, and J. Devars. 1996. Multidirectional stereovision sensor, calibration and scenes reconstruction. In *Proc. ICPR*, Vol. 1. 161–165.

P. Bourke. 2010. Capturing omni-directional stereoscopic spherical projections with a single camera. In *Proc. IEEE VSMM*. 179–183.

Matthew Brown and David G. Lowe. 2007. Automatic Panoramic Image Stitching Using Invariant Features. *IJCV* 74, 1 (2007), 59–73.

Antoni Buades and Jean-Michel Morel. 2005. A non-local algorithm for image denoising. In *Proc. IEEE CVPR*.

V. Chapdelaine-Couture and S. Roy. 2013. The omnipolar camera: A new approach to stereo immersive capture. In *Proc. ICCP*. 1–9.

Oliver Cossairt, Mohit Gupta, and Shree K. Nayar. 2013. When Does Computational Imaging Improve Performance? *IEEE Trans. Im. Proc.* 22, 2 (2013), 447–458.

Oliver S. Cossairt, Joshua Napoli, Samuel L. Hill, Rick K. Dorval, and Gregg E. Favalora. 2007. Occlusion-capable multiview volumetric three-dimensional display. *OSA Appl. Opt.* 46, 8 (2007), 1244–1250.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Im. Proc.* 16, 8 (2007), 2080–2095.

Donald G. Dansereau, Glenn Schuster, Joseph Ford, and Gordon Wetzstein. 2017. A Wide-Field-of-View Monocentric Light Field Camera. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Ho-Chao Huang and Yi-Ping Hung. 1998. Panoramic Stereo Imaging System with Automatic Disparity Warping and Seaming. *Graph. Models Image Process.* 60, 3 (May 1998), 196–208. https://doi.org/10.1006/gmip.1998.0467

H. Ishiguro, M. Yamamoto, and S. Tsuji. 1990. Omni-directional stereo for making global map. In *Proc. ICCV*. 540–547.

Andrew Jones, Ian McDowall, Hideshi Yamada, Mark Bolas, and Paul Debevec. 2007. Rendering for an Interactive 360&Deg; Light Field Display. *ACM Trans. Graph. (SIGGRAPH)* 26, 3 (2007).

Kevin Matzen, Michael F. Cohen, Bryce Evans, Johannes Kopf, and Richard Szeliski. 2017. Low-cost 360 Stereo Photography and Video Capture. *ACM Trans. Graph.* 36, 4 (July 2017), 148:1–148:12.

David W. Murray. 1995. Recovering Range Using Virtual Multicamera Stereo. *Proc. CVIU* 61, 2 (1995), 285 – 291.

S. Peleg, M. Ben-Ezra, and Y. Pritch. 2001. Omnistereo: panoramic stereo imaging. *IEEE Trans. PAMI* 23, 3 (2001), 279–290.

C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung. 2013. Megastereo: Constructing High-Resolution Stereo Panoramas. In *Proc. IEEE CVPR*. 1256–1263.

Heung-Yeung Shum and Li-Wei He. 1999. Rendering with Concentric Mosaics. In *Proc. SIGGRAPH*. 299–306.

Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, and Gordon Wetzstein. 2016. Saliency in VR: How do people explore virtual environments?. In *arXiv:1612.04335*.

Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications*. Springer.

Kenji Tanaka, Junya Hayashi, Masahiko Inami, and Susumu Tachi. 2004. TWISTER: An immersive autostereoscopic display. In *Proc. IEEE VR*. 59–66.

K. Tanaka and S. Tachi. 2005. TORNADO: omnistereo video imaging with rotating optics. *IEEE TVCG* 11, 6 (2005), 614–625.

Christian Weissig, Oliver Schreer, Peter Eisert, and Peter Kauff. 2012. The Ultimate Immersive Experience: Panoramic 3D Video Acquisition. In *Proc. MMM*. 671–681.