

POSTED ON APR 12, 2016 TO [VIDEO ENGINEERING](#)

Introducing Facebook Surround 360: An open, high-quality 3D-360 video capture system



Today we announced [Facebook Surround 360](#) — a high-quality, production-ready 3D-360 hardware and software video capture system.

- Facebook has designed and built a durable, high-quality 3D-360 video capture system.
- The system includes a design for camera hardware and the accompanying stitching code, and we will make both available on GitHub this summer. We're open-sourcing the camera and the software to accelerate the growth of the 3D-360 ecosystem — developers can leverage the designs and code, and content creators can use the camera in their productions.
- Building on top of an optical flow algorithm is a mathematically rigorous approach that produces superior results. Our code uses optical flow to compute left-right eye stereo disparity. We leverage this ability to generate seamless stereoscopic 360 panoramas, with little to no hand intervention.
- The stitching code drastically reduces post-production time. What is usually done by hand can now be done by algorithm, taking the stitching time from weeks to overnight.
- The system exports 4K, 6K, and 8K video for each eye. The 8K videos double industry standard output and can be played on Gear VR with Facebook's custom Dynamic Streaming technology.



1 OF 5



In designing this camera, we wanted to create a professional-grade end-to-end system that would capture, edit, and render high-quality 3D-360 video. In doing so, we hoped to meaningfully contribute to the 3D-360 camera landscape by creating a system that would enable more VR content producers and artists to start producing 3D-360 video.

Defining the challenges of VR capture

When we started this project, all the existing 3D-360 video cameras we saw were either proprietary (so the community could not access those designs), available only by special request, or fundamentally unreliable as an end-to-end system in a production environment. In most cases, the cameras in these systems would overheat, the rigs weren't sturdy enough to mount to production gear, and the stitching would take a prohibitively long time because it had to be done by hand. So we set out to design and build a 3D-360 video camera that did what you'd expect an everyday camera to do — capture, edit, and render reliably every time. That sounds obvious and almost silly, but it turned out to be a technically daunting challenge for 3D-360 video.

Many of the technical challenges for 3D video stem from shooting the footage in stereoscopic 360. Monoscopic 360, using two or more cameras to capture the whole 360 scene, is pretty mainstream. The resultant images allow you to look around the whole scene but are rather flat, much like a still photo.

However, things get much more complicated when you want to capture 3D-360 video. Unlike monoscopic video, 3D video requires depth. We get depth by capturing each location in a scene with two cameras — the camera equivalent of your left eye and right eye. That means you have to shoot in stereoscopic 360, with 10 to 20 cameras collectively pointing in every direction. Furthermore, all the cameras must capture 30 or 60 frames per second, exactly and simultaneously. In other words, they must be globally synchronized. Finally, you need to fuse or stitch all the images from each camera into one seamless video, and you have to do it twice: once from the virtual position for the left eye, and once for the right eye.

This last step is perhaps the hardest to achieve, and it requires fairly sophisticated computational photography and computer vision techniques. The good news is that both of these have been active areas of research [1, 2, 3] for more than 20 years. The combination of past algorithm research, the rapid improvement and availability of image sensors, and the decreasing cost of memory components like SSDs makes this project possible today. It would have been nearly impossible as recently as five years ago.

The VR capture system

With these challenges in mind, we began experimenting with various prototypes and settled on the three major components we felt were needed to make a reliable, high-quality, end-to-end capture system:

- The hardware (the camera and control computer)
- The camera control software (for synchronized capture)
- The stitching and rendering software

All three are interconnected and require careful design and control to achieve our goals of reliability and quality. Weakness in one area would compromise quality or reliability in another area.

Additionally, we wanted the hardware to be off-the-shelf. We wanted others to be able to replicate or modify our design based on our design specs and software without having to rely on us to build it for them. We wanted to empower technical and creative teams outside of Facebook by allowing them full access to develop on top of this technology.

The camera hardware

As with any system, we started by laying out the basic hardware requirements. Relaxing any one of these would compromise quality or reliability, and sometimes both.

Camera requirements:

- The cameras must be globally synchronized. All the frames must capture the scene at the same time within less than 1 ms of one another. If the frames are not synchronized, it can become quite hard to stitch them together into a single coherent image.
- Each camera must have a global shutter. All the pixels must see the scene at the same time. That's something, for example, cell phone cameras don't do; they have a rolling shutter. Without a global shutter, fast-moving objects will diagonally smear across the camera, from top to bottom.
- The cameras themselves can't overheat, and they need to be able to run reliably over many hours of on-and-off shooting.
- The rig and cameras must be rigid and rugged. Processing later becomes much easier and higher quality if the cameras stay in one position.
- The rig should be relatively simple to construct from off-the-shelf parts so that others can replicate, repair, and replace parts.

We addressed each of these requirements in our design. Industrial-strength cameras by Point Grey have global shutters and do not overheat when they run for a long time. The cameras are bolted onto an aluminum chassis, which ensures that the rig and cameras won't bounce around. The outer shell is made with powder-coated steel to protect the internal components from damage. (Lest anyone think an aluminum chassis or steel frame is hard to come by, any machining shop will do the honors once handed the specs.)

The camera control and capture software and storage

Once the hardware design is complete, camera control and data movement become the next big issues to tackle. Because we wanted to keep the system modifiable, we chose to control the cameras using a Linux-based PC that contained enough system bandwidth to support the transfer of live video streams from all the cameras to disk.

Furthermore, the isochronous nature of the cameras meant that we needed a real-time thread for capturing the frames followed by other lower-priority buffered disk writing threads to ensure that all frames were captured and not dropped. At 30 Hz, this required on the order of a 17 Gb/s sustained transfer rate. Similarly, we used an 8-way level-5 RAID SSD disk system to keep up with the isochronous camera capture rates. This, in coordination with the sturdy cameras, allows for minutes to hours of continuous capture.

Complicating matters, the camera control must be performed remotely since the camera itself is capturing the full 360 scene. Our solution was to control the camera with a simple web interface, making it possible to control the camera from any device that supports an HTML browser.

In our system, exposure, shutter speed, analogue sensor gain, and frame rate are controlled in the custom-designed capture software on a per-camera basis. The cameras are globally synced from the same software. We capture “raw” Bayer data to ensure quality throughout the image rendering pipeline.

The stitching software

The computational imaging algorithms that stitch all the images together are arguably at the heart of the system and represent the most difficult part of the system. Luckily, the past 20 years of computational photography and computer vision research and an additional 60 years of aerial stereo photogrammetry (used to create topological maps) gave us a strong starting point.

Since the stitching process generates a final video image, visual image quality was paramount in our image processing and computational imaging pipeline. A subtle aspect of digital image capture is that care must be taken at each step of computation to ensure pixel quality is maintained. It’s quite easy to mistakenly degrade the image quality at any step in the process, after which that resolution is forever lost. Repeat that several times, and quality is completely compromised.

We process the data in several steps:

- Convert raw Bayer input images to gamma-corrected RGB.
 - Mutual camera color correction
 - Anti-vignetting
 - Gamma and tone curve
 - Sharpening (deconvolution)
 - Pixel demosaicing
- Perform intrinsic image correction to remove the lens distortion and reproject the image into a polar coordinate system.
- Bundle adjusted mutual extrinsic camera correction to compensate for slight misalignments in camera orientation.
- Perform optical flow between pairs of cameras to compute left-right eye stereo disparity.
- Synthesize novel views of virtual cameras for each view direction separately for the left and right eye view based on the optical flow.
- Composite final pixels of left and right flows.

The key algorithmic element is the concept of optical flow. Our code builds upon an optical flow algorithm that is mathematically trickier than other stitching solutions but delivers better results. Optical flow allows us to compute left-right eye stereo disparity between the cameras and to synthesize the novel views separately for the left and right eyes. We flow the top and bottom cameras into the side cameras, and similarly we flow pairs of cameras to match objects that appear at different distances. In fact, optical flow remains an open research area since it’s an ill-posed inverse problem. Its ill-posedness stems from ambiguities caused by occlusions — one camera not being able to see what an adjacent camera can see. While this is mitigated by having multiple cameras and time-varying capture, it nevertheless remains a difficulty. Our code uses optical flow to compute left-right eye stereo disparity. We leverage this ability to generate seamless stereoscopic 360 panoramas automatically. Dailies are delivered overnight. This is a dramatic reduction in post-production time.



Playback

We output at 4K, 6K, and 8K per eye. Due to high bandwidth and data demands, 6K and 8K output use our [Dynamic Streaming](#) codec for Gear VR, doubling the industry standard. The output file from the system can be viewed in VR headsets such as the Oculus Rift and Gear VR. These can also be outputted and shared in places like the Facebook News Feed — in which case only one of the monoscopic views is displayed when you're scrolling through News Feed, but the full stereo is downloadable.

Build and create

We believe that we were successful in creating a reliable, production-ready camera that is a state-of-the-art capture and stitching solution. But we know that there is more work to be done.

Technology challenges remain. Different combinations of optical field-of-view, sensor resolution, camera arrangement, and number of cameras used present a fairly complicated engineering design challenge. More cameras help ease the job of subsequent stitching, but also increase the volume of captured data and bandwidth needed to process it. Similarly, increasing sensor resolution improves final rendering quality but, again, at the cost of increased bandwidth and data volume.

On the optical side, increasing each camera's field of view helps either reduce the number of cameras or ease the amount of required stitching at the cost of spatial resolution. Wide-angle optics are also more sensitive to sensor back-focal distance and angular misalignment.

There is no one right answer here, and we know there are ideas we haven't explored. In fact, this is one of the reasons we're making our design freely available. We want others to join us in refining this technology. We know from experience that a broader community can move things forward faster than we can on our own. All the software and hardware blueprints will be made available by the end of this summer. Make it faster, cheaper, smaller. Make it better. We'll be working to do the same here. We can't wait to see what you develop and, ultimately, what you create with it.

A final note — making a personal connection

When our small engineering team started this project, we were excited about the prospects of the 3D-360 technology and, honestly, just excited about the plain coolness and geekiness of working on something as leading-edge as a “VR camera.”

However, the potential of the technology didn't hit home for me until a recent trip to see my parents. I wanted to show them what I was up to at work — they always ask, and unlike when I'm working strictly on software, with this project I had tangible things for them to explore, like a sample video we had captured in our new Frank Gehry office building. My dad is at a point in life where traveling to my office isn't going happen, so this was a wonderful way to show him my work world. As he was looking around, he started asking about the people, the place, and

space. “Are these the people you work with? How many people work in the place? Wow, this looks big — how many square feet is it?” And so it went for several minutes: his questions, my answers. At that point the technology disappeared, and I was simply connecting with my dad in a way I didn’t expect and in a way that was personally important to me.

It was then that I deeply understood how this technology could connect people and bring them closer together — how it is fundamentally tied to Facebook’s mission, and why working on this matters and goes far beyond just being cool. We hope it has the same impact for you.

TAGS: [HARDWARE](#)

Like

Share

5 people like this. Be the first of your friends.




◀ Prev

[Breaking ground at Clonee](#)

Next ▶

[Open-sourcing ReDex: Making Android apps smaller and faster](#)



Read More in Video Engineering

[View All](#) ▶



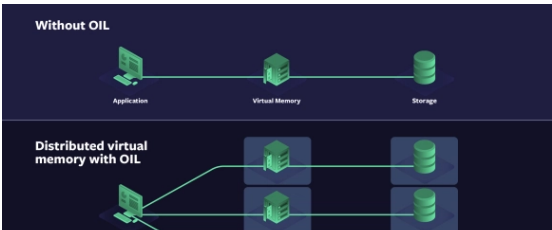
NOV 17, 2019

[Evaluating COPA congestion control for improved video performance](#)



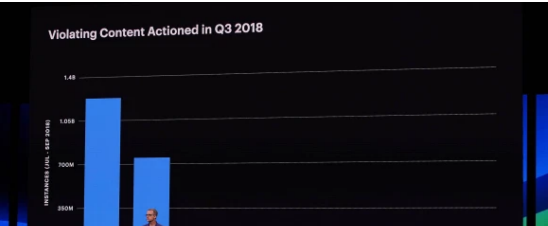
OCT 17, 2019

[Video @Scale 2019 recap](#)



JUL 9, 2019

[OIL+VCache: File abstraction for distributed systems](#)



MAY 1, 2019

[F8 2019 Day 2 keynote and session videos](#)



OCT 12, 2018

[Oculus Connect 5: Tech talks roundup](#)



MAY 2, 2018

[F8 2018 Day 2 Keynote Speakers](#)

Related Posts

Related Positions

[Software Engineer - Video Specialist](#)

[SEATTLE, US](#)

[Software Engineer, Video Presence Media Team](#)

[MENLO PARK, US](#)

[Software Engineer, Video Ingestion](#)

[MENLO PARK, US](#)