Griffith UNIVERSITY
Learning Futures

Streaming 360 Video Live at #aNewReality

Provided by Espen Dammen / Learning Futures / Griffith University

The following equipment was used during the 360 Live Stream during the Immersive Symposium:

- Insta360 Pro Camera
- Sony x70 Handicam
- 2x Sennheiser G3 microphones (one handheld, one lavalier)
- Apple iMac
- Blackmagic Mini Recorder

The following software was used:

- Open Broadcast Software (OBS) – *Open Source*
- Local RTMP Server – *Open Source*
- YouTube via internet browser

How it all fits together:

Streaming in higher resolution will give a better result, but can complicate things. Our camera is capable of Live Streaming up to 3840x2160 (4K), and outputs via HDMI. The problem with this is that you need a computer and capturing hardware capable of receiving 4K video via HDMI. Our solution was to use RTMP via Ethernet. This allows us to send the video signal via a network cable to a local host. While our camera allows this natively, we found that using the Built-In RTMP server in the camera created massive delays and unstable video image. The solution was setting up a local RTMP server on an iMac, and have the camera send its video to this local server. The same iMac can then decode this network signal and send the video to any streaming software of our choice. By doing this we are relieving the camera from any strenuous work, and we do all the heavy lifting on the iMac. Our streaming software (OBS) received the signal from the Local RTMP server.

To incorporate a higher quality of audio than what the camera can do on its own, we decided to use wireless microphones for the presenters. To get this audio into our stream, we needed a way of capturing the audio and putting it on top of the video coming from our 360 camera. Our solution was to set up a standard video camera which received the audio signals, and which then output an A/V signal to our iMac via HDMI using a Blackmagic Mini Recorder to capture it.

In our streaming software we were then able to combine the audio and the video, adjust audio levels, fade in/out etc. With some testing beforehand, we realised that the 20m network cable running from our camera to our iMac created some delay in the video, while the audio had no delay. With some trial and error we found out that the exact delay difference was 1800ms (1.8 seconds). This can be adjusted in our streaming software so that video and audio arrives at the viewer in sync.

We decided to stream to YouTube as the platform is easy to set up, and allows for easy 360 video navigation. YouTube also allows for Live Chat for viewers, and we had someone in the audience curating the chat and allowing questions to asked live.

Handy notes:

- Always test all equipment before the event, and make sure you are testing the actual equipment you will be using. We found out that our intended streaming machine (MacBook Pro 2014) was not powerful enough to stream in 4K
- YouTube Live Events have a lot of settings, and many of them can not be changed after the event is created. E.g. if you forget to tick “This is a 360 Stream” your stream will not be in 360, and you will have to create a new event, which has a new URL for the viewers etc.
- When streaming to YouTube, there will be a delay between the actual event and what the viewers see. This can be anywhere between 10 seconds to 30 seconds, depending on your latency settings in YouTube. The caveat; lower latency will restrict your resolution. If you intend to stream in 4K, you will have to go with the

highest latency setting, and there will be about 30 seconds delay. Keep this in mind if you are taking questions from the Live Chat.

- The YouTube Live Event can be embedded into other websites, but to access the Live Chat you will need to be watching via YouTube.
- The procedure explained in this document work with the equipment listed, if you are user other equipment or software (e.g. different cameras on a Windows PC) you may have different results.