

You can add live virtual experiences to your application to allow people to remotely inspect and analyze physical sites. I'll explain how to use open source software and a Jetson Nano running Linux to stream 4K video to another computer with RTSP over Wi-Fi. This technique can also be adapted to add a WebRTC server for multiple people to view the stream on web pages. You can go through many of the examples with a standard webcam or Raspberry Pi v2 camera and a Linux machine or a Raspberry Pi.

The examples provided are from an active online developer community discussion. Thank you to the many people who contributed solutions to the problems we overcame.

Advantages of 360° Video Streaming

A single 360° camera provides a seamless viewing experience in all directions. You don't miss anything and you don't wait for servos to move your standard camera. You can either turn your head and immediately change your view or use a mouse in a web page to rotate the live scene. When you transmit a scene to your audience, they get everything and they get it immediately.

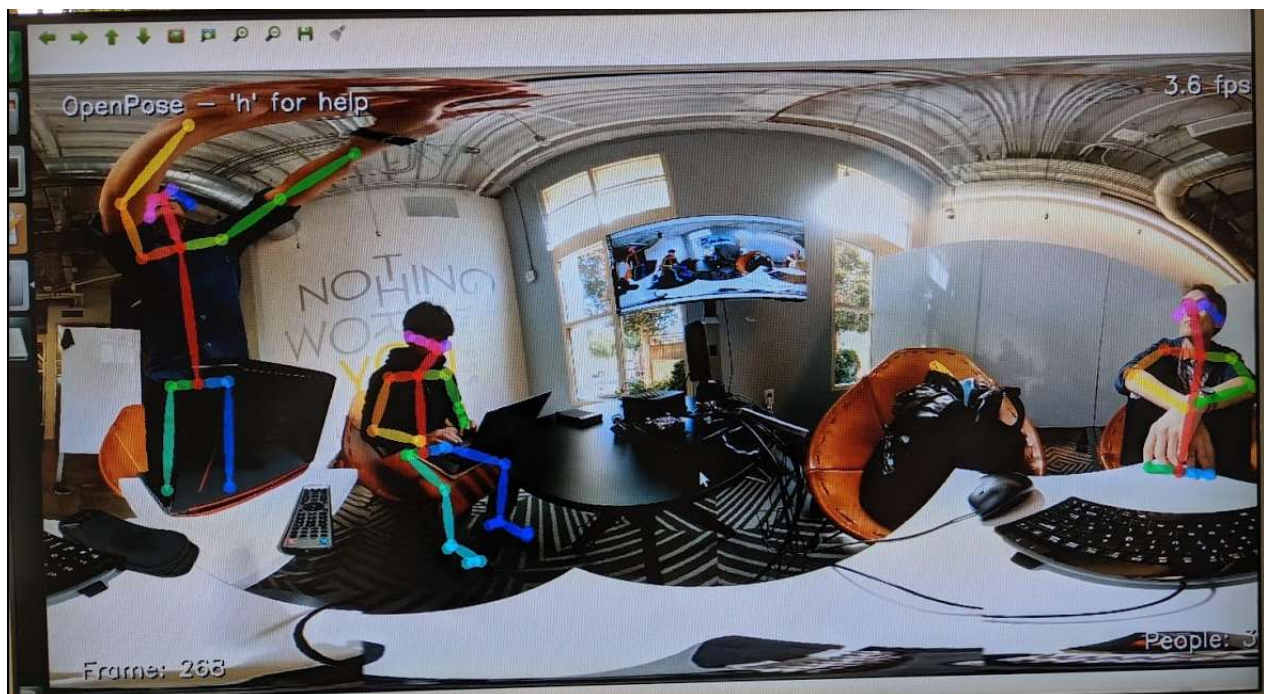
Increasing Dev Performance: KPIs, AI, and Bottlenecks

In this session, we'll provide targeted solutions to these problems. Learn how to identify and address the “weakest link” effect, establish meaningful KPIs for accurate performance tracking, and integrate AI tools without sacrificing human creativity and natural flow. Walk away with actionable strategies designed to elevate both your team's and your own individual performance.



WATCH NOW

In many cases, the scene is being analyzed by an AI. Your smart application can analyze the entire scene in each frame. It will see everything and analyze every object on every frame.



Getting Started With Consumer 360° Cameras

Instead of expensive multi-camera rigs with external image stitching software, we used consumer 360° cameras in our low-cost projects. A 360° camera provides a seamless view of up, down, left right, front, and back at 30fps with 400ms latency using a cheap Linux computer as the relay.

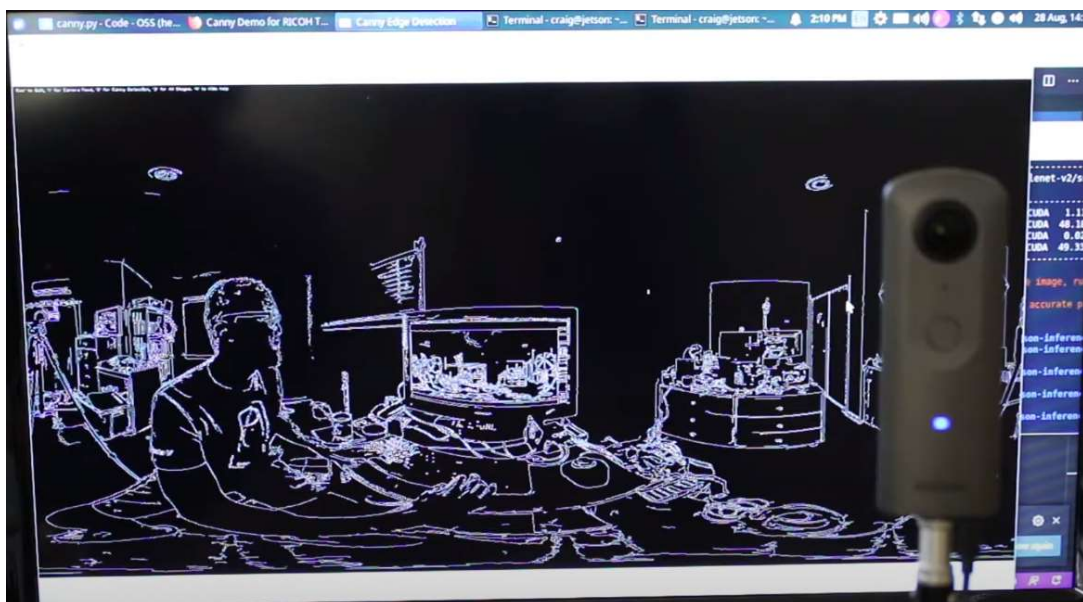
We used open source libuvc to connect the camera to the Linux computer and used gstreamer to manage the video stream. To enhance the stream, we used OpenCV and DetectNet to automatically analyze spaces for further analysis by people. In most of my examples, a 360° camera is connected to an NVIDIA Jetson Nano with a USB cable. When I refer to the "Linux computer," it is usually the Jetson Nano, which is \$100 without a power support and microSD card.

In the image below, the left view is what the human operator of the robot sees. The scene can instantly be moved in any direction in real-time. The 360° camera stream can be used to pilot the robot below if the robot moves slowly. Even at less than 500ms latency, the technology of consumer 360° cameras is still not on the same level as industrial cameras. The current latency may be too high for many commercial projects. The robot in this example is for educational purposes and not intended for industrial use.

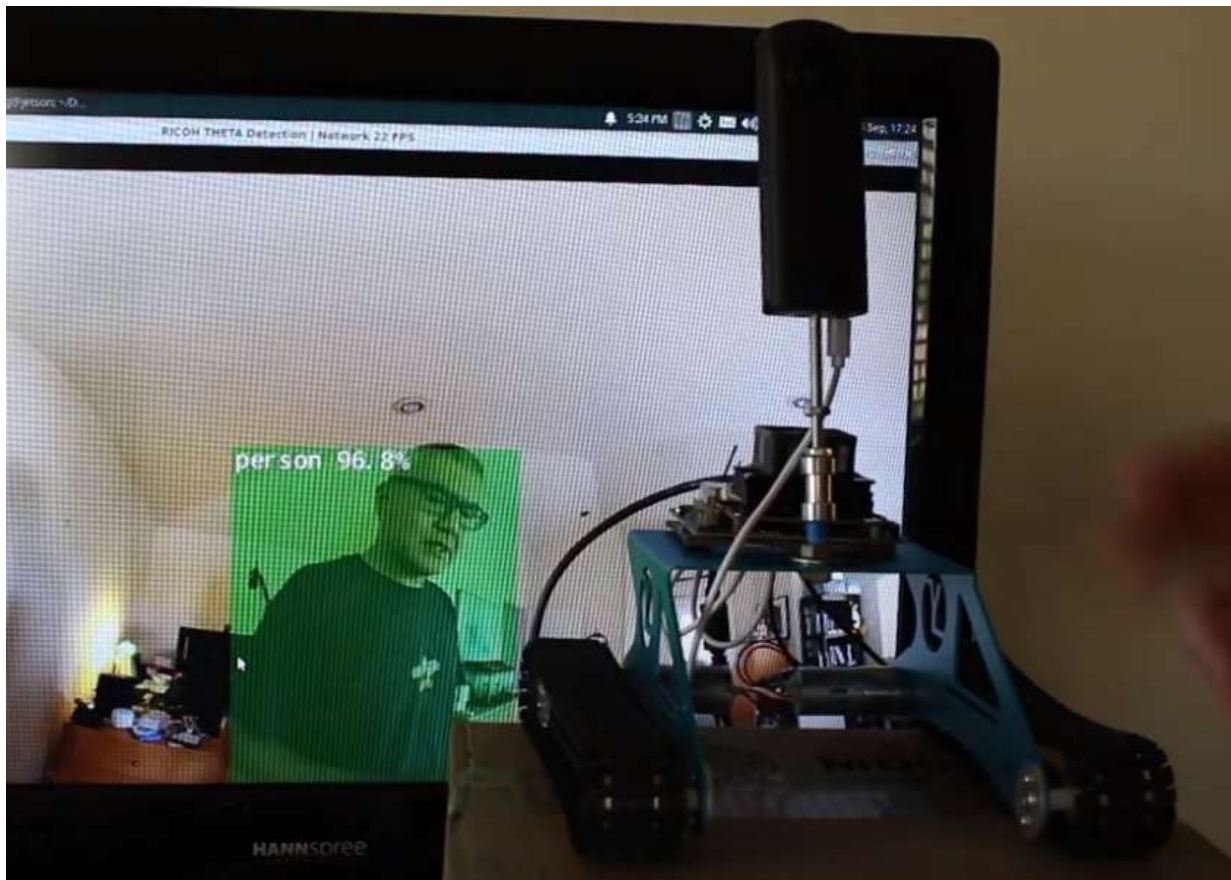


Instead of simply using the video feed to stream the experience, most people are using a Linux computer to add object detection to the video feed.

The example below uses OpenCV and Python to identify edges with Canny.



The example below uses DetectNet for object detection.



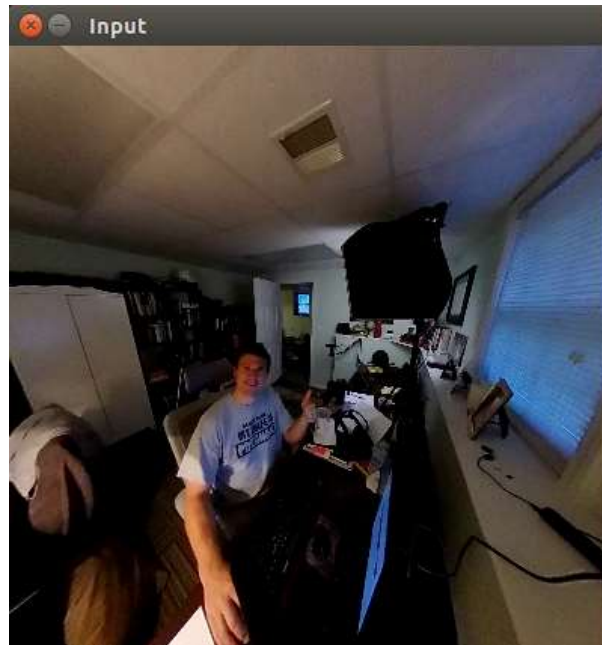
These demos were built with the community based on open source projects and shown at a recent online meetup for 360° video streaming.

The source code and additional information for you to build these demos yourself is available [here](#). If you do not have a 360° camera, you can use a standard web cam, or a Raspberry Pi v2 camera connected to the CSI port of the NVIDIA Jetson. If you do not have an NVIDIA Jetson, you can use a Raspberry Pi or a x86 Linux computer.

Although the Raspberry Pi 4 is cheaper and more popular than the NVIDIA Jetson Nano, the Nano has better hardware acceleration for 4K H.264 video. In our tests, we were not able to use the Raspberry Pi for 4K streaming. You can definitely use the Raspberry Pi with a standard USB webcam or the Raspberry Pi cameras. However, it didn't work with the RICOH THETA 360° camera we used in our projects.

Converting 360° Live Video Into Standard Perspective

Community member Jaap contributed a solution to use OpenCV to convert sections of a 360° video stream into standard perspective in real-time. Here is Jaap sheltering in his basement and hacking on 360° video streams.



The solution is only partially completed because we are still trying to use the CUDA acceleration of the NVIDIA Jetson with OpenCV to handle the `cv.cuda.remap` calls. At the moment, the solution only works on smaller sections. The camera that most of us use is a RICOH THETA V or Z1 that streams 30fps at 4K. This puts a lot of demand on the small board computers for processing.

The remapping solution is intended to enable us to use standard deep learning models with common image databases. The equirectangular perspective of the 360° video stream causes challenges for non-fisheye image models.

Helping People Interact with Equirectangular Video

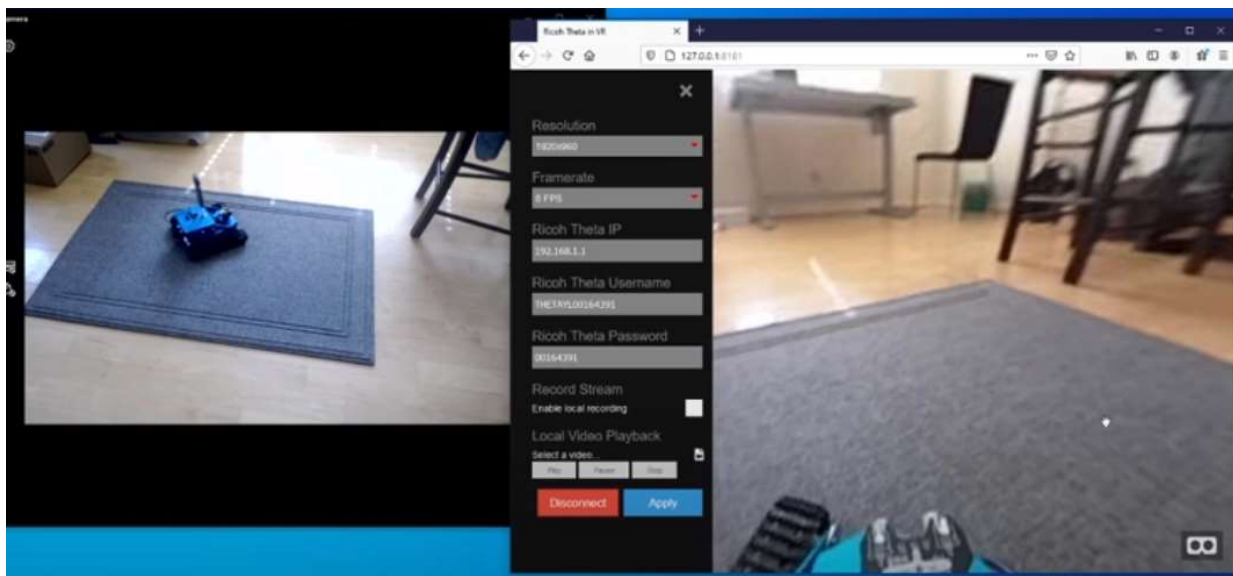
Although computers can deal with the equirectangular video frames, people prefer a way to navigate around the 360° video, similar to a Google Maps or Google Streetview experience.

A simple technique is to convert the 360° video stream into MotionJPEG and then use A-Frame to display the live stream. Members from the Lockheed Martin Amelia Drone project at RIT produced an open source cockpit for the RICOH THETA. Discussion and code is available [here](#). Direct link to GitHub repo.

This is one of the RIT team members testing the headset. The video feed is using specialized radio transmitters as their project is based on a flying drone.



I used the open source project the RIT team built with my test educational robot and displayed the view in a standard web browser.



Video Transmission Over IP Networks

Although the camera we're using runs Android OS inside of it and can stream directly to another device, the most flexible option is to connect the 360° camera to a Jetson Nano with a USB cable and then relay the feed to another device.

Most of our efforts have focused on using gstreamer. We're still having some problems using FFmpeg with the RICOH THETA.

From an x86 Linux machine, the following gstreamer pipeline should work with standard webcams as well as the RICOH THETA.

Java


```
1 | gst-launch-1.0 decodebin ! jpegenc ! rtpjpegpay ! udpsink host=192.168.2.100 port=5000 qos=false
```

The pipeline will send the stream to another computer at 192.168.2.100. You need to change the IP address to the computer you will be viewing the stream on.

The computer that I've viewing the stream on is a Jetson. I used this pipeline.

Java

```
1 | gst-launch-1.0 udpsrc port=5000 ! application/x-rtp,encoding-name=JPEG,payload=26 ! rtpjpegdepay
```

If the computer that you're viewing the stream on is an x86 machine, then you should change `nveglglessink` to `autovideosink`.



Community member `zdydek` contributed another pipeline using `gst-rtsp-server`. This contains some techniques that are specific to the RICOH THETA.

To help more people experiment with `gst-rtsp-server`, I tested my Raspberry Pi camera v2 connected to the Jetson Nano using the onboard CSI ribbon cable. The examples directory of `gst-rtsp-server` has a file, `test-launch`. Using that file, I used this pipeline to serve the Raspberry Pi camera.

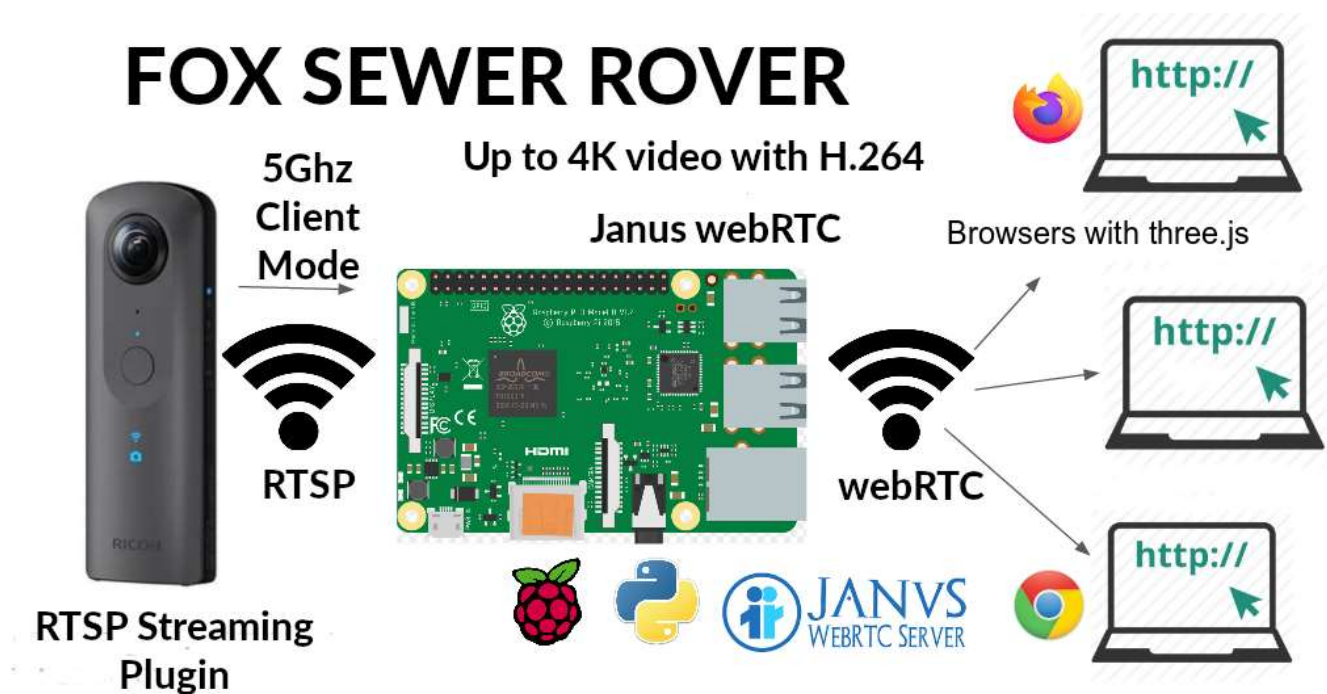
```
./test-launch "nvarguscamera src ! nvvidconv ! nvv4l2h264enc ! h264parse !  
rtph264pay name=pay0 pt=96"
```

RTSP can be viewed from a number of players, including VLC. You need to point VLC at the IP address of the computer with the webcam connected to it.

WebRTC With Janus

Community member Hugues contributed a tutorial to use a Raspberry Pi to stream WebRTC to web browsers using the open source Janus WebRTC Server.

This is the architecture he used for his inspection robot.



Summary

Our developer community is seeing a big surge in 360° live streaming projects. AI and object detection technology are moving into mainstream business warehouses and factories. Standard cameras leave gaps in a view, which can impair analysis and decision-making. Inexpensive consumer 360° cameras can stream an entire view and use the same USB connection interface as standard cameras.

Established open source video transmission projects like gstreamer are stable and work great with 360° video. As the 360° camera functions as USB webcam, you can experiment with the same concepts using your existing webcam or a cheap Raspberry Pi camera.

If you have any questions, or ideas about 360° video streaming, you can post them at the bottom of this article or in our free and open 360° camera developer community. This is an exciting time with new developments happening every week. Please join the action.