Fish-eye Camera Calibration and Video transmission System

This project was offered by the Third Research Institute of China Aerospace Science and Engineering Group. As the requirement, a real-time video calibration and transmission system with a fish-eye camera lens is needed. Some specialisations requests resolution of the video should be beyond 720p and Frame Ratio should be more than 20 frames per second.

1.1 Overview of the project

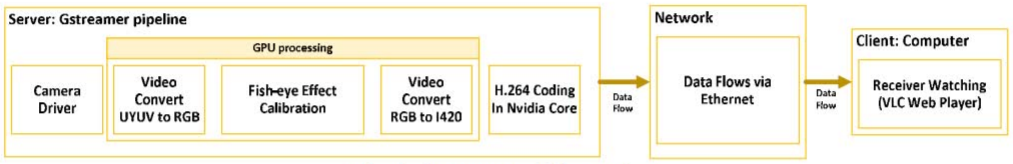


Fig 1. Structure of the project

In order to complete the task, my colleagues and I were supposed to establish a server system, which included camera, fish-eye effect calibration, video coding and Ethernet transferring interface. Users could receive data through internet on an ordinary computer with a flow data player. The structure of system is shown in Fig.1.

1.2 Scheme of solution

To establish the system, my team divided project into hardware design and software design parts. We combined Gstreamer pipeline, OpenCV video tools and CUDA accelerating tools together to build our Real-time Fish-eye Camera Video Calibration system.

1.2.1 Hardware design

(a) (b)

Fig 2. (a) Nvidia TX-1 and (b) CSI-interface camera

To build up the whole system, we choses Nvidia TX-1 as our platform since it has powerful calculation units and 192 GPU core which are very suitable for parallel calculation. Along with high performance in counting, TX-1 is also attached with H.264 video coding/decoding circuit which provide convenience during video processing.

1.2.2 Software design

As for the software design, we established a Gstreamer pipe line including camera driver, video format transforming, video reshaping (fish-eye effect calibrating) and H.264 coding circuit driver. OpenCV and CUDA played significant role in this process.

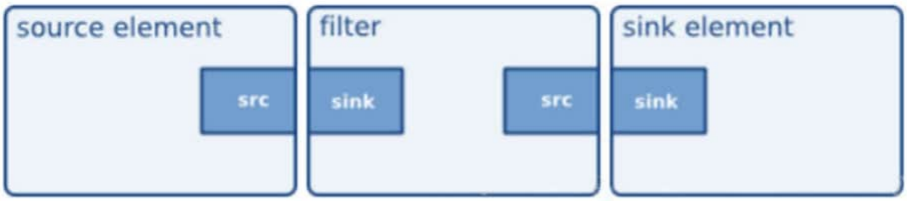
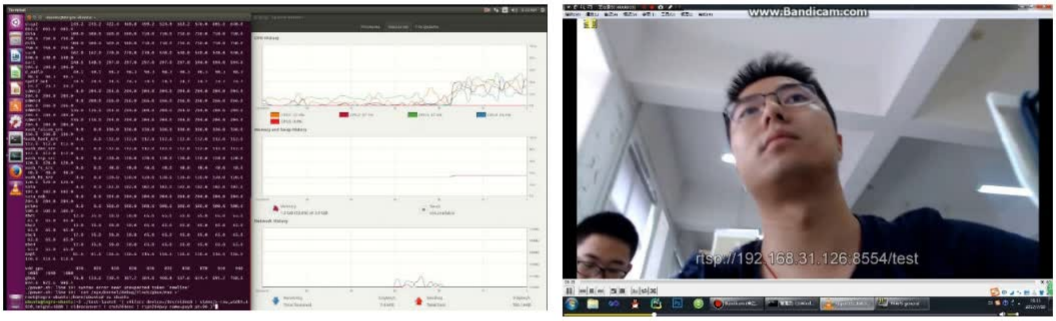


Fig 3. Plug-in system

The Gstreamer framework is plug-in based shown in Fig 3.

1.3 Results

Fig 5. Shows two interfaces of our program and player. We finally achieved a performance of 720p (30 frame/s) and 1080p (12 frame/s)



(a) (b)

Fig 5. (a) Order in server (Linux), (b)video in client (Windows)