A Real-Time Architecture of 360-Degree Panoramic Video Streaming System

Myungwoo Han,[1] Suk-Hwan Lee,[2] Sooyol Ok[3*]

[1]Dept. of Computers & Media Engineering, Tongmyong University, Busan, Republic of Korea
mwhan326@gmail.com
[2]Dept. of Information Security, Tongmyong University, Busan, Republic of Korea
skylee@tu.ac.kr
[3]Dept. of Game Engineering, Tongmyong University, Busan, Republic of Korea
sooyol@tu.ac.kr[*(CA)]

Abstract

In this paper, we discuss how to develop a system architecture that produces and transmits 360 degree panoramic video data with multiple cameras in real-time. Our system consists of three stages; panoramic video production using ORB(Oriented FAST and rotated BRIEF)[1] feature point detection, H.264 video compression format and RTSP(Real Time Streaming Protocol) streaming. In particular, 360 degree panoramic video is produced by acceleration of camera calibration, stitching, blending and encoding processes by GPU parallel programming named "CUDA". We measured the fps(frames per second) of panoramic video streaming that is transmitted to the client and verified that our system has at least 30fps at 4K output resolution. Thus, we could see that our system is suitable for 360 degree panoramic video streaming service in real time.

Key words: 360-degree panorama video streaming, GPGPU, Video stitching, RTSP streaming

Introduction

360 degree panoramic images not only provide users with a wide field of view (FOV) but also provide high sense of presence and immergence when using HMD(Head Mounted Display). Due to these advantages, interest in 360 degree images in the VR market is increasing as various 360 degree cameras are produced and viewers are available to view 360 degree panoramic images on social media platforms such as youtube, facebook, and Instagram. From now on, let us call a 360 degree panoramic image as a panoramic image.

Panoramic images are produced by image stitching, which makes by linking two or more images together into one image. Image stitching in this paper refers to putting multiple images in a frame by transforming and attaching multiple images from multiple cameras, and then projecting them into an extended shape by the Equirectangular projection. The process of image stitching is as follows. 1) Extract feature point for each image. 2) Perform the homography transformation to stick(or attach) images using feature points [2]. 3) Render to make the boundary of the attached image look natural [3]. 4) Project the transformed image into an Equirectangular Projection Frame.

Of these stages, feature extraction is the most important stage in image stitching, and should extract feature points that robust to the change of camera viewpoints or the lighting position. We can figure out whether feature points from images taken from multiple cameras represent the same point in the real world within the region of interest (ROI) where images overlap and we can decide how to transform the images in the process of matching those feature points.

Equivalent Projection Frame, one of the projection models, allows users to perform spherical transformations when viewing images. When viewing images using HMD or a 360-degree panoramic image-only viewer, it can actually feel as if they are surrounded by images, adding senses of high immersion and presence.

Since panoramic images need to be inputted from multiple cameras and be able to express their wide picture angles, their resolution naturally increases. For image raw data with 4K resolution (3840x1920) of the RGBA channel, the data size of one frame is about 28MB (Mega Byte) for panoramic image and it is about 56MB for HMD, where stereo images are required. With this complex process for producing panoramic image and high-resolution of panoramic image, it is very difficult to stream panoramic video in real time.

In this paper, we discuss the system architecture based on GPGPU and RTSP to solve the real-time problem of panoramic video streaming. The main content of our method is to generate panoramic images through the extraction of fast performing ORB feature points based on GPGPU, then to compress the high resolution panoramic images into H.264 and to transmit compressed images through the RTSP (Real Time Streaming Protocol) server.

Real-time 360-degree Panoramic Video Streaming System

In this chapter, we will first look at the main stages of Panoramic Video Streaming technology and then explain how to implement the real-time system.

A. Brief Description of Main modules

1) ORB Feature Detection

Our system uses ORB(Oriented FAST and Rotated BRIEF) algorithm based on FAST keypoint detector[4] and BRIEF descriptor[5] for a fast and robust feature detection. First, is uses FAST to find keypoints by using the difference of intensities of pixels within a circle radius around the pixel, then applies Harris corner measure to find top N points among them. However, since FAST cannot computer the orientation, the corner orientation of keypoints should be added for the invariance of rotation by measuring the corner orientation by

ISBN: 978-1-7281-0110-1

Rosin's method[6]. The keypoints with the corner orientation are defined by BRIEF descriptor that use Binary Test of pixels for feature matching. Here, the BRIEF descriptor has the problem of being variant to rotated image, especially in-plane-rotation[7]. Therefore, BRIEF is an extended descriptor that multiplies the rotation matrix to the Binary Test. ORB is much faster than Harris Corner Detector[8], SIFT(Scale Invariant Feature Transform)[9] and SURF(Speeded-up Robust Features)[10] and it works better than SURF. Therefore ORB is a good choice in low-power devices for panorama stitching.

Fig. 1. Feature Detection using ORB keypoint detector [11].

2) H.264 Encoding and RTSP

Since all frames of the panoramic video are raw data with vast amounts, our system compresses the panorama video data by the H.264 codec for streaming real-time panoramic video and transmits them to the client through RTP(Real-Time Transport Protocol) of NAL(Network Abstraction Layer). Here, NAL consists of parameters such as H.264 compressed data (VCL, Video Coding Layer), SPS(Sequence Parameter Set) and PPS(Picture Parameter Set) for decoding. When compressed panoramic video data is sent to the client through RTPS, which provides command/control of multimedia streams with higher-level protocols than RTP, the client can verify the video data by controlling remotely the media server through the commands specified in the RTSP convention.

B. System Implementation

The structure of the proposed system is shown in Figure 2. First, we get multiple images from 360 degree multi-cameras, and then perform the stitching process by GPU-based high-speed parallelism. Then, we compress the stitched of data of all frames in 360-degree panoramic video by the H.264 codec and then sent them to the client via the RTSP server.

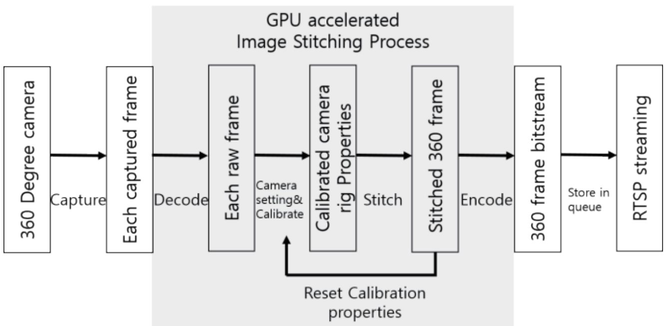


Fig. 2. Proposed 360degree Panoramic Video Streaming System.

1) 360-degree Panoramic Video Generation

Our system generates 360-degree panoramic image, calibrates multiple-cameras for high-quality panoramic image, and compresses 306-degree panoramic image data by H.264 codec

GPGPU based Panoramic Video Processing : Panoramic video processing consists of a series of processes on all frames, such as multiple-image acquisition from multiple-cameras, feature extraction and transformation for multiple-images, and natural blending. However, because of the high-resolution multiple-images of each frame in panoramic video, the panoramic video processing is very complex and requires a lot of computations to handle with the normal CPU. Therefore, our system speed up the panoramic video processing, which requires such a large amount of complex computations, to GPGPU(General-purpose computing on graphics processing units) based on GPU for computing purposes.

Calibration for Multiple-Cameras and Multi-Band Blending : Camera calibration and blending are required to reduce camera distortion and color difference for high quality panoramic video generation. Our system calibrates multiple-cameras using Camera Matrix with intrinsics elements and extrinsics elements of the camera for the input video image data be approximately the same as the actual shooting environment. Here the intrinsics elements of the camera are the focal length, the principal point, the distortion coefficient, and the image sensor information of the camera and the extrinsics elements of the camera are the position and rotation of the camera in the process of projecting the actual object to the image panel. Our system calibrate the camera intrinsics and extrinsics elements using the Camera matrix simultaneously. On the other hand, multiple cameras may have different colors for the same point by lighting or viewing of the camera. Our system blends multiple cameras through Multi-band Blending.

360-degree Panoramic Video Compression : The generated 360-degree panoramic video data is compressed by H.264 codecs with Hardware-Accelerated Video Encoding provided by NVIDIA's NVENC SDK and then it is transmitted to RTSP at H.264 bitstream.

2) Real-time Panorama Video Streaming

Our system implements the RTSP Server using Live555 media streaming library , an open source library that supports RTSP/RTP/RTCP. The RTSP Server initially becomes a read state for reading H.264 bitstreams. When the frame data of the compressed panoramic video is accumulated in the frame buffer, the RTSP Server detects the frame buffer event and puts the frame data into the payload, then sends the payload through the RTP to the RTSP Client.

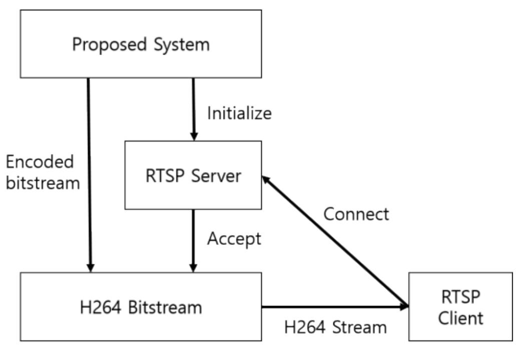


Fig. 3. RTSP Server call flow of proposed system.

ISBN: 978-1-7281-0110-1

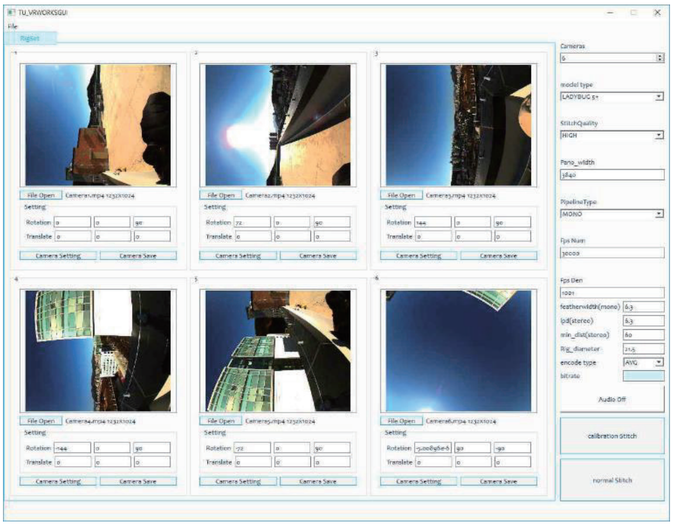
Figure 3 shows the RTSP call process in our system. Communication between the server and the client continues while the RTSP server reads a continuous H.264 bitstream and terminates at End Of File (EOF) in bitstream.

Experimental Results

This chapter evaluates the performance of our system by assessing visually the generated panoramic video images and by measuring FPS(Frame Per Second) for the processes of the panoramic video generation, compression, and streaming and then discusses the real-time performance from these experimental results.

A. Graphic User Interface of Our System

The verification process of camera calibration by the Camera Matrix setup determines whether panoramic video images are generated or video data is streamed to confirm externally. Through the GUI representing our system, we get the basic values of extrinsics such as rotation and position values by 6 cameras as shown in Figure 4-(a) and automatically calibrate 6 cameras. In addition to the basic values of extrinsics, the user can also set the detailed requirements for Camera Matrix, such as the focal length, main point, lens type, and distortion factor of the intrinsics, as shown in Figure 4-(a).



(a)

(b)

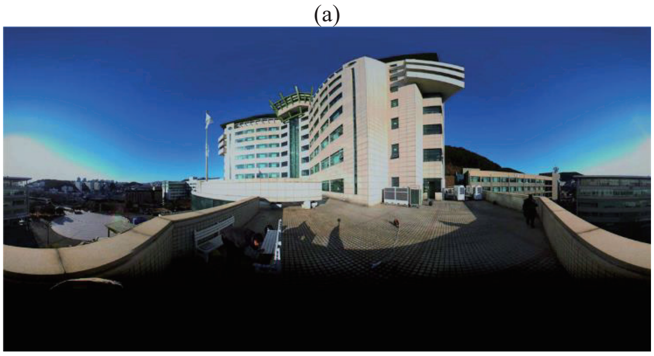
Fig. 4. Graphic user interface of our system.

After calibrating 6 cameras, we verify to ensure that the panoramic image is accurately produced, as shown in Figure 4-(b), by the stitched result of the first frames of 6 cameras.

B. Visual evaluation of 360-degree panoramic video image

Figure 5 shows a stitched frame in the form of Equilateral Projection in the generated panoramic video. The Equivalent Projection Frame can be viewed by using HMD or by using a 360-degree image only viewer to be converted into a Spiritual Projection shape. Looking at the stitched frame shown in Figure 5, we can see that the image of the frame provides a wide angle, as if it were surrounded in all directions, and provides a high sense of immersion. Panoramic video streaming can be viewed remotely through VLC Media Platter in a desktop environment or Stream's WanshiVR application in an HMD environment.

(a)



(b)

Fig. 5. Stitched Panorama image in proposed system (a),(b)

C. FPS of 360-degree panoramic video processing e

Our experiment measured fps(frames per second) of 360° panoramic video streaming based on the average 30-60 fps supported by social media platforms. It is natural that fps largely depends on the computational performance of the GPU. We used Multi GPU of Dual GTX 1080ti based on Pascal and observed the changes of fps measurable values by changing the resolution of 360° panoramic video. Here, the fps measurable value is the integer value that removes the fractional part from the average fps per minute.

1) Z-cam v1 pro camera

We used Z-cam v1 pro camera with 8 cameras of 3680x2428 resolution and then measured the fps value of 360° panoramic video processing, which was shown in Figure 6. 45 fps and 33 fps were measured at the output resolution of 2K (2560 x 1280) and 4K (3840 x 1920) at the transmission of stereo screens for viewing in HMD environments.

2) Ladybug5+ Camera

To evaluate the performance in different resolution and

ISBN: 978-1-7281-0110-1

camera environments, we used Radybug5+ camera consisting of 6 cameras with a 1232x1024 resolution and measured the fps value for 360° panoramic video processing. Looking at Figure 6 of the results, we know that 95 fps are measured at a resolution of 4K(3840x1920) and 126 fps at a resolution of 2K(2560x1280).

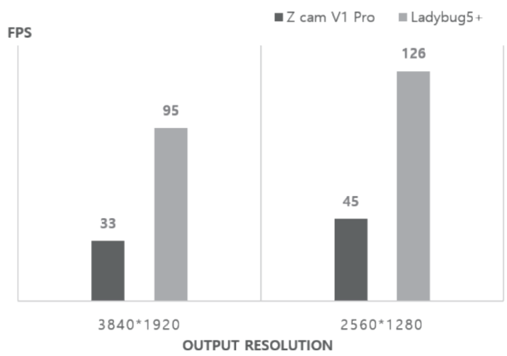


Fig. 6. Estimated average fps values in processing panoramic video captured by Z-Cam V1 pro and Ladybug5+

The measured results of the above two cameras verified that the computational processing time was affected not only by the output resolution of 360° panorama video, but also by the number of input cameras and input resolution that make up 360° camera. The computational processing time includes the panoramic video generation and streaming process from input multiple cameras and the decoding process to view. Our system can be found to be suitable for real-time by processing 360° panorama video with a maximum 4K output resolution from 6 cameras with a maximum input resolution of 3680x2428 at about 33fps.

Conclusions

In this paper, we proposed a 360-degree panorama video streaming technology based on GPGPU, and confirmed from Z-cam v1 pro camera and Ladybug5+ camera experiments that our system is suitable for real-time 360-degree panorama video streaming service up to 4K output resolution. The proposed system not only provides 360° panoramic video streaming service but also has high value in real-time field of high-definition video processing. However, our system takes up a large portion of the time to receive video image data from multiple cameras. Solving this problem needs to build buses that receives video image data from multiple cameras without delay and needs to align and match the video image data directly without decoding. As future works, we will implement the real-time live 360-degree panoramic video streaming service that produces accurate video stitching, such as fast reading of video image data taken from multiple cameras, and adjusting time-difference between input video image data.

Acknowledgement

This research is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No.2016R1D1A3B03931003) and also supported by project for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Ministry of SMEs and Startups in 20 (S2659910).

References

[1] E. Rublee, V. Rabaud, K. Konolige, and G.R. Bradski, *International Conference on Computer Vision (ICCV)*, November 2011.

[2] C. Loop and Z. Zhang, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, pp. 125-131, June 1999.

[3] Multiband Blending, https://developer.nvidia.com/vrworks/vrworks-360video/ Multiband-Blending, Accessed on May 2019.

[4] E. Rosten and T. Drummond, *European Conference on Computer Vision (ECCV)*, pp. 430-443, May 2006.

[5] M. Calonder, V. Lepetit, C. Strecha, and P.Fua, *European Conference on Computer Vision*, pp. 778-792, September 2010.

[6] P. L. Rosin, *Computer Vision and Image Understanding*, 73(2), pp. 291-307, February 1999.

[7] Plane of rotations, https://en.wikipedia.org/wiki/Plane_of_rotation, Accessed on May 2019.

[8] C. Harris and M. Stephens, *Alvey Vision Conference*, pp. 147-151, 1988.

[9] D. G. Lowe, *International Journal of Computer Vision*, 60(2), pp. 91-110, November 2004.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, *European Conference on Computer Vision (ECCV)*, pp. 404-417, May 2006.

[11] Calibrating 360 video, https://devblogs.nvidia.com/ calibrating-videos-vrworks-360-video, Accessed on May 2019.

ISBN: 978-1-7281-0110-1