END-to-END STEREOSCOPIC VIDEO STREAMING SYSTEM

Selen Pehlivan[1], Anil Aksay[2], Cagdas Bilen[2], Gozde Bozdagi Akar[2], M. Reha Civanlar[1]
1 College of Engineering, Koç University, Istanbul, Turkey
2 Electrical and Electronics Engineering Department, Middle East Technical University
Ankara, Turkey
{spehlivan, rcivanlar}@ku.edu.tr, {anil, cbilen, bozdagi}@eee.metu.edu.tr

ABSTRACT

Today, stereoscopic and multi-view video are among the popular research areas in the multimedia world. In this study, we have designed and built a platform consisting of stereo-view capturing, real-time transmission and display. At the display stage, end users view video in 3D by using polarized glasses. Multi-view video is compressed in an efficient way by using multi-view video coding techniques and streamed using standard real-time transport protocols. The entire system is built by modifying available open source systems whenever possible. Receiver can view the content of the video built from multiple channels as mono or stereo depending on its display and bandwidth capabilities.

1. INTRODUCTION

An end-to-end 3D video system should accommodate selective transmission of mono or stereo video depending on the available bandwidth or the user's receiver equipment. A system based on independent transmission of the two channels of stereo video can be used in order to achieve this purpose. Moreover, such a system can be built by modifying existing platforms built for regular video streaming.

Among the existing open source video streaming platforms, we investigated Darwin Streaming Server [1], GPAC [2] and VideoLAN Client/Server [3]. Apple QuickTime Streaming Server (QSS) and its open source version Darwin Streaming Server (DSS) supports streaming of H.264 [4] coded video wrapped inside MPEG-4 [5] or 3GPP [6] file format across the Internet using RTSP and RTP protocols [7]. In order to stream video, these systems need special information about the media files. This information is carried in a track called hinted track and hinted tracks are only used when streaming media over RTP. Another project called GPAC, which is developed as a multimedia framework based on MPEG-4 standard, also supports streaming H.264 coded media files inside MPEG-4 file format [2]. In addition to these two platforms, VideoLAN Client (VLC) also provides streaming capabilities. VLC supports H.264 video in local playback and streaming when encapsulated in MPEG-TS file format over RTP. It can open H.264 coded media files in MPEG-4 file format streamed by DSS, but it can not stream them over RTP by itself using any file formats other than MPEG-TS. Our main goal while investigating these projects was to adopt their mono streaming features to our stereo streaming platform.

Besides monoscopic streaming platforms, a 3D TV prototype system with real-time acquisition, transmission and auto-stereoscopic display of dynamic scenes is offered by MERL. This system is composed of a multi-projector 3D display, an array of cameras and network connected PCs. Multiple video streams are individually encoded and sent over a broadband network to the display. The 3D display shows high-resolution stereoscopic color images for multiple viewpoints without special glasses. This system uses light-field rendering to synthesize views at the correct virtual camera positions [8].

We developed a stereo streaming system using current standards for H.264 video coding and real time streaming features. We chose to stream views as H.264 coded videos because of its coding efficiency.

The H.264 video codec has a very wide range of applications covering low bit-rate Internet streaming applications, HDTV broadcast and Digital Cinema applications. Compared to the current sate of technology, H.264 is reported to provide bit rate saving of 50% or more [9]. The codec specification contains two conceptual layers, a video coding layer (VCL) and a network abstraction layer (NAL). The VCL contains traditional video coding steps and NAL specifies the encapsulation and transformation format that are suitable for the underlying network. NAL encapsulates the slice output of VCL into Network Abstraction Layer Units which are well suited for transmission [4]. In this study, we used new NAL unit types which are specified in RTP Payload format for H.264.

The rest of the paper is organized as follows: In Section 2, we present an overview of the system. Acquisition, encoding, transmission, decoding and display parts of the end-to-end system and its architecture are presented in detail. Section 3 presents results and section 4 contains the concluding remarks.

2. SYSTEM OVERVIEW

A general diagram of the system architecture is shown in Figure 1.

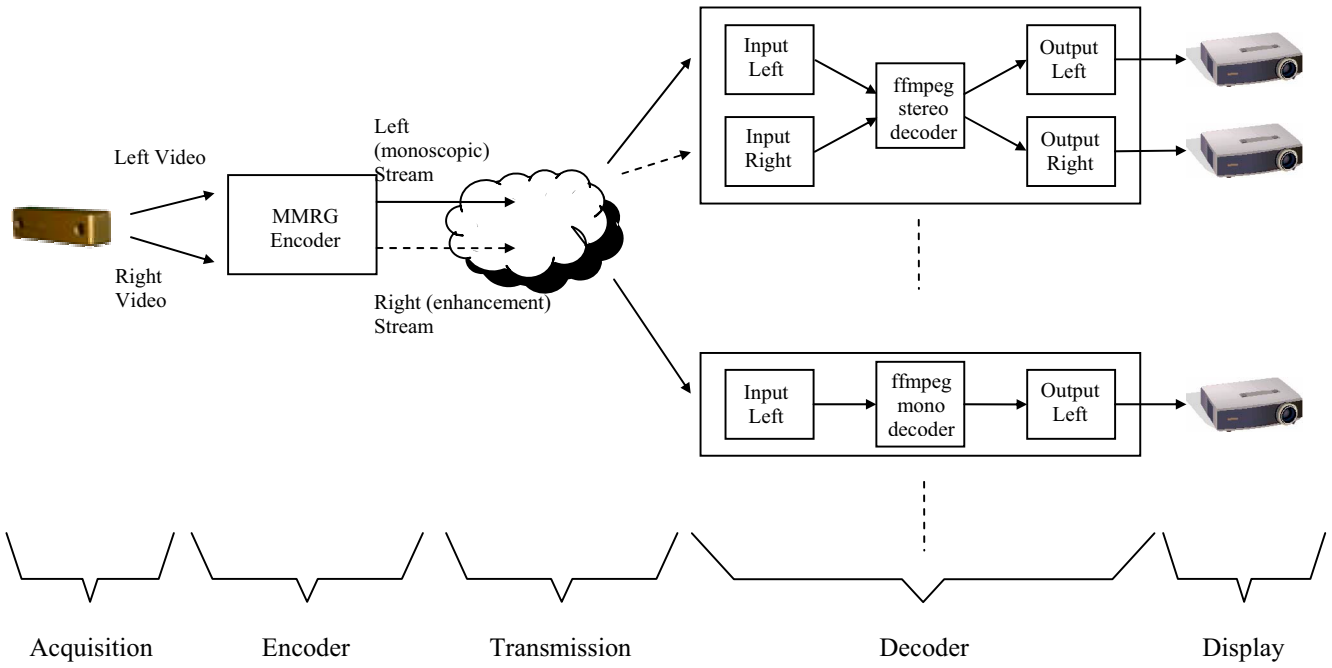

Figure 1: End-To-End System Overview

Stereo images captured by the stereoscopic camera are encoded by using an extended H.264 codec [11]. Due to the low encoding speed of the implemented codec, real-time encoding is not yet possible. As a result, previously encoded videos are streamed instead of live video. However, real-time implementation of this codec using TI642 is currently in progress. The coded bitstreams are transmitted over the Internet on separate channels by the streaming server. The end users can view either monoscopic or stereo streams based on their display capabilities, using the specialized player. If end user system can support stereo video then both channels are utilized by the decoder, otherwise only the left channel will be received and displayed. In the following sections, the system components are explained in detail.

2.1. ACQUISITION

In our system, Bumblebee Stereoscopic Camera [10] is used to capture stereoscopic video sequences. As shown in Figure 1, this camera consists of two Sony ICX424 1/3" progressive scan CCD sensors with the resolution of 640x480. It can capture up to 30fps. Camera control and data transfer is possible through high-speed IEEE-1394 digital communication.

2.2. ENCODER

The H.264/MPEG-4 design covers a Video Coding Layer (VCL), which efficiently represents the video content, and a Network Abstraction Layer (NAL), which formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by particular transport layers (such as Real Time Transport Protocol) or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL Units is the same for both packet-oriented transport and bitstream delivery. The only difference is each NAL unit can be preceded by a start code prefix in a bitstream-oriented transport layer [4].

The encoding phase of the stereo codec is based on the MMRG Multiview Codec [11] which is developed based on H.264. The codec structure is shown in Figure 2. The codec is used in two camera-stereoscopic mode with standard compatibility option enabled. In this mode, left frames are predicted only from left frames. So just using the left NAL Units, left video can be decoded with a standard AVC decoder. However, right frames are predicted from both left and right frames to reduce the bandwidth. Therefore, to decode right video, both left and right NAL Units are required.

In order to cope with packet losses, frequent intra frames are inserted and frames are coded in slice mode where whole frame is a slice. Although the system is tested by encoding each frame as a single slice, number of slices can be increased or fixed size slices can be used according to the network state.

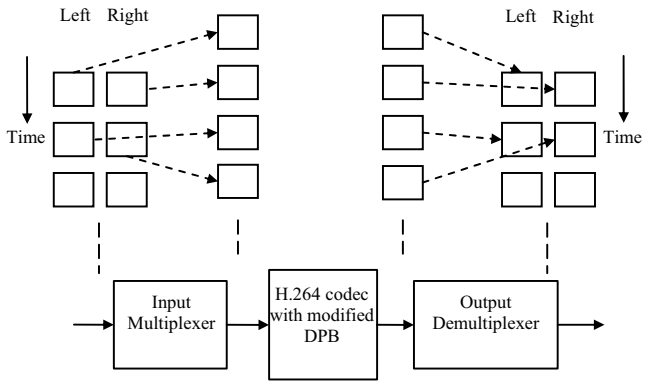

Figure 2: Stereoscopic Encoder

2.3. TRANSMISSION

This part explains Sender side and Client side of the developed system and their architectures.

2.3.1. Sender

A bitstream encoded by MMRG Multiview Codec contains NAL units of both left and right views. These NAL units are prepared to be streamed over separate channels. The NALUs are packetized before being transmitted over the network. The sender side packet format is implemented based on the RTP payload format for H.264 video [12]. Three packetization modes are defined in this payload format. We implemented Single NAL Unit mode and Non-interleaved mode which are intended for low-delay applications. Single NAL Unit Mode allows transmission of a single NAL unit and Non-Interleaved Mode uses single NAL unit packets, STAP-As(Single-time Aggregation packet without Decoding Order Number) and FU-As (Fragmentation Unit without Decoding Order Number) [12]. We used FU-As packetization structure to transfer NALUs the sizes of which are exceeding the network MTU. So we fragmented the NALUs on the application layer instead of relying on the IP layer fragmentation. Other packets with smaller sizes are sent in Single NAL unit packets.

In both of these packetization modes, the transmission order of the RTP packets shown by the sequence numbers is taken as the decoding order of the NAL units. Since our encoder does not support B frames, packet structures which do not contain decoding order numbers are usable in our application. The timestamp carried on the RTP header are used to arrange the decoding order of the frames.

The RTP timestamps are used to synchronize the frames. The display application arranges the play-out time by using the relative order of the frames positioned by the RTP timestamps. Since we stream two video files, we set related frames to the same timestamp supposing same sampling rate for videos with a 90 kHz clock.

In addition to these, the H.264 parameter sets are fundamental parts for video coding. A more reliable transfer is required for their transmission and receiver must receive them before the decoding process. So we transfer them out-of-band to the receiver side reliably prior to the actual RTP sessions as in [13].

For conveying session the video parameters from the stereo video streamer to the developed player on the receiver side, we used session description protocol (SDP) [14]. For indicating stereo view, an additional session attribute is used specifying stereo data and the left and the right channels. For future extension of stereo streaming to multi-view streaming, the same session descriptor can be used. Additionally, we defined a new attribute "view" which gives the address and the port information of the other sessions broadcasting extra views of the video.

a=view : mono
a=view : stereo <address-Left> <port-Left>
<address-Right> <port-Right>

a=view : multi <address> <port> <address> <port>,
<address> <port> <address> <port>,
<address> <port> <address> <port>, ...

where "mono" for monoscopic, "stereo" for stereoscopic and "multi" for multi-view gives the view type and "<address> <port> <address> <port>" pair gives the access information of two corresponding views of the multi-view video.

2.3.2. Receiver

To be used as a player, open source software VideoLAN Client (VLC) is modified for our purpose. VLC is a highly portable multimedia player for various audio and video formats (MPEG-1, MPEG-2, MPEG-4, DivX, mp3, ogg ...) as well as DVDs, VCDs, and various streaming protocols [3]. Although, the VLC player can play raw H.264 bitstreams from the local disk, VLC system does not support raw H.264 streaming over RTP. Currently, VLC only supports streaming of MPEG-TS file format over RTP. Thus, we used VLC as a player and modified its stream receiver.

The modified VLC handles packets of left and right views using two separate threads. Then, corresponding decoder for H.264 coded data is opened by the player. Since the received data is in raw data format, we do not need demux operation and we send NALU units received in RTP packet payload directly to the decoder after a simple depacketization.

As the decoder, open source H.264 decoder implementation inside the ffmpeg library is used [15]. Each raw data stream processed inside the player is called an elementary stream. We have two elementary streams corresponding to left and right views. Before sending to the decoder, the data are buffered in order to synchronize related left and right frames. The decoder decodes these elementary streams and sends the decoded picture to the video output modules. The video output units visualize the left and right frames in a synchronized manner by using the time information in the RTP timestamps.

We are also working on defining a file format for multi-view video streaming of H.264 over RTP which includes information about how many views are there, which two of them we use, the frame rate, etc. Also, we hope to increase the concurrent streaming speed of video files for each view by decreasing the file read time on the disk. However, in this case, each bitstream received is unwrapped and processed as a separate elementary stream and the remaining process will be the same.

2.4. DECODER

MMRG Multi-view Codec is developed based on JM Reference Software [16] of H.264 which has been developed without much concern on speed optimization. Therefore, it is not possible to decode the received stream in real-time. As a result, we improved the implementation of H.264 decoder inside ffmpeg [15] which can decode H.264 streams in real time with the same structure of MMRG Multi-view Codec Decoder in order to support stereo view decoding in our system.

2.5. DISPLAY

The display system consists of two Sharp MB-70X DLP projectors and a silver screen as shown in Figure 3. Light from projectors are polarized using circular polarized filters. One filter polarizes light of one projector in right circular way and other filter polarizes light of the second projector in the left circular way. Both projectors

projects onto a silver screen. This screen is covered with a dielectric material which keeps the polarization of light coming from the projectors. The users wear glasses which guarantee to provide the left and right eyes the corresponding left and right images reflected from the screen using filters matching with the projector's filters. The projectors' inputs are taken from a high performance PC which has 2-VGA graphic card. Using extended desktop feature of the operating system, left views from one projector is shown on the left half of the desktop and right views projected from other projector are displayed on the right half of the desktop. Each projector shows one half of the extended desktop. So we obtain two overlapped frames seen as 3D images by the users.

Figure 3: Stereoscopic Display System

3. RESULTS

For transmission and display process, we have implemented all the modules and run the system with already encoded files. We tried the system using different videos and the video quality is found quite satisfactory by viewers. The system is initially tried on a local area network with no packet losses.

The H.264 coded video increased the efficiency of bandwidth usage and this also affects the quality of the views. We coded videos as 25 fps and we inserted one intra frame per 12 frame. Two different quantization values were used, one has Y channel PSNR value of 38.27 dB and other one has 33.47 dB. For 320x240 video, their bandwidth usage were 744.665 and 415.335 kbits/sec respectively. The videos that are shot by the proposed camera setup are used as the test videos can be downloaded from http://mmrg.eee.metu.edu.tr/stereovideo/.

The system scalability was also an important factor for the system design. The data can be multicasted from anywhere and the users can view as mono or stereo depending on their connection capacity and display system. Moreover the player functionality and integrity also increases the usage of system with the future improvements on different file formats and codec standards.

4. CONCLUSIONS AND FUTURE WORK

In this study we have implemented an end-to-end stereoscopic video streaming system using open source components with minor modifications.

In the future, we will use an hardware encoder to achieve real-time encoding and streaming of live video. We will try to improve coding efficiency to reduce the bandwidth requirements. We will also investigate the system under network conditions with considerable packet losses. In addition to these, we plan to increase the streaming speed by defining a new file format for multi-view video files. This file format will also define the multi-view video specification.

Audio support for our system will be another consideration under this project.

5. ACKNOWLEDGEMENTS

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

MMRG H.264 Multiview Extension Codec can be downloaded from http://mmrg.eee.metu.edu.tr/multiview

6. REFERENCES

[1] http://www.apple.com/

[2] http://gpac.sourceforge.net/

[3] http://www.videolan.org/vlc/

[4] ITU-T ISO/IEC 14496-10, "Recommendation H.264: Advanced video coding for generic audiovisual services," May 2003

[5] ISO/IEC International Standard 14496 (MPEG-4); "Information technology - Coding of audio-visual objects", January 2000.

[6] http://www.3gpp.org/

[7] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[8] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems", Picture Coding Symposium (PCS), December 2004

[9] A. Luthra, G.J. Sullivan, T. Wiegand (eds.), "Special Issue on H.264/AVC", IEEE Transactions on Circuits and Systems on Video Technology, July 2003.

[10] http://www.ptgrey.com/products/bumblebee/index.html

[11] C. Bilen, A.Aksay, G. Bozdagi Akar, "A Multi-View Video Codec Based on H.264", ICIP 2006 (submitted).

[12] S. Wenger, M. M. Hannuksela,T. Stockhemmer, M. Westerlund, D. Singer, "RTP Payload Format for H.264 Video", RFC 3984, February 2005.

[13] M. Civanlar, G. Cash, B. Haskell, "AT&T's Error Resilient Video Transmission Technique", RFC 2448, November 1998.

[14] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[15] http://ffmpeg.sourceforge.net/

[16] http://iphome.hhi.de/suehring/tml