Stereo Video Broadcasting Simulation for DVB-H

M. Oguz Bici[1], Anil Aksay[1], Antti Tikanmaki[2], Atanas Gotchev[2], Gozde Bozdagi Akar[1]

[1] Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey
[2]Department of Signal Processing, Tampere University of Technology, Tampere, Finland

E-mail: {mobici, anil, bozdagi}@eee.metu.edu.tr, {tikanmaa@cs, atanas.gotchev@}tut.fi

Abstract: **This paper presents modelling and analysis of a mobile 3DTV broadcast system for handheld devices. The underlying technology behind our system is the DVB-H specification which brings ordinary TV broadcast services to battery-powered handheld receivers. In our system, we simulated the broadcast of stereo video content over DVB-H to mobile handheld devices with autostereoscopic displays. In this way, mobile users can watch 3D content without need for eye-glasses or any special equipment.**

Keywords: Mobile 3DTV, DVB-H, stereo video streamer

1 INTRODUCTION

Television has been evolving gradually from black-and-white to colour to high definition. The next awaited step is the introduction of 3D. Compared to the rather conservative market of big TV sets and traditional broadcasters, the market for mobile devices has always been much more dynamic and receptive to new technologies. Recently, dedicated mobile television systems have been introduced. The most mature of these is the European system based on the DVB-H standard. DVB-H is based on the terrestrial digital video broadcasting (DVB-T) standard. It is designed to improve performance in mobile environments, to add flexibility in network planning and to enable efficient power control in handheld receivers.

In the literature there exists a few systems for stereo video streaming. In [1], a 3DTV prototype system, with real-time acquisition, transmission and autostereoscopic display of dynamic scenes, has been introduced by MERL. Multiple video streams are encoded and sent over a broadband network. 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. In [2], a hand-held device with autostereoscopic display is proposed which plays stereo videos locally. In [3], a stereoscopic 3D video streaming server and clients have been introduced by modifying available open source platforms. Broadcasting 3D video content over T-DMB to mobile terminals has been addressed in [4] In this paper, we extend this study to stereo video streaming over DVB-H.

2 SYSTEM OVERVIEW

The building blocks of the proposed system can be seen in Figure 1. Stereo video content with right and left view is first compressed with a stereo video encoder. Resulting Network Abstraction Layer (NAL) units (NALU) are fed to the stereo video streamer. The streamer encapsulates the NAL units into Real Time Transport Protocol (RTP), User Datagram Protocol (UDP) and finally Internet Protocol (IP) datagrams. The resulting IP datagrams are encapsulated in the DVB-H link layer where the Multi Protocol Encapsulation Forward Error Correction (MPE-FEC) and time slicing occurs. The link layer output MPEG-2 Transport Stream (TS) packets are passed to physical layer where the transmission signal is generated with a DVB-T modulator. After the transmission over a wireless channel, the receiver receives distorted signal and possibly erroneous TS packets are generated by the DVB-T demodulator. The errors are tried to be corrected in the link layer by the MPE-FEC functionality and the TS packets are decapsulated into IP datagrams. IP datagrams are handled in the stereo video streamer client and resulting NAL units are decoded with the stereo video decoder to generate right and left views. Finally these views are put into an appropriate format to be displayed as 3D in the displayer. The details of each aforementioned block are described in following sections.

3 STEREO VIDEO ENCODER

Currently, there are a few candidate formats for delivery and storage of 3D video. The first alternative is to transmit a pair of stereoscopic views captured by stereoscopic cameras. These views can be either coded separately, also known as simulcast [5], or they can be coded together using a multi-view codec such as the MVC extension of H.264 [6]. Multi-view codecs achieve higher coding efficiency than simulcast by exploiting the inter-view correlation in compression.

Alternatively, the 3D video can be represented as a conventional monoscopic video sequence that is augmented with depth information. In this case, the

Corresponding author: M. Oguz Bici, Middle East Technical University, Ankara, TURKEY, +903122104509, mobici@eee.metu.edu.tr

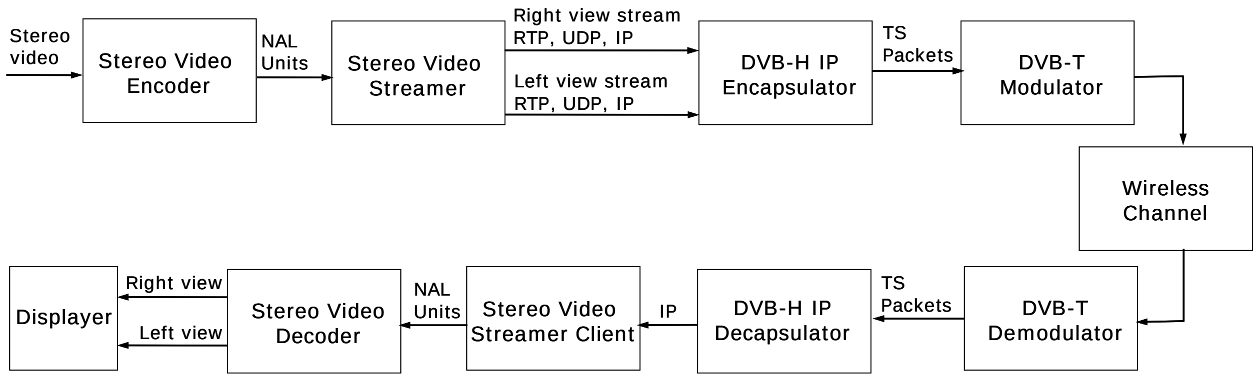

Figure 1 The system overview

receiver has to take care of generating the views appropriate for the display device.

There are three main ways to encode these representations as shown in Figure 2. Our system supports all of them.

a) Encoding both views independently using H.264/AVC. Our system uses H.264/AVC reference encoder version 10.1 [7]. b) Video-plus-depth (VPD) representation is coded by independent coding of video and depth signal by H.264/AVC with small information about depth data embedded into the high-level syntax. MPEG specified a corresponding container format "ISO/IEC 23002-3 Auxiliary Video Data Representations", also known as MPEG-C Part 3 [8]. Similarly same encoder is used with extra syntax.

c) Exploiting temporal and inter-view redundancy by interleaving camera views and coding using a hierarchical manner with some Multi-View Video Coding (MVC) specific tools like illumination and color compensation, improved disparity estimation and coding and some high-level syntax changes. MVC is decided to be an amendment (Amendment 4) to H.264/AVC which is scheduled to be finalized in early 2008 [9]. MVC encoder used in our system is JMVM 3.0.2 [10]. Main prediction structure is quite complex introducing a lot of dependencies between images and views. These dependencies make use of the redundancies present in both spatial and temporal directions to reduce the bitrate, however they also impose many restrictions in decoding and packet loss sensitivity. An alternative simplified structure is presented in [11], and shown to be very close to the main prediction structure in terms of overall coding efficiency. In this simplified prediction structure the temporal prediction using hierarchical B-pictures remains unchanged when compared to original MVC prediction structure, but spatial references are only limited to anchor frames, such that spatial references are only allowed at the beginning of a group of pictures (GOP) between I and P pictures.

4 STEREO VIDEO DECODER

We are currently using FFMPEG library [12] for real-time decoding of H.264/AVC streams. In order to decode MVC streams, we modified FFMPEG library with the appropriate syntax changes and interleaving of two views.

In order to cope with losses, decoder examines picture order count (POC) of each frame and identifies missing frames. Missing frames are then concealed by frame copy using the nearest available frame from the same view.

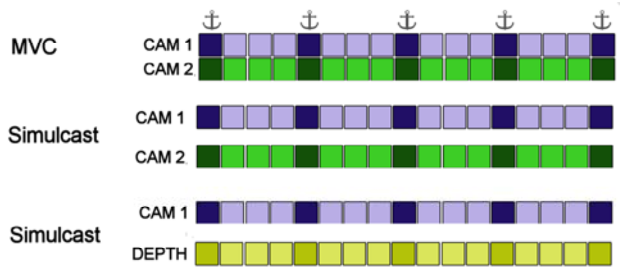

Figure 2 Supported Stereo Video Coding

5 STEREO VIDEO STREAMER/CLIENT

The streaming of H.264/AVC streams over RTP is standardized by the IETF in RFC 3984 [13], which defines the RTP header usage and necessary packetization rules for H.264/AVC. For simulcast coded representations (a-b), [13] is used. For MVC (c), stream packetization is applied in a similar fashion to [13] as proposed in [3]. Instead of sending the MVC stream as a single H.264/AVC stream over RTP, the stream is broken into two parts. NALUs are streamed over two different RTP port pairs as if they are separate H.264/AVC streams.

6 DVB-H LINK LAYER

Transmission of 3D video over an end-to-end DVB-H channel is simulated using a software system that implements the transmitter and receiver functionality. On the transmitter-side, an IP encapsulator program packetizes and multiplexes the IP packet streams in a DVB-H compatible time-sliced Transport Stream. Erroneous transmission is simulated by filtering the TS with an appropriate physical channel model. On receiver-side the stream is decapsulated and the FEC data is decoded to recover possibly lost data. The received IP packets are finally delivered to the H.264/MVC client for decoding of the 3D video. This process as well as the software environment used to implement the DVB-H simulator is explained in [14].

7 PHYSICAL LAYER

The time slicing and MPE-FEC technology elements on the link layer does not touch the DVB-T physical layer in any way. Therefore the existing receivers for DVB-T are not disturbed by DVB-H signals and DVB-H is totally backward compatible to DVB-T. However there are additional modes in DVB-H physical layer like 4K mode and in-depth interleaver. For more information on modulation/demodulation techniques in DVB-H physical layer, reader is referred to [15], [16].

In our system, we realize physical layer in two ways: real transmission or software simulation. In real transmission, we broadcast with a hardware modulator and receive the signal by another device which has a DVB-T demodulator. In software simulation, we model the transmission parameters and the channel with Matlab Simulink.

8 DISPLAYER

The display unit of the clients is an autostereoscopic display which allows a 3D vision without any need for eye glasses or special equipment. In an autostereoscopic display, the right and the left views go into an interdigitation process in which an appropriate picture is generated from the pixels of left and right views. In the displayer of our system, as a pair of left and right frames is decoded, the interdigitation process takes places and the generated image is displayed on the autostereoscopic display.

9 BACKWARD COMPATIBILITY

An important feature in our system is backward compatibility, i.e. existing mobile users without 3D features are able to receive stereo video broadcast and play the monoscopic video (either right view or left view). The backward compatibility is accomplished in two ways: Either in application layer or in link layer.

The application layer approach is straightforward and easy to implement. In this approach, the stereo video is compressed such that a non-stereo compliant decoder can decode one of the views from the compressed bitstream. The disadvantage of this approach is that, the user needs to receive all the data although the data corresponding to one of the views is not necessary.

In the link layer approach, the IP datagrams corresponding to right and left views are encapsulated in the link layer and put into different ESs with different PIDs. In this way, the data for left and right view are transmitted in different bursts so that a non-stereo compliant user can play monoscopic video by only receiving corresponding ES. In this approach, the IP decapsulator is modified so that the receiver knows that it needs to filter two PIDs for left and right view ESs.

10 EXPERIMENTAL RESULTS

To validate the performance of the simulation setup, we performed a simple transmission experiment where an MVC coded test sequence is transmitted over a simulated DVB-H channel. The test sequence was encapsulated in a Transport Stream by putting each view in its own MPE stream. The erroneous channel conditions were simulated on the physical layer using the previously mentioned Simulink model of the transmission system. Six different channel conditions with SNR between 10 and 15dB were tested and the simulation results were saved as TS level packet error traces [14].

The error traces were then mapped on to the test transport stream in order to introduce errors. We evaluated the impact of MPE-FEC by comparing two scenarios when extracting the H.264/MVC video from the TS: 1) Client receiving the FEC codes within the MPE-FEC frame, 2) Client receiving only the data part of the MPE-FEC frame where power consumption is reduced. Finally, the quality of received and decoded video was measured by computing Y-PSNR of the views using original, uncompressed video as the reference. This simulation process was repeated 50 times for each channel SNR condition in order to obtain figures corresponding to the average behaviour of the channel. The average PSNR of the stereo video together with no loss case are shown in Figure 3. The results show a clear improvement of quality when the received MPE-FEC data is used especially in low channel SNR cases. Another observation is that using MPE-FEC frame achieves almost no loss performance for channel SNR values higher than 13.

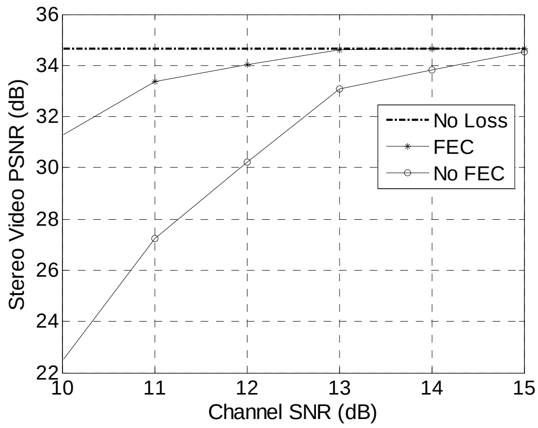

Figure 3 Simulation results

11 CONCLUSION

In this paper, we have presented a Mobile 3DTV system architecture which handles stereo video broadcast using the DVB-H specification. To the best of our knowledge, this is the first DVB-H based 3D broadcast system proposed in the literature. The system is backward compatible, i.e. existing mobile users without stereo support can receive and view monoscopic video. Our system can be used for real-time broadcasting and receiving as well as offline software simulations. The future work will concentrate on optimizing link layer and physical layer parameters to achieve best video quality in case of transmission errors.

Acknowledgments

MOBILE3DTV project has received funding from the European Community's ICT programme in the context of the

Seventh Framework Programme (FP7/2007-2011) under grant agreement n° 216503.

The text reflects only the authors views and the European Community or other project partners are not liable for any use that may be made of the information contained herein.

The early version of the streamer was developed under EU 3DTV Project.

References

[1] Matusik, W.; Pfister, H., "3D TV: A Scalable System for Real-Time Acquistion, Transmission and Autostereoscopic Display of Dynamic Scenes", ACM Transactions on Graphics (TOG) SIGGRAPH, ISSN: 0730-0301, Vol. 23, Issue 3, pp. 814-824, August 2004 (ACM Press)

[2] Kai Willner, Kemal Ugur, Marja Salmimaa, Antti Hallapuro, Jani Lainema, "Mobile 3D Video Using MVC and N800 Internet Tablet," 3DTV-CON 2008, Istanbul, Turkey

[3] E. Kurutepe, A. Aksay, C. Bilen, C. G. Gurler, T. Sikora, G. Bozdagi Akar, A. M. Tekalp, "A Standards-Based, Flexible, End-to-End Multi-View Video Streaming Architecture," Packet Video Workshop 2007, Lausanne, Switzerland

[4] S. Cho, N. Hur, J. Kim, K. Yun, and S-I. Lee, "Carriage of 3D audio-visual services by T-DMB", Electronics and Telecommunications Research Institute, Republic of Korea, in Proc ICME 2006.

[5]I. Rec, "H. 264 & ISO/IEC 14496-10 AVC, Advanced video coding for generic audiovisual services," ITU-T, May, 2003.

[6]K. Müller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Wiegand, Multi-view Video Coding Based on H.264/AVC Using Hierarchical B-Frames, in Proc. PCS 2006, Picture Coding Symposium, Beijing, China, April 2006.

[7] JM Reference Software Version 10.1. http://iphome.hhi.de/suehring/tml/download/

[8] A. Bourge and C. Fehn (Editors). White Paper on ISO/IEC 23002-3 Auxiliary Video Data Representations. ISO/IEC JTC 1/SC 29/WG 11, WG 11 Doc. N8039, Montreux, Switzerland, April 2006.

[9] A. Vetro, P. Pandit, H. Kimata, and A. Smolic, "Joint Draft 3.0 on Multiview Video Coding," Joint Video Team, Doc. JVT-W209, 2007.

[10] P. Pandit, A. Vetro, and Y. Chen, "JMVM 3 software," ITU-T JVTV208, 2007

[11] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Comparative Study of MVC Prediction Structures," ITU-T JVT-V132, 2007.

[12] Ffmpeg homepage. [Online]. Available: http://ffmpeg.sourceforge.net/

[13] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RFC 3984: RTP payload format for H.264 video." [Online]. Available: http://tools.ietf.org/html/rfc3984

[14] M. Oksanen, A. Tikanmäki, A. Gotchev and I. Defee, "Delivery of 3D Video over DVB-H: Building the Channel", NEM Summit, 2008

[15]G. Faria , J. A. Henriksson , E. Stare and P. Talmola "DVB-H: digital broadcast services to handheld devices," Proc. IEEE, vol. 94, pp. 194, Jan. 2006.

[16] ETSI, EN 300 744, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television." V1.5.1 (2004-11)