Evaluation of Stereo Video Coding Schemes for Mobile Devices

Anil Aksay
Department of Electrical and Electronics Engineering,
Middle East Technical University, Ankara, Turkey
+903122104509
anil@eee.metu.edu.tr

Gozde Bozdagi Akar
Department of Electrical and Electronics Engineering,
Middle East Technical University, Ankara, Turkey
+903122102307
bozdagi@eee.metu.edu.tr

ABSTRACT
Mobile devices such as mobile phones, personal digital assistants and personal video/game players are somehow converging and getting more powerful, thus enabling 3D mobile devices a reality. In order to store or transmit stereo video in these devices, coding techniques from both monoscopic video coding and multi-view video coding can be used. In this work, we analyze the possible stereoscopic encoding schemes for mobile devices. We have used rate-distortion curves for coding efficiency and decoding speed tests for decoder complexity. Depending on the processing power and memory of the mobile device, we concluded to use two of the settings used in our experiments.

Keywords
Stereo video coding, DVB-H,

1. INTRODUCTION
3D Video is a new area and getting very popular with advances in display technologies. There are several research projects working on capture, representation, rendering and transmission of 3D Video under various research programs such as ATTEST project [1], 3DTV, 3Dphone, 3D4you, 3DPresence, Mobile3DTV projects [2]in Europe and FTV [3] project in Japan.

Meanwhile mobile devices such as mobile phones, personal digital assistants and personal video/game players are somehow converging and getting more powerful, thus enabling 3D mobile devices a reality. The most challenging technical issues for commercializing a 3D mobile device are the stereoscopic display technology that is suitable for mobile devices, the renderer and an efficient video coding standard to represent 3D video.

In literature, there exist a few 3D mobile device prototypes [6]-[9] which are based on auto-stereoscopic 3D displays either parallax barrier or lenticular lens structures or stereoscopic display which can be observed with anaglyph glasses. For coding, they use different technologies. In [6], stereo video is encoded using H.264/AVC MVC extension; however some of the tools such as Hierarchical B pictures are not used to decrease decoding complexity. In [7], stereo video of QVGA size is coded using simulcast MPEG-4 encoder with asymmetric coding (Left and right videos are encoded with different resolutions). In [8], both stereo video and video plus depth representations are used. Stereo video of QVGA size is first converted into monoscopic video by tiling the images (side-by-side) and then encoded with MPEG-4 (simple profile). Video and depth are encoded by MPEG-4 as separate streams. In both representations, 24 fps can be achieved on the decoder side. In [9], stereo video is fed into H.264/AVC monoscopic video encoder as an interlaced video. 10 fps can be achieved on the decoder side.

Even though in this preliminary studies different techniques are used, in order to efficiently store or transmit stereo video to mobile devices, coding techniques from both monoscopic video coding and multi-view video coding should be examined in detail. In this work, we tried to achieve this task, i.e. examine the video codec performances for stereoscopic videos with mobile device resolutions with different profiles. In Section 2, a detailed description of tools used in both H.264/AVC and its MVC extension is presented. Section 3 present results and Section 4 contains the concluding remarks.

2. STEREO VIDEO CODING
2.1 Video Coding Standards
Currently, state of the art monoscopic video codec is H.264/AVC [4]. MVC extension of H.264/AVC (MPEG-4 Part 10, Amendment 4) addressing 3D video applications is scheduled for early 2009 [5]. MVC exploits the similarities between multiple-camera video captures of a scene and achieves a reduction in bit rate of approximately 20% on average, when compared with coding each camera separately (simulcast coding). MVC is based on H.264/AVC High Profile.

Mobile devices have smaller displays and the current prototypes mostly use QVGA resolution. Previous MVC experiments were also performed on Multi-View Video sequences with multiple cameras. When the number of cameras is only two, coding efficiency decreases. Also using MVC requires larger Decoder Picture Buffer (DPB), causing problems with mobile devices.

Besides deciding on the video codec to use, there is also an issue of selecting profile and coding tools that are going to be used. Selecting a profile changes both coding efficiency and decoding complexity. Higher profiles increase coding efficiency with the expense of decoding complexity. Such prototypes tend to use baseline/simple profiles of the video encoders due to limited processing power.

2.2 Video Coding Tools and Properties
In this section, tools and properties of this codec that is related to this work are presented. More information can be found in [10].

H.264/AVC has several profiles to suit the needs of different applications: Baseline Profile (BP), Main Profile (MP), Extended Profile (XP), High Profiles (HiP). In mobile applications, mostly BP is used.

978-1-4244-4318-5/09/$25.00 ©2009 IEEE

There are 3 picture types in H.264/AVC. I-pictures are encoded without the use of motion compensation, thus they are independently decoded. P-pictures are predicted using only the previously decoded frames. B-pictures are bi-directionally predicted (both from past and future frames). B-pictures are not supported in BP.

Video frames are encoded with Group-of-Pictures (GoP). Each GoP starts with I frame and followed by B or P frames. By increasing GoP size coding efficiency increases while capability of dealing with losses decreases with having less frequent I-pictures.

Hierarchical B-pictures [11] can also be used within the syntax of H.264/AVC and achieve better coding efficiency, however the decoding complexity increases. Pictures at the GoP boundaries are encoded as I-frames and frames in between are encoded as B-frames in an hierarchical order. For example for GoP size of 8, Frame#0 and Frame#8 is encoded as I-frame. Then B-frames are encoded in the following order: Frame#4*[0,8]*, Frame#2*[0,4]*, Frame#6[4,8], Frame#1[0,2], Frame#3[2,4], Frame#5[4,6], Frame#7[6,8] *(Frame numbers in the brackets show the pictures used in motion estimation for encoded the required frame.)*

In H.264/AVC, two different entropy coding method can be used. Context-adaptive binary arithmetic coding (CABAC) is using the probabilities of syntax elements in a given context to losslessy compress syntax elements. Context-adaptive variable-length coding (CAVLC) is lower-complex algorithm to encode those elements. Only CAVLC is used in BP.

MVC extension of H.264/AVC is based on High Profile. Mainly it uses Hierarchical B-pictures, CABAC and disparity compensation between the frames of different cameras. Therefore, it requires more pictures in DPB and also requires more buffering before the pictures can be given to display in actual display order. Although general MVC requires a complex prediction structure, in [12] a simplified prediction scheme is proposed without significant loss of coding efficiency. In simplified prediction scheme, right view pictures can only be predicted from from right view pictures and I frames of left view.

Tested configurations in the experiments are given below and in Table 1.

IPP: Left and right videos are encoded separately using H.264/AVC with baseline profile settings and pictures are encoded as I-frame followed by P-frames for each GoP. No B-frames are used. Entropy coder is CAVLC.

IPP+CABAC: Similar to IPP with additional CABAC entropy coding instead.

IBP: Left and right videos are encoded separately using H.264/AVC with main profile and pictures are encoded as I-frame followed by P- and B-frames. Entropy coder is CABAC.

Hier: Left and right videos are encoded separately using H.264/AVC with main profile and Hierarchical-B pictures.

IPP-Stereo: Left and right videos are interleaved into a single sequence and encoded using IPP settings.

IPP+CABAC-Stereo: Left and right videos interleaved into a single sequence and encoded using IPP+CABAC settings.

MVC-Simp: Left and right videos are encoded using MVC Extension (High Profie, Hierarchical B-pictures, CABAC and Disparity Compensation). Right view pictures can only be predicted from right view pictures and I frames of left view.

MVC-General: Similar to MVC-Simp with the prediction structure allowing right view pictures to be predicted from all left pictures.

Table 1 Tools used by different coding schemes

	B-frames	*CABAC*	*# of Reference Frames (Forward - Backward)*
IPP	No	No	1-0
IPP-CABAC	No	Yes	1-0
IBP	Yes	Yes	2-2
Hier	Yes	Yes	4-4
IPP-Stereo	No	No	2-0
IPP+CABAC-Stereo	No	Yes	2-0
MVC-Simp	Yes	Yes	6-6
MVC-General	Yes	Yes	8-8

3. EXPERIMENTAL RESULTS

3.1 Coding Efficiency Experiments

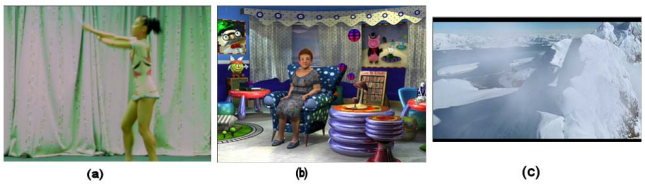

Figure 1 Stereo test sequences: (a) Rena, (b) Adile and (c) Ice.

The results are provided for stereoscopic video pairs “Rena” (Recorded by cameras with a stereo distance and provided by Tanimoto Laboratory, Nagoya University [13]), “Adile” (Computer generated animation by Momentum [14]) and “Ice” (Converted to 3D from 2D scene using [15] [Source: BBC documentation “Planet Earth”]) . Videos are first downsampled to QVGA sizes. Resolution of Rena and Adile sequences is 320x240 and resolution of Ice sequence is 320x192. Frames from the sequences can be seen in Figure 1. GoP size is selected as 8 frames. Monoscopic codec used is H.264/AVC Reference Software JM 14.2 [16] and Multi-view codec is H.264 MVC Reference Software JMVC 2.0 [17]. First 81 frames are encoded from both left and right sequences. Fixed Quantization Parameters (QP) {26, 32, 36, and 40} are used to generate rate-distortion curves. Distortion metrics are PSNR and SSIM [18] and averaged over both left and right frames.

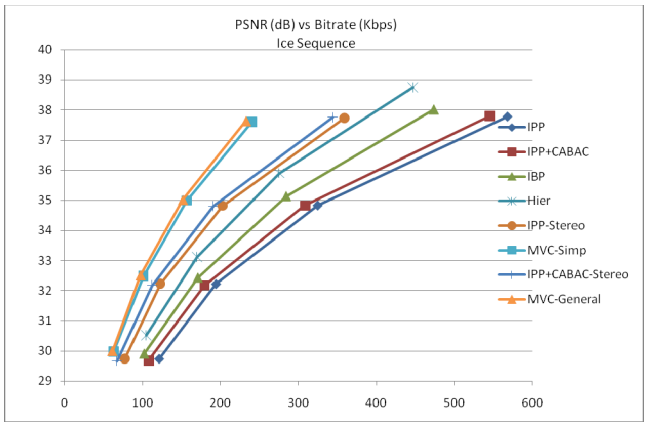

Figure 2 PSNR vs. bit rate for Ice sequence

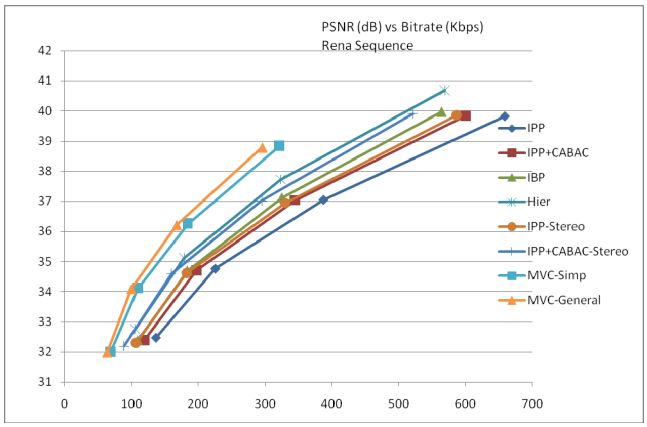

Figure 3 PSNR vs. bit rate for Rena sequence

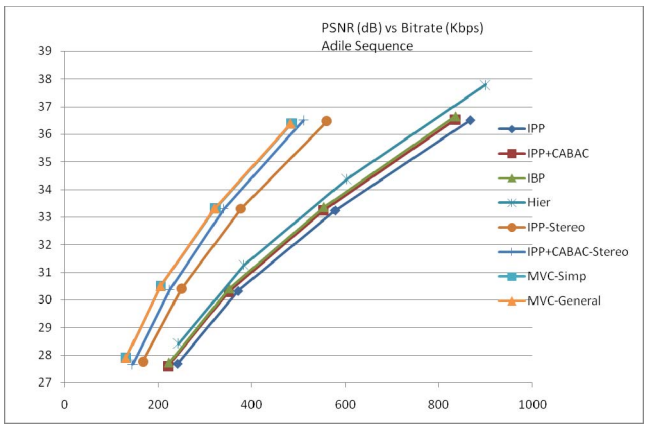

Figure 4 PSNR vs. bit rate for Adile sequence

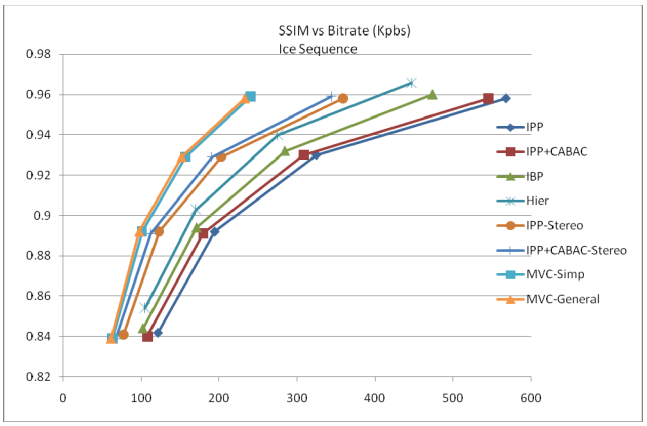

Figure 5 SSIM vs. bit rate for Ice sequence

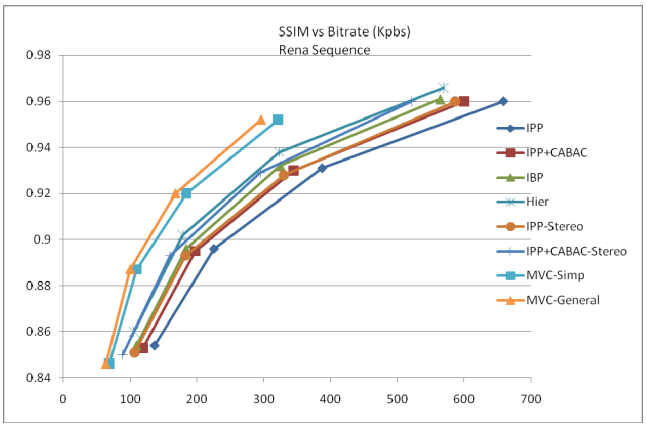

Figure 6 SSIM vs. bit rate for Rena sequence

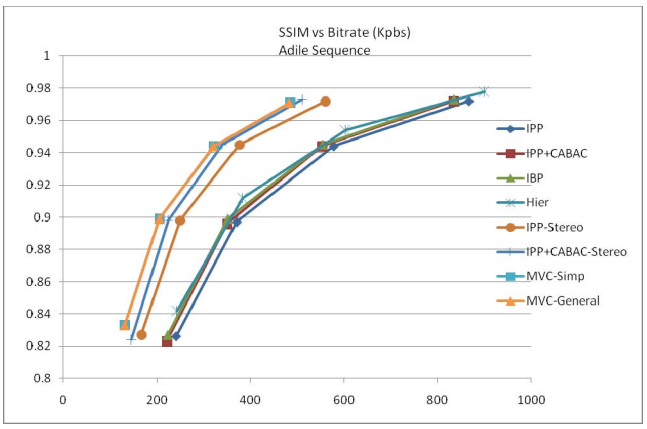

Figure 7 SSIM vs. bit rate for Adile sequence

RD-curves for each sequence are given in Figures 2-7. In all sequences difference between **MVC-Simp** and **MVC-General** is negligible. In case of MVC encoding **MVC-Simp** is preferable as stated in [12]. Similarly, MVC schemes provide a significant improvement over **IPP** and **IPP-CABAC** for all sequences. However in lower bitrates, difference between **IPP+CABAC-Stereo** and **MVC-Simp** is about 0.5-1 dB.

3.2 Decoding Complexity Experiments

In [19], H.264/AVC encoder and decoder usage is extensively studied for complexity and memory usage. It is stated that

(a) B-frames are one of the main tools that affect the access frequency and the decoding speed,

(b) Complexity increase due to CABAC is minor,

(c) Multi-reference frame usage causes a linear increase in memory peak usage.

Although the decoders in reference softwares are not optimized for speed, an analysis on decoding speed of the compressed bitstreams are given in Table 2. Since decoding is fast, decoding speeds are calculated by decoding the compressed videos (Rena sequence encoded with QP=32) 100 times and then averaging the results on a PC with 3.4 Ghz processor and 3 GB RAM.

Table 2 Decoding performance of coding schemes

Coding Scheme	*Frames per second (fps)*
IPP	68.43846
IPP-CABAC	64.76772
IBP	59.61354
Hier	57.13102
IPP-Stereo	64.87067
IPP+CABAC-Stereo	64.36886
MVC-Simp	47.14757
MVC-General	46.41807

4. CONCLUSIONS and FUTURE WORK

In this work, we analyze the possible stereoscopic encoding schemes for mobile devices. Experiments with two encoders with several configurations are carried out to figure out coding efficiency of different tools. Decoding tests also give useful information about the possible processing performance when implemented on a mobile device platform.

Depending on the processing power and memory of the mobile device the following two schemes can be used: H.264/AVC MVC extension with simplified referencing structure and H.264/AVC monoscopic codec with IPP+CABAC settings over interleaved stereoscopic content.

As a future work, video plus depth coding and asymmetric coding can also be experimented to extend the results of this work. Visual quality tests will give more information since mixed-resolution/quality sequences perceived similarly, especially in the case of asymmetric coding.

5. 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.

A. Aksay is supported in part by TUBITAK (Scientific and Technical Research Council of Turkey).

6. REFERENCES

[1] C. Fehn, P. Kauff, M. de Beeck, F. Ernst, W. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, "An Evolutionary and Optimised Approach on 3D-TV," IBC'02.

[2] http://cordis.europa.eu/fp7/ict/netmedia/projects_en.html

[3] M. Tanimoto, "Free viewpoint television-ftv," Picture Coding Symposium 2004, pp. 15–17.

[4] ITU-T Rec. H.264—ISO/IEC IS 14496-10, "Advanced video coding for generic audiovisual services," v3, 2005.

[5] A. Vetro, P. Pandit, H. Kimata, A. Smolic and Y-K. Wang "Joint Draft 8.0 on Multiview Video Coding," Joint Video Team, Doc. JVT-AB204, July 2008.

[6] K. Willner, K. Ugur, M. Salmimaa, A. Hallapuro, J. Lainema, "Mobile 3D Video Using MVC and N800 Internet Tablet", IEEE, 3DTV-CON 2008, May 2008

[7] 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.

[8] Flack, Julien; Harrold, Jonathan; Woodgate, Graham J., "A prototype 3D mobile phone equipped with a next-generation autostereoscopic display", Stereoscopic Displays and Virtual Reality Systems XIV. Proceedings of the SPIE, Volume 6490, pp. 64900M (2007).

[9] J. Kwon, M. Kim, C. Choi, "Multiview Video Service Framework for 3D Mobile Devices", Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP'08, 15-17 Aug. 2008.

[10] T. Wiegand, G.J. Sulivan, G. Bjøntegaard,, & A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 560-576.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF", Proceedings of the IEEE ICME'06, pp. 1929-1932.

[12] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient Prediction Structures for Multiview Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol.17, no.11, pp.1461-1473, Nov. 2007.

[13] "Updated Call for Proposals on Multi-View Video Coding " ISO/IEC JTC1/SC29/WG11 Doc. N7567, Nice, France, Oct. 2005.

[14] 3DTV Network of Excellence, "Public software and data repository," Nov. 2005. [Online]. Available: https://www.3dtvresearch.org/publicSwLibrary.php.

[15] S. Knorr, T. Sikora,"An Image-Based Rendering (IBR) Approach for Realistic Stereo View Synthesis of TV Broadcast Based on Structure from Motion," IEEE ICIP'07. Oct 2007.

[16] H.264/AVC Reference Software JM 14.2 http://iphome.hhi.de/suehring/tml/download/jm14.2.zip

[17] P. Pandit, A. Vetro, and Y. Chen, "WD 2 Reference software for MVC," ITU-T JVT-AB207, July 2008.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[19] S. Saponara, C. Blanch, K. Denolf, J. Bormans, "Data Transfer and Storage Complexity Analysis of the AVC/JVT Codec on a Tool-by-Tool Basis", JVT-D138, July 2002.