14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

AN H.264-BASED VIDEO ENCODING SCHEME FOR 3D TV

M. T. Pourazad, P. Nasiopoulos, and R. K. Ward

Electrical and Computer Engineering Department, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
phone: +1 (604) 822-4988, fax: +1 (604) 822-9013, email: {pourazad, panos, rababw}@ece.ubc.ca

ABSTRACT

This paper presents an H.264-based scheme for compressing 3D content captured by 3D depth range cameras. Existing MPEG-2 based schemes take advantage of the correlation between the 2D video sequence and its corresponding depth map sequence, and use the 2D motion vectors (MV) for the depth video sequence as well. This improves the speed of encoding the depth map sequence, but it results in an increase in the bitrate or a drop in the quality of the reconstructed 3D video. This is found to be due to the MVs of the 2D video sequence not being the best choice for encoding some parts of the depth map sequence containing sharp edges or corresponding to distant objects. To solve this problem, we propose an H.264-based method which re-estimates the MVs and re-selects the appropriate modes for these regions. Experimental results show that the proposed method enhances the quality of the encoded depth map sequence by an average of 1.77 dB. Finding the MVs of the sharp edge-included regions of the depth map sequence amounts to 30.64% of the computational effort needed to calculate MVs for the whole depth map sequence.

1. INTRODUCTION

Stereoscopic or three-dimensional television (3D TV) can spectacularly enhance the viewer's experience by allowing the images to emerge and penetrate from the screen into the spectator's space. It generates a compelling sense of physical space and makes the viewer feel as being part of the scene. To make a full 3D TV application available to the mass consumer market, researchers put much effort in recent years. In terms of 3D content generation and 3D display system production, significant improvements have been achieved. For 3D TV transmission, however, no compression standard has yet been agreed upon.

Recent investigations in this area are mostly focused on the efficient compression of 3D content captured by 3D depth range cameras rather than 3D video recorded via the dual-camera configuration [1]. 3D depth range cameras capture 3D content as two video sequences: a conventional two-dimensional (2D) RGB view and its accompanying depth map [2]. This format allows easy capturing, simplifies post-production, and requires lower transmission bandwidth compared to the dual-camera configuration. The latter captures stereo pair data from two slightly different perspectives, one for the left eye and the other for the right eye. To perceive the 3D content captured by a 3D depth range camera, the left- and right-eye views must be reconstructed at the receiver end. This is achieved using image-based rendering techniques [3]. Therefore at the receiver end, the viewer has the choice of watching the content either in 2D or 3D format.

The generated 3D content needs to be compressed and transmitted for consumer use. Experiments on 3D video compression for 3D TV applications show that transmitting the depth information needs about 20% of the required bitrate for MPEG-2 compressed 2D video (this is at a typical broadcast bitrate of 3 Mbit/s) [4].

One method used to compress 3D video takes advantage of the existing relationship between the 2D video sequence and the depth map sequence, and uses the motion vectors (MVs), obtained for the 2D video sequence to encode the depth map sequence as well [1]. This compression scheme was based on the MPEG-2 standard and resulted in improvement in the encoding speed. The bitrate of the depth map was fixed at 20% of the 2D video bitrate. The results showed that this approach did not hamper the quality of the encoded depth map sequence in the case of bi-directional temporal prediction when compared to the case of the depth map sequence being encoded separately. Though, for the unidirectional temporal prediction case the quality of the reconstructed depth map decreases.

However, MPEG-2 seems to yield adequate results by copying the MVs from 2D to the depth sequence, but that is true only because this video standard does not take advantage of the differences in the "texture" structure between the two streams. For instance, due to the resolution limitation of 3D depth cameras, some edges in the depth map sequence are sharper than the corresponding counterparts in the 2D video sequence. These areas may be compressed much more efficiently if motion estimation was more accurate that the one supported by MPEG-2. In addition, the depth map of all distant objects in the depth sequence has many zero valued pixels, a fact that may be exploited to reduce the number of macroblocks (MB) encoded compared to the 2D sequence. In other words, the MVs and MB modes used for the 2D sequence are not the best choices for encoding the edges and regions of distant objects in the depth map sequence.

We developed a new coding method for 3D video streams which is based on the H.264/AVC standard. H.264/AVC is the most advanced video coding standard available today, reaching approximately 50% bitrate saving when compared to previous standards like MPEG-4 and MPEG-2 [5]. Our method uses special features of the H.264/AVC standard,

such as the variable block size and the block skipping mode, addressing the above mentioned shortcomings of MPEG-2, and leading to more accurate and efficient encoding of 3D video. The implementation of our scheme is simple, taking advantage of the Network Abstract Layer (NAL) structure supported by H.264/AVC.

The rest of this paper is organized as follows: Section 2 provides an overview on some special features of H.264/AVC which are relevant to our scheme. In Section 3, the proposed video compression scheme is explained in details. Section 4 presents the discussion on experimental results followed by Section 5 that concludes on this paper.

2. OVERVIEW OF H.264/AVC'S SPECIAL FEATURES RELEVANT TO OUR SCHEME

2.1 Variable block size

Unlike MPEG-2, which uses fix macroblock size of 16x16, H.264/AVC allows variable block size for motion estimation/compensation. It partitions the frames into 16x16, 16x8, 8x16, 8x8 macroblocks. Additionally the 8x8 macroblocks are split into 8x4, 4x8, or 4x4 sub-macroblocks [6]. This feature significantly improves the accuracy of the motion estimation process and subsequently the compression efficiency. Smaller block sizes are able to predict the video contents better than larger ones especially in the regions with high complexity, high level of motion and sharp boundaries.

2.2 Block skipping mode

If a macroblock has motion characteristics that allow its motion to be effectively predicted from the motion of neighbouring macroblocks, and it contains no non-zero quantized transform coefficients, then it is flagged as skipped [6]. This mode is identified as the skip mode. Note that, when a block is skipped neither the transformed coefficients nor motion data are transmitted. In contrast to prior standards, non-zero motion vectors can be inferred when using the skip mode in P slices. Actually, in earlier standards, for a skipped MB in a P-slice, no explicit motion vectors and no coded prediction error are sent. Thus, a co-located macroblock from previous reference picture is simply copied to reconstruct the current skipped MB. This has a detrimental effect when coding video containing global motion because there is still substantial MV overhead. H.264/AVC addresses this by using a 16x16 block prediction motion vector to copy a motion-compensated block rather than assume zero motion for such a block (and simply copy a co-located block). Further in H.264/AVC, a skipped MB in a B-slice is defined as having no coded prediction error but uses ''direct'' mode motion vectors of 16x16 or 4, 8x8 blocks, depending on the coding of the co-located MB for motion compensated prediction.

2.3 Network abstract layer

H.264/AVC is the only video standard that supports a video coding layer (VCL), which is designed for efficient representation of the video content, and a network abstraction layer (NAL), which formats the VCL representation of the video and provides header information in a way that is appropriate for conveyance by different transport layers or storage media [6]. This feature provides "network friendliness" to enable simple and effective customization of the use of VCL for a broad variety of systems, including 3D video transmission.

NAL units are classified into VCL NAL units that contain the data that represents the values of the samples in the video pictures, and the non-VCL NAL units that contain any associated additional information such as parameter sets (important header data that can apply to a large number of VCL NAL units) and supplemental enhancement information. The parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded representations of the values of the samples in the video pictures. Thus, a small amount of data can be used to refer to a larger amount of information without repeating that information within each VCL NAL unit. Parameter sets can be sent well ahead of the VCL NAL units that they apply to.

3. PROPOSED METHOD

Our method is based on the fact that the depth map sequence expresses the content of the same scene as does the 2D video sequence. Since the available 3D depth range cameras operate in a limited distance range (up to 10 m) [7-9], the depth information of objects in further distance is not available and the value of depth for corresponding pixels is set to zero. As has been shown in Fig. 1, this abrupt change causes many edges of the depth map sequence to be much sharper than the corresponding edges in the 2D video sequence. As a result the MB sizes selected for efficiently encoding these edges in the depth map sequence end up being smaller than the corresponding MB sizes in the 2D video sequence. Also, due to the 3D camera distance range, the silhouette of the objects outside the camera's range is not included in the depth map sequence and consequently in this sequence, no motion is detected for these objects. In the case of MPEG-2, since the motion estimation (ME) process uses only 16x16 size macroblocks, this effect is not present. However, in the case of H.264/AVC, the ME process is much more accurate and simple, so copying of the 2D MVs is extremely inefficient. These two cases imply that the selected MB modes and the estimated MVs of the 2D video sequence are not the best choice for encoding these areas in the depth map sequence when H.264/AVC is used. Therefore special provisions should be taken into account while encoding these regions in the depth map sequence. The following subsections elaborate on the proposed approaches.

3.1 Encoding sharp edges of depth map sequence

To accurately encode the above mentioned sharp edges in the depth map sequence, the proposed method makes an effective use of the variable block size feature of H.264/AVC. To do so these edges should be detected first.

The points of an image in which the derivative of the intensity crosses a predefined threshold, are marked as edge points. The Sobel operator is a spatial domain edge detector

that uses the first order derivative to calculate the gradient of an image.

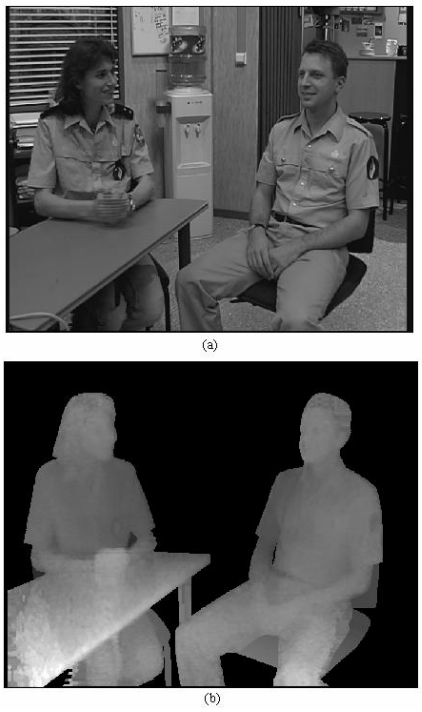

Figure 1 - Two-dimensional RGB view (a) and its accompanying depth map (b)

Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to estimate the gradient in the x-direction (columns) and y-direction (rows) at each point of image. The Sobel masks are shown in Fig. 2:

-1	0	1
-2	0	2
-1	0	1

Horizontal Operator

-1	-2	-1
0	0	0
1	2	1

Vertical Operator

Figure 2 - Horizontal and vertical Sobel operator

The magnitude of the gradient of an image at position (x, y), $|G|$, is defined in terms of the partial horizontal and vertical derivatives, G_x and G_y, as follows [10]:

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad (1)$$

If the gradient of every pixel is compared to a selected threshold and the pixels whose gradients are below the threshold value are set to 0, the edges of image will be highlighted. As can be observed in Fig. 3a, the sharp edges in the depth map sequence applying the Sobel operator with the threshold of 8 are successfully detected. Throughout the encoding procedure of each frame in the depth map sequence, if the gradient magnitude of more than one pixel was greater than the selected threshold, the whole MB is marked as edge-included MB. Edge-included MBs are encoded separately, i.e., they do not use the MVs of the 2D video sequence. This is shown to enhance the quality of the encoded depth map sequence, at the cost of increasing effort. This is because the motion vectors are re-estimated for edge-included MBs. Fig. 3b shows the edge-included MBs of Fig. 3a.

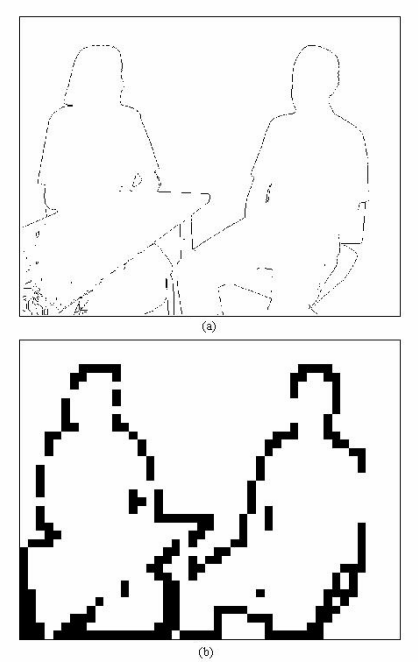

Figure 3 - Depth map sequence after edge detecting procedure with the threshold level of 8 (a) and the 16x16 MBs that are detected to contain sharp edge

3.2 Encoding distant object corresponded regions

In order to efficiently encode the regions of the depth map sequences that correspond to objects outside the camera's range, the proposed scheme takes advantage of another useful feature of the H.264/AVC standard, the skip mode (please refer to subsection 2.2 for details). The range limitation of 3D depth range cameras, creates large "background" areas with very little or not at all information (black areas in the depth sequence). The skip MB mode is ideal for compressing such areas, and thus it would be very inefficient to simply use the corresponding MB modes and MVs of the 2D sequence. In other words, despite the fact that the values of the 2D MVs for these areas may be zero or equal to the predicted MV (i.e., a prerequisite for the MB to be skipped), the corresponding 2D macroblock may contain non-zero transform coefficients and thus it is not going to be skipped. In our scheme, we identify the MBs in the depth sequence with zero transform coefficients and if the 2D MV values are zero or equal to the predicted MV values, we change the MB mode to skip mode. Therefore, the skip mode selection is done separately for the depth map sequence, improving the overall bitrate achieved by our method.

The proposed scheme in this paper can be summarized in a flowchart, which is shown in Fig. 4.

3.3 Implementation using the NAL structure of H.264/AVC

This feature of H.264/AVC is beneficial in our case, because the depth map sequence can be encoded based on the parameter sets of the 2D video sequence and the VCL NAL of the depth sequence can be sent right after sending 2D video VCL NAL. Therefore there is no need for repeating the sequence and the picture parameter sets. In addition, the VCL NAL of 2D which relates to MVs and MB modes can be shared with the depth map sequence. As a result this information is encoded and sent only once for both sequences.

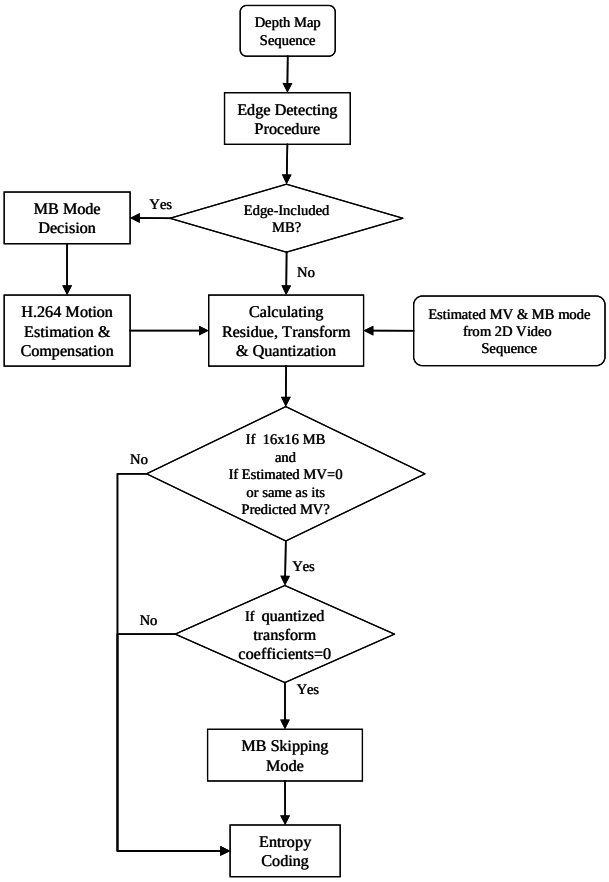

Figure 4 - Flowchart of the proposed fast encoding scheme

4. EXPERIMENTAL RESULTS AND DISCUSSION

The T264 reference software [11], which is based on the H.264/AVC standard (Baseline profile), is modified based on the proposed scheme and its performance is tested on two 3D test sequences, *Interview* and *Orbi*. In order to evaluate the encoding speed and also to choose the best threshold value for the edge detecting procedure, the depth map sequence is first encoded using the original H.264/AVC reference software. A fixed bitrate corresponding to a high level PSNR of 45dB was used. Having the bitrate fixed at the same rate, the depth map sequence is again encoded based on "Shared MV" method, i.e., using the MVs of the 2D video sequence for the depth sequence as well. The depth map sequence is again encoded using the proposed scheme at different threshold levels. The best threshold level is the one which yields the best trade off between the picture quality and the motion estimation computational cost (in terms of the average number of search points per frame). The experiments show that the best threshold value is 8. For this experiment, a typical broadcast encoder setting is considered, i.e., a GOP (Group of Pictures) length equal to 12 with GOP structures of IBBPBBP and IPPPPP. As can be observed from Table 1, applying the proposed encoding approach improves the PSNR value on average by about 1.77 dB, when compared to the "Shared MV" method. Our method however, requires finding the MVs of the sharp edge-included regions of the depth map sequence. This amounts to 30.64% of the computational effort needed to calculate MVs for the whole depth map sequence.

Test Sequence	Interview Gop structue: IPPPPPP		Interview Gop structure: IBBPBBP	
Method	Average PSNR (dB)	Average no. of search points /frame	Average PSNR (dB)	Average no. of search points /frame
Direct	46.01	91845	44.17	159141
Shared MV	41.17	0	40.23	0
Modified Shared MV	43.17	28357	43.13	53140

Test Sequence	Orbi Gop structue: IPPPPPP		Orbi Gop structure: IBBPBBP	
Method	Average PSNR (dB)	Average no. of search points /frame	Average PSNR (dB)	Average no. of search points /frame
Direct	44.60	190447	48.00	250127
Shared MV	41.57	0	43.54	0
Modified Shared MV	42.62	52841	44.68	76445

Table 1 - Average PSNR value of encoded depth map sequence using existing H.264/AVC standard (direct method), shared MV method and the proposed method (modified shared MV method) and also the average number of search points per frame.

5. CONCLUSION

We presented a new video compression scheme which is based on H.264/AVC and is designed for compressing 3D content captured by 3D depth range cameras. The existing correlation between 2D video and the corresponding depth map is exploited in encoding the depth map sequence using the MB modes and estimated MVs of the 2D video sequence. However, due to the limitations of 3D depth range cameras and the unique “texture” structure of the depth sequence, special provisions are here suggested for dealing with sharp edges in the depth map sequence and for encoding the distant objects. Our experimental results show that the PSNR value is improved by about 1.77 dB on average compared to the case where the same MVs and MB modes are used for the 2D and depth sequences. Our method, however, requires finding the MVs of the regions of depth map with sharp edges. This amounts to 30.64% of the computational effort is required to calculate MVs for the whole depth map sequence.

6. REFERENCES

[1] S. Grewatsch, and E. Miiller, "Sharing of motion vectors in 3D video coding," in *Proc. ICIP 2004*, vol. 5, pp. 3271–74.

[2] L. M. J. Meesters, W. A. Ijsselsteijn, and P. J. H. Seuntiens, "A survey of perceptual evaluations and requirements of three-dimensional TV," *Circuits and Systems for Video Technology*, vol. 14, Issue 3, pp. 381–91, Mar. 2004.

[3] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. van Gool, E. Ofek, and I. Sexton, "An evolutionary and optimized approach on 3D-TV," in *Proc. IBC* 2002, pp. 357–65.

[4] C. Fehn, "A 3D-TV system based on video plus depth information," *Signals, Systems and Computers*, vol. 2, pp. 1529–33, Nov. 2003.

[5] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *Circuits and Systems Magazine*, IEEE, vol. 4, Issue: 1, pp. 7–28, 2004.

[6] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14\ 496-10 AVC," *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050*, 2003.

[7] G. Iddan and G. Yahav, "3D imaging in the studio (and elsewhere…)," in *Proc. SPIE* 2001, vol. 4298, pp. 48–55.

[8] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav, "Depth keying," in *Proc. SPIE*, 2003, vol. 5006, pp. 554–63.

[9] 3DV Systems [Online]. Available: http://www.3dvsystems.com/news/press_releases.html

[10] H. R. Myler, A. R. Weeks, *The Pocket Handbook of Image Processing Algorithms in C*. Englewood Cliffs, New Jersey: PTR Prentice Hall, 1993.

[11] Available: http://sourceforge.net/project/showfiles.php?group_id=119399