Compression Methodology for 3D Video

Nirav Patel[1], Parth Desai[2], Prof. Manoj Patel[3]
[1, 2, 3] Department of CSE
[1, 2, 3] Alpha college of Eng & Tech, Gujarat technological University, Ahmedabad, India

Abstract- *The focus of the paper is the study of the 3D Video Compression Techniques and the Codec standards used for compression with the aim to achieve the compression of the 3D video to a more extent by merging the depth map into multiple views. This paper deals with the study of H.264/AVC standard and its encoding phase. Three-dimensional (3D) video and imaging technologies is an emerging trend in the development of digital video systems, as we presently witness the appearance of 3D displays, coding systems, and 3D camera setups. We extend the standard H.264/AVC video compression algorithm for handling the compression of the 3D video. This paper contains the information of how more compression is achieved. So to get single bit stream having information of both color and depth, we implement the concept of hiding depth map along-with the views (steganography). In this method single bitstream is generated, in which depth information is to be hidden in the information of color encode bits.*

Keywords- MVC; 3DV; FTV;FVV;H.264;MVC

I. INTRODUCTION

As successful result of 3D movies and 3D television, the Television industry seeing a major move from 2D to 3D display. Because of the main part of stereoscopic and auto-stereoscopic technologies, we can see that within few of years present display will replaced by 3D display. While changing form 2D to 3D display, 3D display user are not accept law video or image quality. Currently for 3D video display stereoscopic direct view technology is used. This display uses stereo video coding technology to display image by two camera inputs. Multiple users required to wear 3d glasses to see the result video produce by two cameras at the receiver side. Still this type of 3D video transmission is in beginning phase. In near few of year auto-stereoscopic technology produce 3D video for which multiple user not required to wear 3D glasses to see 3D video. The depth is attached with each view to produce 3D effect. Here required multi view technology of stereo video coding is required to represent huge amount of views. so bit rate is highly depends on the count of views. Because of high band width requirement, this kind of transmission is not available. Compression, decompression and rendering are the main issue while using multi view video. Here we put more importance on compression issue of multi view video and depth. So we suggest 3D video compression method where we hide depth detail by stegnography point of view.

In recent times, the 3D video is taking over the Standard 2D video and has become an increasingly common feature of our TVs, video cameras, home theatre system, and gaming consoles. 3DTV is commonly understood as a type of visual media that provides depth perception of the observed scenery and is also referred to as stereo video. Due to the rapid growth of 3D TV and 3D consumer electronic products, the issue of compressed stereo images for easier transmission and storage is attracting much attention. 3DTV is commonly understood as a type of visual media that provides depth perception of the observed scenery and is also referred to as stereo video. Initiated by the recent popularity of 3DTV, extensive activities for developing new technologies and standards for the complete processing chain can be observed, including production, representation, compression, storage, transmission, and display. This paper presents a study on video plus depth compression by modifying the available MPEG standards and its optimization for 3D services. The standard to be used for 3D Video Compression is H.264.Because of good result of stereoscopic 3D, there is more interest for three dimensional experience of TV watching. It can be accomplish by rendering huge amount of views on 3D display [1].After 3DTV, new concepts of FTV (Free View point Television) come, which enable user to choose any view point of 3D display[2]. The main problem in video coding is generating a proper method for encoding and decoding different views with their related depth. In first cycle MPEG had concentrated on development of multi video coding method and multi video transport, which is standardized as H.264/MVC [3].MVC provides a compact representation for multiple views of a video scene, such as multiple synchronized video cameras. Design of algorithm and extension of H.264/MPEG-4 AVC towards multi video coding is discussed in [3]. Here the bandwidth is linearly proportional to the count of view which is the main issue of multi vide coding (MVC) in respect to Free view point Television (FTV) [4].Decoupling of views which are transmitted and actual views rendering at customer side is main need. In March 2011 cpf (Call for Proposal) [5] for 3D video has been issued as 2nd phase. To increase compression efficiency of combination of multiple views and related depth, many researcher had put their innovative efforts. Reference[5] explores that NTT corporation get good result to utilize view synthesis prediction, with different good methods like decoder side motion vector derivation, adaptive filtering to improve on VPS[7], [8]. HHI had researched on different aspect of FTV[9],[10],[11]. Reference [10], [11] shows methods for encode multiple depth map or video. Disney Research Zurich shows that how to generate multiple views without depth maps[12].There are not any solution above which are focusing of multiple texture and depth together in single bit stream. NTT Corporation [6] has investigated conceivable outcomes for FTV continuously by CPU and GPU compacting both various composition and profundity by H.264. Reference [13] of NTT Corporations had spoken to strategies to encode composition and profundity utilizing view blend which is keep running at encoder side. Specialized University of Berlin Presented [14] two strategies for encode both numerous composition and profundity. one is same as [6] and the second one uses the assistant picture linguistic structure of the H.264 for encoding profundity map. Our methodology is utilize this idea and apply a better approach to combine the encoded profundity maps in the single piece stream of surfaces diminishing the general data transfer capacity without trade off much on the quality.

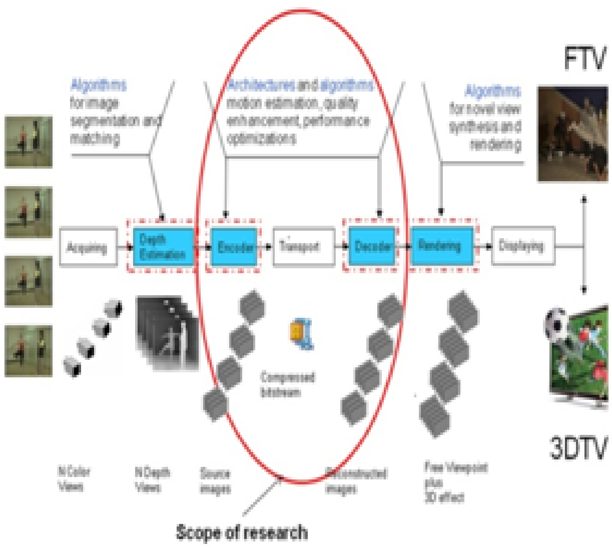

Figure 1. Free viewpoint TV structure

transmission of multi views with its related depth information at minimum bandwidth.

Rest of the paper is sorted out as clarification of the proposed plan of encode and decode of different texture frame and depth frames maps in section II, trailed by test results in section III. section IV condenses the discoveries of our investigations alongside the future extent of work in this heading.

II. PROPOSED METHODOLOGY

The video output which is generated using H.264/AVC standard can still be further compressed. Our objective is to have a single compressed bit stream of multiple views along-with depth maps which consumes lesser channel bandwidth. A depth map is simply a grayscale image which represents the distance between a pixel and its left counterpart using a grayscale value between black and white. So to get single bit stream having information of both color and depth, we implement the concept of hiding depth map along-with the views (steganography). In this method single bitstream is generated, in which depth information is to be hidden in the information of color encode bits. In order to achieve more compression, the depth bitstream generated can be merged into the color information using steganography. Here, the merging should be implemented before applying Entropy coding to the color information. Then the merged bits should be send to the Entropy coding block and the final bitstream is generated.

For each frame, 4x4 or 16 x 16 macroblocks are extracted, transformed, quantized and encoded using run-length coding obtaining a sequence as follows.

To carry out merging, each least significant bit (LSB) of the last level of the sequence of color information is replaced with a single message bit of the depth information. In this way data is hidden in noise of image.

A. Encoding Phase

Before merging the depth bits into texture, first the depth is to be encoded. The idea is that, we need to have depth encoded bit stream for each frame so that we can merge this bitstream into the texture encoded bits for the respective frame. Figure 2 shows the block diagram of the encoder where merging is implemented for view 0 (V0) and associated depth map i.e. depth 0 (D0). First, each frame of depth is predicted, transformed, quantized and encoded to form the bitstream of depth. Now, for each texture frame, 4x4 blocks are transformed and quantized and the coefficients are generated. These coefficients are then reordered in zigzag scan fashion and the last significant coefficient is found. Then, one bit of encoded depth is merged with the LSB of the last significant coefficient. Rest of the process is as usual as that of H.264/AVC and is similarly implemented for other enhanced views and associated depth maps too.In brief, the MERGING block is only addition in the encoding process which is explained in next section.

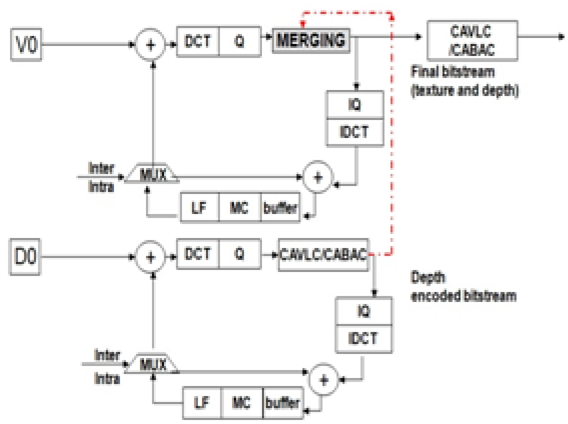

Figure 2. H.264/MVC Encoder for single view and related depth map

B. Merging Phase

The converging of the encoded depth bits into the coefficients of composition is appeared in Figure 3. The depth bitstream is implanted in composition bitstream by supplanting the LSB of the quantized last huge coefficient with a touch of depth bitstream. Each least significant bit (LSB) of the last level of the arrangement is supplanted with a mono message bit. Along these lines information is covered up in commotion of picture. To include the depth bitstream to a composition bitstream, the DCT coefficients in every piece are tried by examine request, till we get the last non-zero coefficient. In the event that a last non-zero coefficient is found, and the LSB of its level is unequal to the profundity bit Li (I = 0,1,2,, l), this least significant bit is supplanted by the depth bit Li . What's more, if the LSB of its level breaks even with the depth bit Li the LSB is not changed. The strategy is rehashed until all depth bits are combined/embedded. The above strategy is rehashed for each piece until all bits of the encoded depth map, from now on called the message, are embedded into the composition bitstream. Along these lines a depth bit is covered up in composition bitstream. The insertion of the depth bits in composition is not visible of the fact that human eye can't recognize immaterial change of color.

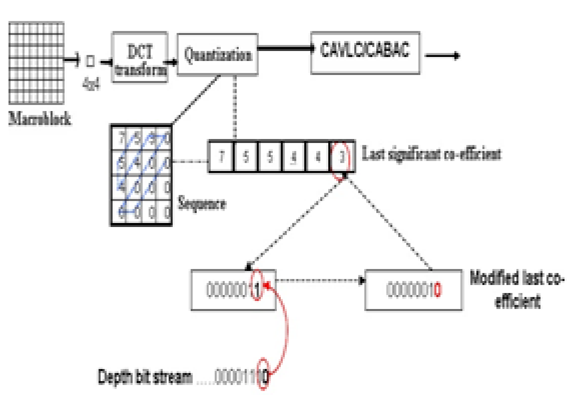

Figure 3. Combination of a depth bit

Likewise, the converging of depth bits is actualized in encoder for different multiple frames. Our methodology in encoder for various frame is appeared in the Figure 4. Here, frame 1 (V1) is anticipated from frame 0 (V0) and depth frame 1 (D1) is anticipated from depth frame 0 (D0). The bitstream of various perspectives are created and are given to the constructing agent to frame the last bitstream containing of depth and composition.

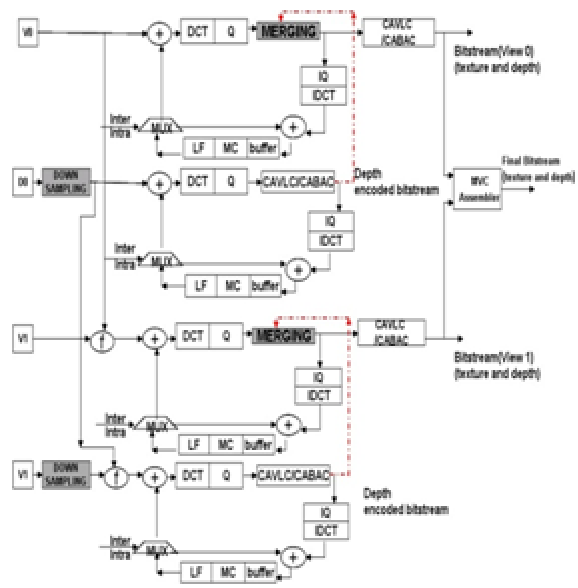

Figure 4. H.264/MVC Encoder for multi view and related depth map

C. Decoding Detail

At the decoder side, first the depth bits are extricated as appeared in fig. 5 and decoding process for profundity map proceeds as common as in H.264/MVC. For composition/color, the translating strategy is same, without extricating consolidated depth bits bringing about in reverse similarity though a "more quick witted" decoder can translate the depth maps also.

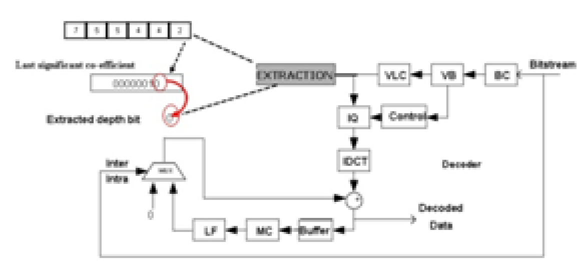

Figure 5. Updated Decoder

III. SIMULATION RESULTS AND ANALYSIS

Simulations on different sequences with 720p and 1080p resolution are carried out. The input stream contains 15 frames and two texture views with its associated depthmaps. Texture views contain luminance and chrominance components while depthmaps contain only luminance components. Different QP values for texture and depth have been used because change in a value of QP generates different number of bits in an encoded bit-stream thereby capturing all possible effect on quality. In this configuration, two views of textures alongwith two views of associated depth maps have been encoded into single bitstream consisting of 4 views which is termed as 'Anchor' hereafter.Both the configurations have been simulated for high delay applications i.e. I, P and B pictures. MSR_BreakDancers sequence is used as an input stream for the simulations. The textureencoding is done at Quantization Parameters (QPs) = 14, 18, 22 and 26 and thedepth encoding is done at QPs = 32, 36, 40 and 44. The GOP = 8, Intra Period =16 and Number of Frames = 15.

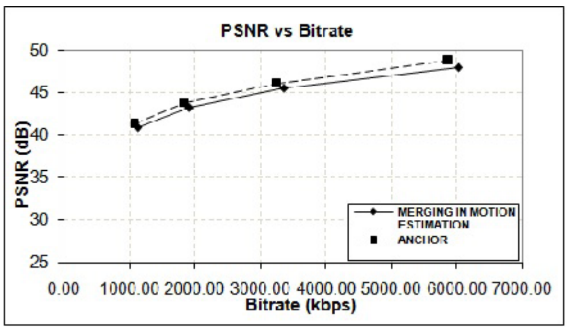

Figure 6:Merging in final bitstream for PSNR vs MSR_BreakDancers

TABLE I. STATISTICS FOR DEPTH BITS MERGING, MSR_BREKDANCER

MSR_brekDancer, GOP=8, IP=16			
QPtexture, QPdepth	Depth bits to be merged	Depth bits Merged	% Depth bits merged
14,32	163104	107933	66.17
18,36	95984	66087	68.85
22,40	55856	38445	68.83
24,34	33544	20548	61.26

Thus, when merging is implemented in motion estimation loop, the degradation in quality of texture is around 0.4dB to 0.6dB which is acceptable. Also, there is not much effect on the amount of merging. So, now onwards we shall apply the merging concept in motion estimation for the simulations.

Simulations were carried out on different HD sequences, 100 frames with merging in only 4x4 transform block. For the current simulations, 8x8 transform block is disabled. Figure 7 and 8 shows the R-D curve for MSR_Breakdancers and GT_Fly respectively. Merging statistics for the same are shown in Table II and III.

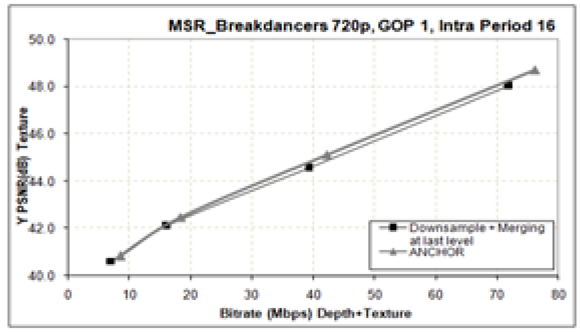

Figure 7 R-D performance for MSR_Breakdancers Low Delay (Merging in 4x4)

TABLE II. MERGING STATISTICS FOR MSR_BREKDANCER

Sequence: MSR_Breakdancers, GOP=1, Intra period=16			
QPtexture, QPdepth	Depth bits to be merged	Depth bits Merged	% Depth bits merged
14,14	14916536	6438760	43.17
18,18	10035280	3986137	39.72
22,22	6697712	1568416	23.42
26,26	4293056	591660	13.78

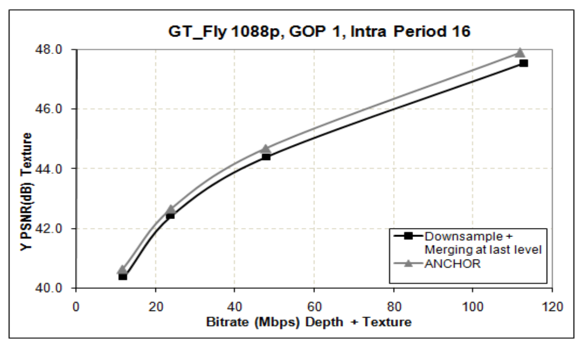

Figure 8 R-D performance for GT_Fly Low Delay (Merging in 4x4)

TABLE III. MERGING STATISTICS of GT_FLY

Sequence: GT_Fly, GOP=1, Intra period=16			
QPtexture, QPdepth	Depth bits to be merged	Depth bits Merged	% Depth bits merged
14,14	6663472	6340672	95.16
18,18	3846624	3492868	90.80
22,22	2198128	1943285	88.41
26,26	1344344	1072232	79.76

Simulations were also carried out for high delay applications which shows the same characteristics. From the above R-D curves it can be seen that the loss in PSNR of texture compared to Anchor is around 0.6dB and the bitrate is also reduced by around 4% to 6%. The amount of merging of depth bits is good when merging is applied only in 4x4 block.

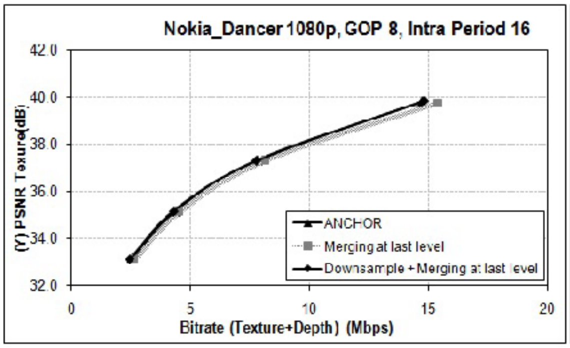

Figure 9: R-D performance for Nokia Dancer (High Delay)

Since depth maps are only used for virtual view rendering as non visual data, the perceived quality of the virtual views which are rendered using the decoded texture and depth maps is more important than the visual quality of depth map.

The reproductions have additionally been done for low delay applications (sequence containing just I and P pictures) furthermore on same QP, which demonstrated comparable qualities as appeared above.

IV. CONCLUSIONS

The powerful and adaptable coding procedure is produced which guarantees that the whole depthmap is suited in a mono bitstream which is completely in reverse good. Simulations show that there is very less effect on texture quality but depth quality is degraded due to upsampling/downsampling of depthmap. It has been derived from the subjective evaluation of synthesized views that this degradation of depth PSNR doesn’t affect much in the overall rendered quality. In this way, we firmly trust that the technique and the framework could be utilized for FVV/3DV use cases. The methodology could be particularly viable as far as encode and/or transmit any metadata incorporating depth maps alongside video information. The methodology is in reverse good as in any standard decoder can decode texture for any H.264/MVC use cases (for instance, stereo 3D by MVC Stereo High Profile).

The merging concept (using steganography) works better for natural sequences compared to synthetic sequences. Also, the main contribution for the decrease in bitrate is due to downsampling rather than merging. So, we can say that merging technique might not be advantageous in multi view and depth coding. Without applying merging, we can send the downsampled depth appended at the end of every slice and decrease the bitrate of the system with no loss in PSNR. If we consider other applications, this robust coding strategy can be useful to send some other meta data information of the video like camera parameters, etc. As merging amount is considerably decreased while enabling 8x8 mode, it can be derived that this approach might not be applied to HEVC which uses still larger transform blocks. Thus, we can conclude that this proposed approach of steganography based coding can be used as an exploration platform to send some meta data with auto stereoscopic multi view 3D video.

ACKNOWLEDGMENT

We might want to express our true appreciation towards everyone who has contributed different HD streams to MPEG for investigations towards 3D video coding.

REFERENCES

[1] Hakan Urey, Kishore V. Chellappan and Phil Surman, "State of the Art in Stereoscopic and Autostereoscopic Displays", Proceedings of the IEEE, Vol. 99,No. 4, April 2011.

[2] Masayuki Tanimoto, Mehrdad P. Tehrani, Toshiaki Fujii and Tomohiro Yendo, "Free-Viewpoint TV - A Review of Ultimate 3DTV and it's related Technologies", IEEE Signal Processing Magazine, January 2011.

[3] Anthony Vetro, Thomas Wiegand, and Gary J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/ MPEG-4 AVC Standard", Proceedings of the IEEE, Vol. 99, No. 4, April 2011.

[4] Karsten Muller, Philipp Merkle and Thomas Wiegand, "3-D Video Representation Using DepthMaps", Proceedings of the IEEE, Vol. 99, No. 4, April 2011.

[5] MPEG2011/N12036, ISO/IEC JTC1/SC29/WG11, "Call for Proposals on 3D Video Coding Technology", March 2011.

[6] Shinya Shimizu et al, NTT Corporation, "Real-Time Free-Viewpoint Viewer From Multiview Video Plus Depth Representation Coded By H.264/AVC MVC Extension", 3DTV-CON 2009.

[7] Shinya Shimizu and Hideaki Kimata, "Improved View Synthesis Prediction Using Decoder-Side Motion Derivation For Multiview Video Coding", IEEE, 2010.

[8] Shinya Shimizu, Hideaki Kimata and Y. Ohtani, "Adaptive Filtering in View Synthesis Prediction for Multiview Video Coding", NTT Corporation, Proceedings of APSIPA, Japan, October 2009.

[9] P. Merkle, Y. Morvan, A. Smolic, D. Farin, and T. Wiegand, "The Effect of Depth Compression on Multiview Rendering Quality", IEEE Proc. of 3DTV-CON'08, May 2008.

[10] Karsten Mller, A. Smolic, K. Dix, Philipp Merkle, and Thomas Wiegand, "Coding And Intermediate View Synthesis Of Multiview Video Plus Depth", ICIP 2009.

[11] Philipp Merkle, Aljoscha Smolic, Karsten Mller and Thomas Wiegand, "MultiView Video Plus Depth Representation And Coding," ICIP, 2007.

[12] Manuel Lang, "Nonlinear Disparity Mapping for Stereoscopic 3D", Disney Research Zurich, July 2010

[13] Shinya Shimizu, Hideaki Kimata, Kazuto Kamikura and Yoshiyuki Yashima, "A Backward Compatible 3D Scene Coding Using Residual Prediction", ICASSP, 2008.

[14] P. Merkle, Y. Wang, K. Mller, A. Smolic, and T. Wiegand, "Video Plus Depth Compression for Mobile 3D Services", 3DTV-CON, 2009.