DEPTH TEMPLATE BASED 2D-TO-3D VIDEO CONVERSION AND CODING SYSTEM

Zhenyu Wang[1,2], Ronggang Wang[1,2], Shengfu Dong [1,2], Wei Wu[3], Longshe Huo[4], Wen Gao[2]*

[1]Peking University Shenzhen Graduate School, [2] Peking University National Engineering Laboratory for Video Technology, [3]Skyworth Group, [4]China Unicom Research Institute
{wangzhenyu, *rgwang, sfdong}@pkusz.edu.cn, wuwei@skyworth.com, huols@chinaunicom.cn, wgao@pku.edu.cn

ABSTRACT

A Depth Template based 2D-to-3D Video Conversion and Coding system (DTVCC) is proposed by this paper. In DTVCC, triangle meshes are exploited to describe depth template for scenes in 2D video, an interactive system is designed to generate reliable depth template, depth map can be automatically reconstructed based on the depth template and 2D video frame pixels information, and the generated 3D video is compressed just by coding the 2D video plus depth template. Experiment results show that not only high quality 3D video is generated, but also the bit rate of coding the converted 3D video is saved by 12%~38% with our proposed system of DTVCC.

Index Terms—Depth template, 2D-to-3D conversion, video coding, triangle mesh

1. INTRODUCTION

With the various 3D display devices emerging in the market, stereoscopic video applications are becoming more and more popular in recent years. However, 3D content is still not enough to boost 3D video industries. Converting the huge amount of existing 2D video to 3D video is highly desired. Extracting the depth information from 2D video frame is crucial for 2D-to-3D conversion. Two typical depth cues of binocular and monocular cues are exploited by the human being to perceive the world in three dimensions. Depth information is mainly provided by binocular cues when viewing a scene with two eyes through exploitation of differences between the perceived video frames, while the depth information can still be provided when viewing a scene with one eye by monocular cues. The estimation of scene depth information aims to convert monocular depth cues contained in 2D video sequences into quantitative depth values. Monocular depth cues can be subdivided into pictorial and motion cues [1]. Various automatic methods have been proposed to estimate depth information from these monocular depth cues, such as Depth From Focus/Defocus[2], Depth From Geometric[3], Depth From Color and Intensity[4], and Depths From Motion[5] etc. Multiple cues fusion scheme has also been proposed [6].

Even though much research has been done to enable automatic 2D-to-3D conversion, the techniques are still far from mature. Most available products and methods are only successful in certain circumstances. False monocular depth cues may be estimated, and the depth ambiguity from monocular depth cues is another issue not resolved.

A more robust 2D-to-3D conversion method is to involve human interaction by utilizing the stereoscopic perception of eyes on 2D video. The manual scheme is to shift the pixels horizontally with an artistically chosen depth value for different regions/objects in the image to generate a new image [7], where hand drawing produces high quality depth, but is very time consuming and expensive. The human-assisted scheme is to convert 2D images to stereoscopic 3D with some corrections made "manually" by an operator [8]. Even though this scheme reduces the time consumed in comparison to the manual conversion scheme, a significant amount of human engagement is still required to complete the conversion.

To convert the vast collection of available 2D material into high quality 3D in an economic manner, a quasi-automatic conversion scheme is desired. In this paper, we propose a Depth Template based 2D-to-3D Video Conversion and Coding system (DTVCC). In DTVCC, an interactive system is designed to generate reliable triangle mesh based depth template, which only involves few human operations. Depth map can be automatically reconstructed based on the fusion of depth template and image segmentation information of 2D video frame. Consequently, high quality 3D video is generated by the quasi-automatic method. By product, the generated 3D video is compressed by just coding the 2D video plus depth template, and bit rate of coding the 3D video is greatly saved.

This paper is organized as follows. After introduction, Section 2 will describe the framework for our proposed 2D-to-3D conversion and coding system of DTVCC. The triangle mesh based depth template will be provided in Section 3. Section 4 is devoted to the codec design for the generated 3D video. Experiment results will be discussed in Section 5, and Section 6 concludes this paper.

2. 2D-TO-3D CONVERSION AND CODING SYSTEM

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.19

The architecture of the proposed depth template based 2D-to-3D conversion and coding system of DTVCC is showed in Fig.1. DTVCC is composed of encoder side and decoder side. At encoder side, a human interactive module is designed to generate a triangle mesh based depth template for each 2D video frame, and then the depth template is coded together with 2D video. The coded bit-stream is transmitted to decoder side. At decoder side, the 2D video and depth template are first decoded, and depth map is reconstructed by exploiting the depth template and pixels information of decoded 2D video frame; at last, the 3D video is synthesized from the decoded 2D video frame and the reconstructed depth map by DIBR method [6].

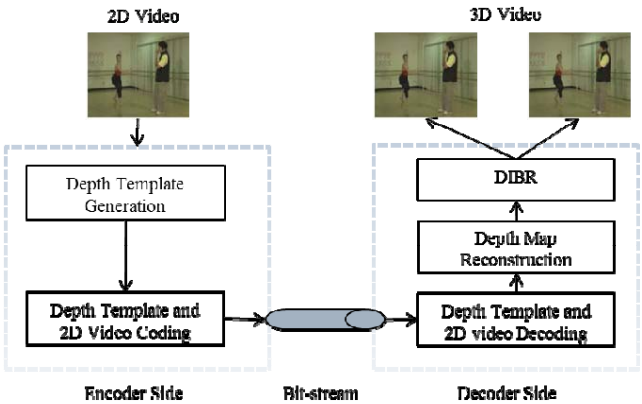


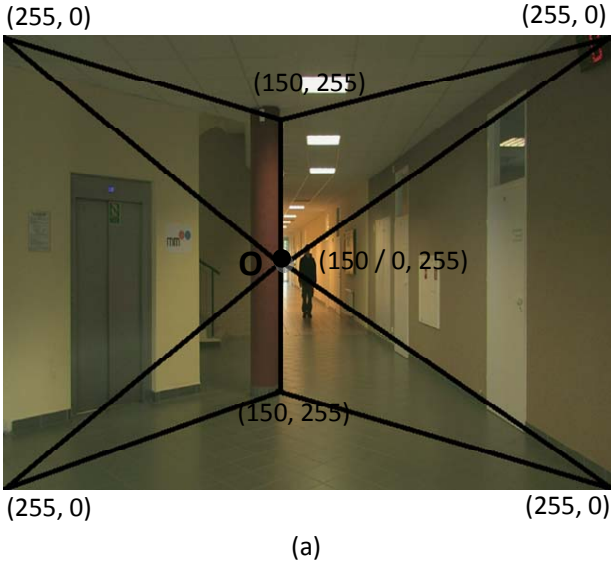
Fig.1. Architecture of our proposed system of DTVCC

3. TRIANGLE MESH OF DEPTH TEMPLATE

Considering the fact that the relative depth differences among different objects in one scene are crucial for stereoscopic experience, in DTVCC, we use a depth template to describe depth change trend within one 2D video frame. Since there may be several depth change directions (trends) in one scene, we use a triangle mesh to represent each depth change direction. For each triangle mesh, the coordinate and depth value of each vertex are recorded. We assume the depth values of pixels in the same triangle mesh are smooth, thus the coordinate and depth value of vertices in a triangle mesh can determine a depth plane. The depth value of other pixels in this triangle mesh can be interpolated by depth values of vertices, and thus is determined by the depth plane. For example, we divide 2D video frame showed in Fig.2 (a) into four meshes, a depth template can be built as shown in Fig.2 (b).

We can also divide the video frame more accurately as shown in Fig.3 (a), and build a better depth template as shown in Fig.3 (b). Take note of the vertex on the edge of sharp depth value change, like the vertex O in Fig.3 (a), more than one depth value will be recorded. We record the coordinate and depth value of the vertex within every triangle mesh one by one.

Fig.2. Triangle mesh based depth template



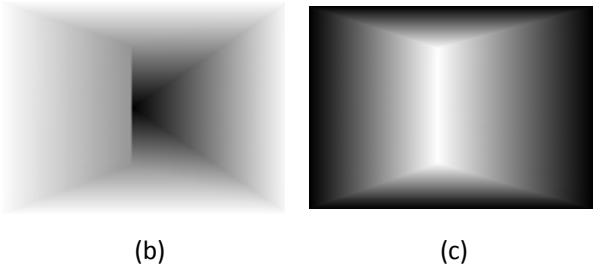


Fig.3. More accurate triangle mesh based depth template and non-smooth template

However, the depth value of objects in a scene may not be same as the one on the depth template. The color information of 2D frame can be used to segment objects. With the result of segmentation, we give each object the average depth value of all pixels in the object from the depth template.

We aim to convert and playback the 3D video from 2D video in real time, so a segmentation method with low complexity is adopted. For each pixel P not segmented, we use breadth-first traverse algorithm to check all its nearest 8 adjacent pixels. If an adjacent pixel P' of P is not segmented and the difference of color between P' and P is no more than a threshold T, we add this adjacent pixel P' to current object

O and insert it into the breadth-first traverse queue. Pseudo-code of the segment process is as follows.

```
BFTQ : breadth-first traverse queue
T: segmentation threshold
FUNCTION CheckNeighbour(x, y, val, idx, BFTQ):
    IF ( pixel[x, y] is segmented ) THEN
        RETURN;
    END IF
    val_cur = color value of pixel[x, y];
    IF ( abs(val_cur - val) < T) THEN
      expand pixel[x, y] to tail of BFTQ;
      give pixel[x, y] the object label idx;
END IF
FUNCTION Segmentation(pixel, width, height):
  FOR i=0 TO height-1 DO
      FOR j=0 TO width-1 DO
        IF ( pixel[j, i] is segmented ) THEN
          CONTINUE;
        END IF
        give pixel[j, i] a new object label (idx);
        empty BFTQ;
        expand pixel[j, i] to tail of BFTQ;
        DO
          remove header of BFTQ and get pixel[x, y];
          val = color value of pixel[x, y];
          CheckNeighbour(x-1, y-1, val, idx, BFTQ);
          CheckNeighbour(x  , y-1, val, idx, BFTQ);
          CheckNeighbour(x+1, y-1, val, idx, BFTQ);
          CheckNeighbour(x-1, y,   val, idx, BFTQ);
          CheckNeighbour(x+1, y,   val, idx, BFTQ);
          CheckNeighbour(x-1, y+1, val, idx, BFTQ);
          CheckNeighbour(x,   y+1, val, idx, BFTQ);
          CheckNeighbour(x+1, y+1, val, idx, BFTQ);
      WHILE (BFTQ is not empty);
    END FOR
END FOR
```

Fig.4 shows the depth map built from the frame in Fig.3 (a) and the depth template Fig.3(b) under the constraint of objects segmentation.

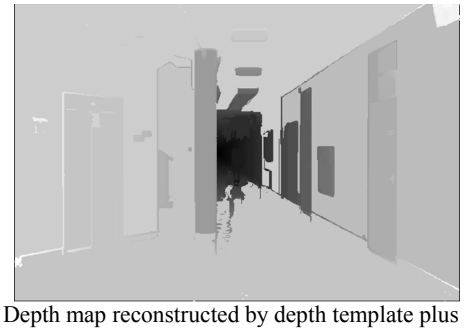


Fig.4. Depth map reconstructed by depth template plus objects segmentation

As shown in Fig.4, the segmentation method groups the pixels on floor (and the wall) as one object. The pixels on the floor are given the same depth value. However, the true depth values of floor and wall are smoothly changing. Though we can adjust the segmentation threshold T and group the floor and wall into many more small objects, the depth values of these objects stills to be discontinuous.

Fig.5. Enhanced depth map reconstructed by enhanced depth template

In fact, the depth template showed in Fig.3 (b) is more accurate than the depth map in Fig.4. But if we use the depth template as the depth map directly, the object crossing the triangle border will be cut by the border, the depth values on the border are discontinuous. If we can use the depth value on depth template directly in areas with smooth depth changing trend, and use the depth value from the result of segmentation in areas with non-smooth depth changing trend, the depth map will be more accurate.

We propose an enhanced depth template, by adding vertex's non-smooth value (NV) manually into triangle mesh. With the non-smooth value, we can build a corresponding non-smooth template like the depth template. The depth value of pixel i on depth map can be calculated by:

$$Depth_i = NV_i * VRS_i + (1 - NV_i) * VDT_i \qquad (1)$$

Where VRS_i denotes the depth value of pixel i deduced from result of segmentation, VDT_i denotes the depth value of pixel i on depth template, NV_i is weight to fusion the two depth values. Since the difference of VRS and VDT can be significant and the depth value on depth template can be discontinuous, to prevent objects to be cut by triangle border, we propose three principles of creating the enhanced depth template:

1. The non-smooth value NV is between 0 and 1;
2. The more discontinuous the depth value is, the bigger the NV value is;
3. The non-smooth value NV should be continuous, so each vertex can have only one non-smooth value;

For example, we give each vertex a depth value and a non-smooth value as shown in Fig.3 (a), we can build a non-smooth template in Fig.3(c). An enhanced depth map can be reconstructed by this method as shown in Fig.5. Some other examples are showed in Fig.6.

Now, the triangle mesh of enhanced depth template can be described with a string:

$$MeshStr = Tri_1 Tri_2 .....Tri_n \ . \qquad (2)$$

Where Tri_n can be described as:

$$Tri_n = X_{n1}Y_{n1}D_{n1}S_{n1}X_{n2}Y_{n2}D_{n2}S_{n2}X_{n3}Y_{n3}D_{n3}S_{n3} \qquad (3)$$

Where, X_{ij} , Y_{ij} , D_{ij} , S_{ij} are the X coordinate, Y coordinate, depth value and non-smooth value of the *j*-th

vertex of the *i*-th triangle area. The value of D_{ij} and S_{ij} are within 0~255. So the NV_i in (1) must be the value of non-smooth value divided by 255.

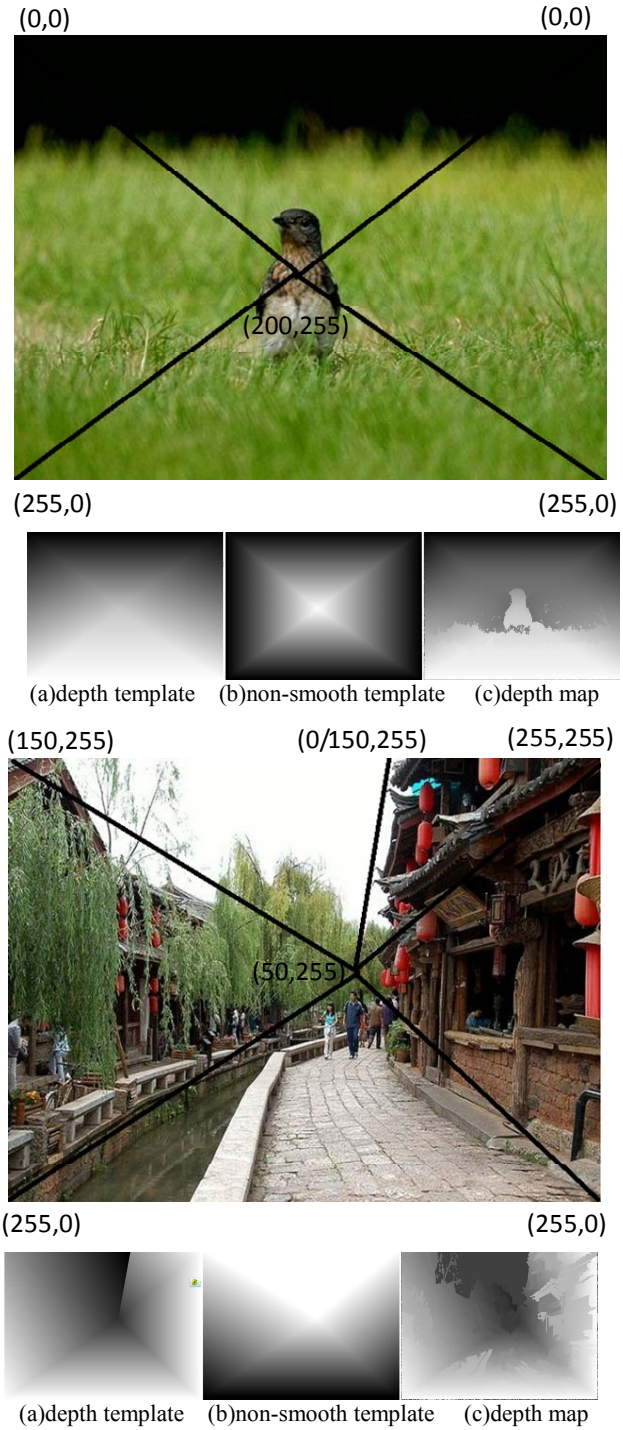


Fig.6. Depth map generated by enhanced depth template

4. 3D VIDEO CODING AND DECODING

After the string of enhanced triangle mesh description is gotten by automatic or semi-automatic method, we insert it into picture header of 2D video stream generated by video encoder. At the client, we decode the video stream first, and get the 2D Video frames and the enhanced triangle mesh descriptions. The depth template and non-smooth template will be reconstructed from the enhanced triangle mesh. Depth map can be built with 2D Video frame, depth template and non-smooth template by automatic method. As shown in Fig.7.

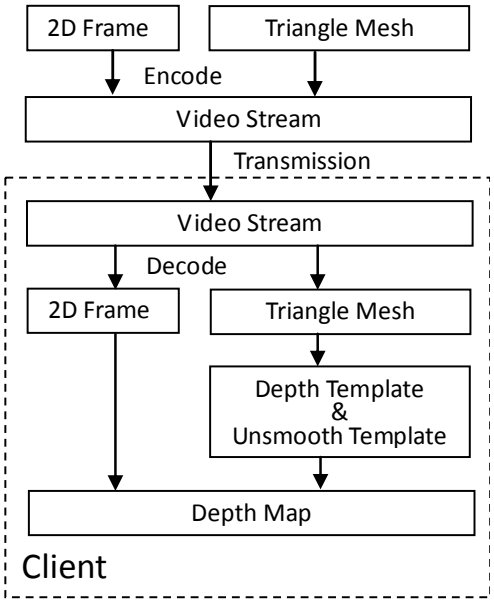


Fig.7. 3D video coding and decoding

4.1. Triangle mesh encoding

The latest video coding standard such as H.264/AVC and AVS allows user to insert enhanced information or user data before the coded data of picture. In H264/AVC, it is supplemental enhancement information (SEI) [9], and in AVS, it is extension and user data [10]. So we can store the string of triangle mesh description here.

We allocate 12 bits for X_{ij} and Y_{ij} in (3), 8 bits for D_{ij} and S_{ij}. Thus, a triangle area cost 40x3 bits. Considering the compression performance, the cost of store triangle mesh is insignificant. For a video with frame rate of 25, if there are 10 triangle meshes for each frame, the bit rate will only increase about 30 kbps. In fact, for most scenes, it is unlikely to use more than 10 triangle meshes describing the depth template. What's more, for static scenes, the depth templates of neighbor frames are nearly the same, so we only need to code the triangle meshes of the first frame. The cost of storing triangle mesh can be reduced further.

4.2. Depth template reconstruction

At decoder side, the depth template and non-smooth template of each frame can be reconstructed when decode the 2D video stream. For each pixel on depth template, we first find the triangle area which this pixel belong to. And then calculate this pixel's depth value and non-smooth value as described in Section 3.

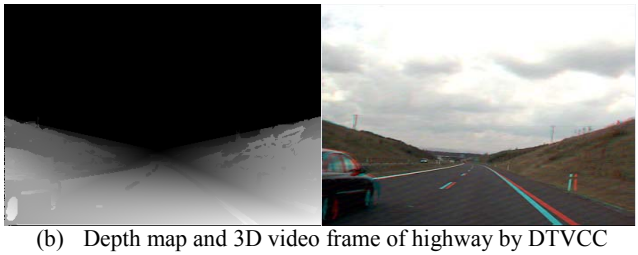
Some pixels on the border of triangle area belong to more than one triangle area, if the depth value on this pixel is discontinuous, it will have more than one depth value on depth template. We choose a value randomly. Since we require the non-smooth value to be bigger, so the depth value of this pixel on depth map is mainly from VRS.

5. EXPERIMENTS

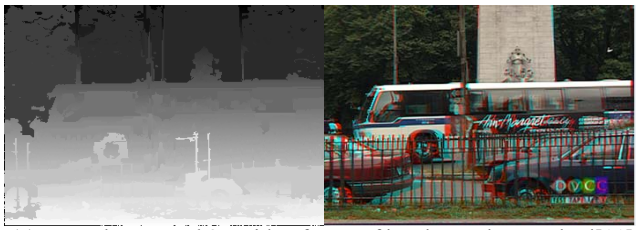
In this section, experimental results are presented to demonstrate the advantages of the proposed triangle mesh based depth template and system of DTVCC. We select sub-video sequences of 90 frames from two typical 2D CIF sequences of bus and highway as test ones. Depth template generated by DTVCC is very valuable for improving 2D-to-3D converted video quality. Fig.8 gives a comparison between the depth map and 3D video frame generated by the traditional anchor 2D-to-3D video conversion method [11] and the one reconstructed by DTVCC, and it is obvious that the reliability of depth map is greatly improved by DTVCC.

To testify the 3D coding performance improvement by DTVCC, it is compared with two representative 3D video coding methods of MVC and 2D-plus-Depth. To facilitate the comparison, we first synthesize raw "left view" video and raw "right view" video by DIBR from raw 2D video for MVC, and generate depth map by our interactive 2D-to-3D conversion module (by exploiting the raw 2D video) for 2D-plus-Depth. In MVC method, we use "Stereo High Profile" implemented in the newest H.264/AVC reference software JM18.0 to encode the generated raw "left view" video and "right view" video, and calculate the average distortion of the two views caused by compression. In 2D-plus-Depth method, we encode the raw 2D video and the depth map by H.264/AVC High Profile respectively, synthesize the "left view" video and "right view" video based on the decoded 2D video and depth map by DIBR method, and calculate the average distortion of two views caused by the above process. In DTVCC, we encode the raw 2D video and depth cues generated by our system, reconstruct the depth map based on the decoded depth template, synthesize the "left view" video and "right view" video based on the decoded 2D video and reconstructed depth map by DIBR method, and calculate the average distortion of the two views caused by the above process.

(a) Depth map and 3D video frame of highway by anchor method[11]



(b) Depth map and 3D video frame of highway by DTVCC



(c) Depth map and 3D video frame of bus by anchor method[11]

(d) Depth map and 3D video frame of bus by DTVCC

Fig.8. Depth map and 3D video quality comparison between anchor method and DTVCC

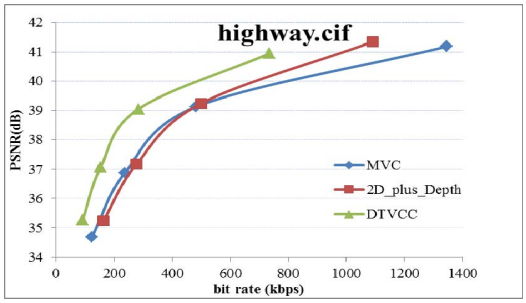
Rate distortion performances of the three methods are showed in Table 1 and Fig.10, and Table 2 lists the results in detail. Test conditions in detail are set as follows. Frame rate is set as 30 fps, GOP structure of "IPPP…" and 5 reference frames are used, search range is set as 32. Four QPs of 24, 28, 32 and 36 are used to generate different bit rates of coded video or depth map. Compared with MVC, DTVCC saves bit rate by 38.68% and 38.69% at the same level of decoded 3D video quality over highway and bus respectively (or improves the decoded 3D video quality by 1.16 dB and 2.56dB at the same bit rate). Compared with 2D-plus-Depth, DTVCC saves bit rate by 38.21% and 12% at the same level of decoded 3D video quality over highway and bus respectively (or improves the decoded 3D video quality by 1.39 dB and 0.66dB at the same bit rate).

Table 1. 3D coding performance gain by DTVCC

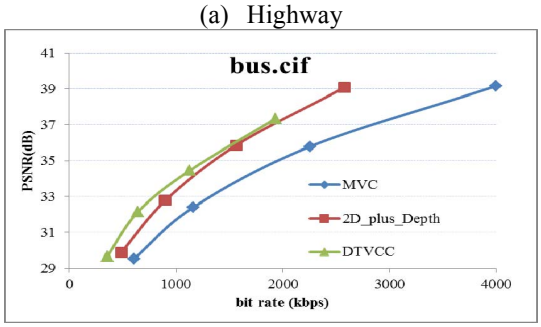
	DTVCC vs. MVC		DTVCCvs. 2D_plus_Depth	
Test sequences	Bit rate saved	Video quality improved	Bit rate saved	Video quality improved
highway	38.68%	1.16dB	38.21%	1.39dB
bus	35.69%	2.56dB	12.00%	0.66dB

Table 2. 3D coding results by the three methods

Method	Seq.		QP			
			24	28	32	36
MVC	High way	Bitrate (kbps)	1344.86	481.57	236.43	123.5
		PSNR (dB)	41.179	39.129	36.87	34.69
	Bus	Bitrate	3996.51	2255.5	1164.38	607.56
		PSNR	39.14	35.78	32.40	29.52
2D plus Depth	High way	Bitrate	1091.46	499.46	276.8	162.89
		PSNR	41.34	39.24	37.17	35.25
	Bus	Bitrate	2575.96	1564.12	900.56	488.64
		PSNR	39.07	35.86	32.78	29.89
DTVCC	High way	Bitrate	734.08	283.15	153.61	91.17
		PSNR	40.94	39.05	37.07	35.27
	Bus	Bitrate	1934.42	1127.59	642.78	356.88
		PSNR	37.35	34.45	32.15	29.67



(a) Highway



(b) Bus

Fig.10. 3D video coding performance comparion of the three methods

6. CONCLUSION

A system of DTVCC based on triangle mesh depth template is proposed for 2D-to-3D video conversion and coding. Experiment results testify that not only the quality of 3D video converted from 2D video is improved, but the bit rate of coding the generated 3D video is greatly saved by DTVCC.

ACKNOWLEDGEMENT

This work was supported by the grant of Shenzhen Fundamental Research Fund JC201104210117A, the grant of Key Projects in the National Science & Technology Pillar Program 2011BAH08B03, and National Basic Research Program of China (973 Program) 2009CB320903, 2009CB320907.

12. REFERENCES

[1] L. Zhang, C. Vazquez, and S. Knorr, "3D-TV Content Creation: Automatic 2D-to-3D Video Conversion," Broadcasting, IEEE Transactions on, vol.57, no.2, pp.372-383, June 2011.

[2] G. Guo, N. Zhang, L. Huo, and W. Gao, "2D to 3D conversion based on edge defocus and segmentation," in IEEE Int. Conf. Acoust., Speech Signal Process., pp. 2181–2184. Mar. 31–April, 4 2008.

[3] X. Huang, L.Wang, J. Huang, D. Li, and M. Zhang, "A depth extraction method based on motion and geometry for 2D to 3D conversion," in 3rd Int. Symp. Intell. Inf. Technol. Appl., pp. 294–298, 2009.

[4] C. Vázquez and W. J. Tam, "CRC-CSDM: 2D to 3D conversion using colour-based surrogate depth maps," in Int. Conf. 3D Syst. Appl. (3DSA), Tokyo, Japan, May 2010.

[5] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, "Generating the depth map from the motion information of H.264-encoded 2D video sequence," EURASIP J. Image Video Process., Article ID 108584, 2010.

[6] Z. Zhang, Y. Wang, T. Jiang, and Wen Gao, "Visual Pertinent 2D-to-3D Video Conversion By Multi-cue Fusion," ICIP, IEEE, 2011.

[7] P. V. Harman, "Home-based 3D entertainment—An overview," in IEEE Int. Conf. Image Process., Vancouver, pp. 1–4, 2000.

[8] P. Harman, J. Flack, S. Fox, and M. Dowley, "Rapid 2D to3D conversion," in SPIE Conf. Stereoscopic Displays and Virtual Reality Systems IX, vol. 4660, pp. 78–86, 2002.

[9] ITU-T, "H.264: Advanced video coding for generic audiovisual services", Recommendation H.264, June, 2011.

[10] Y. Lu, S. Chen, and J. Wang, "Overview of AVS-video coding standards", Signal Processing: Image Communication Volume 24, Issue 4, Pages 247-262, April 2009.

[11] C. Cheng, C. Li, and L. Chen, "A 2D-to-3D conversion system using edge information," Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on , vol., no., pp.377-378, 9-13 Jan. 2010.