Conversion of H.264-Encoded 2D Video to 3D Format

M. T. POURAZAD, P. NASIOPOULOS, and R. K. WARD
Electrical and Computer Engineering Department, University of British Columbia

***Abstract*-- An efficient 2D to 3D video conversion method that utilizes the H.264 motion estimation procedure is proposed. The depth information is determined based on the objects' motion information within the scene. Performance evaluations show that our approach outperforms the other existing H.264-based depth map estimation technique by up to 2.2 dB PSNR, and provides more realistic depth information about the scene. Subjective comparisons also confirm the superiority of our method.**

I. INTRODUCTION

The availability of a wide variety of 3D content is one of the requirements for the successful introduction of 3D TV to the consumer market. While capturing videos in 3D format is one solution, conversion of existing 2D material to 3D format will help enabling the 3D market as well as introducing a new market opportunity for 2D content owners. 2D-to-3D conversion techniques are based on the human visual depth perception mechanism. This mechanism is assisted by monocular depth cues (such as motion, shade and comparative size) that enable the perception of the relative distance of objects within a real scene. In this paper, we present a new and effective 2D-to-3D conversion method which is based on the motion information between consecutive frames. Our proposed scheme estimates the motion of objects within the scene. Then this information is converted to a depth map using a non-linear scaling model that is based on the relationship between depth and the viewing distance of the moving objects.

II. PROPOSED 2D-TO-3D CONVERSION SCHEME

Our objective is to approximate the disparity (which is defined as the displacement between the left and right camera images in stereoscopic set-up). It is estimated using the displacement of moving objects over two consecutive 2D-video frames. To ensure compatibility with future 3D networks and players and with real-time implementation, we use H.264-based motion information to derive the displacement of objects over two consecutive frames (H.264 has been chosen as the platform for 3D TV applications). This information is readily available in the form of motion vectors (MVs) at the receiver-end at no additional computational cost.

A. Motion Correction

H.264/AVC motion vectors are obtained by optimizing the compression performance and not by maximizing the accuracy of the estimated motion of the objects. Thus, the matching blocks determined by a motion vector may not refer to the same part of a moving object. For such cases, MV correction is necessary, unless the block includes boundary pixels of a moving object [1]. For MV correction, the motion of the block is estimated as the median of the motion of the surrounding blocks which belong to a similar object/region as the block [1]. In order to determine which adjacent blocks belong to the same object, we use an unsupervised motion-assisted segmentation approach. First we implement the mean-shift procedure to segment each frame in a five dimensional feature space (luminance and chrominance components and coordinates of each pixel within the frame) without any presumption about the number of objects [2]. As a result, each frame is segmented into different regions, each of which contains pixels that are similar in color and also contiguous in the frame. To ensure the moving objects are not over segmented, our scheme compares non-zero motion information of the segmented boundaries to merge the segmented regions belonging to a common object. This motion-assisted merging process is based on the idea that all segmented regions of the same solid object should have the same motion vector. Note that in this process, only non-zero motion information at the boundary of an object is considered. The reason is that the H.264-based motion estimation process assigns zero-value motion vector or a skip-mode flag to the flat areas of segmented objects (usually middle part), regardless of the motion of the object. Thus the motion information of flat areas of segmented objects may not accurately reflect the motion of the object. In our motion-assisted merging process, the non-zero MVs of the boundaries of each segmented region are compared with those of its adjacent segments, and the segments with most similar motion information are clustered. Clustering takes place in a four-dimensional feature space (the coordinates of each boundary pixel and horizontal and vertical components of motion) using the mean-shift algorithm. Note that the H.264 estimated motion vectors are block-based, meaning that for this step all the pixels within each block are assigned the estimated motion for the block. This does not hamper the accuracy of this step since the boundary pixels contribute the most in estimating the motion of the entire block.

B. Object-based Motion Estimation

In our implementation, we need the displacement value of each object due to its motion over two consecutive frames. However, the H.264-based motion information yields block motion values as it is block-based. To address this issue, we categorized the blocks within the frame to body-region and boundary-region blocks based on the results segmentation process (since the characteristics of these blocks are different in the presence of motion). For body-region blocks (mostly 16x16, 16x8, 8x16 and 8x8), we use an advanced affine motion model, which transforms translational H.246-based MVs to affine MVs through a low-complexity process [3]. This process divides all blocks into 8x8 sizes. Then, the H.264 MVs are assigned to the middle 4-pixels and the MV of the rest of pixels within the block is estimated as the weighted average of the MVs of neighboring blocks. For each pixel the weighting average is defined based on the distance of the pixel from the closest middle 4-pixels of the block and its neighboring blocks (for details see [3]). If this procedure is performed on boundary-region blocks, the borders of the objects will be blurred and small details or objects will be removed from the rendered 3D image. Thus, for boundary-region blocks (mostly 8x8, 4x8, 8x4 and 4x4 sizes), with non-zero MVs, we classify the pixels within each block as either background pixels or object pixels [1]. The object pixels are assigned the H.264 estimated MV, while background pixels are assigned the median of MVs of the surrounding non-object pixels. For boundary-region blocks with zero MVs, all pixels are assigned zero motion.

After finding the pixel-based motion information, we determine the motion of objects since all the pixels within each segmented object region must have similar motion. A simple solution is to use the average motion of all the pixels within each segmented object.

This work was supported by the British Columbia Innovation Council.

The problem with this approach is the presence of H.264 zero-value MVs or skip-mode flags for the flat areas of moving objects. Thus, in our scheme, we average only the non-zero motion object-pixels and this is used in estimating the object's depth value. If all the MVs are zero then the object-based motion will be also zero.

C. Generation of Depth Values and 3D streams

Since disparity is the horizontal displacement between two camera images, we use only the absolute value of the horizontal component of the motion vectors (i.e., $abs(MV_x)$) in estimating the depth map. In order to ensure that the distribution of motion with respect to depth is similar to that of disparity versus depth, we apply the non-linear scaling model presented in [1] to $abs(MV_x)$s. This increases the contrast between motion values, which in turn enhances the visual depth perception. This scaling model is designed so that the closer the object, the greater is the scaling parameter. The scaled $abs(MV_x)$ values are approximated as depth values. For stereoscopic video format, the right-eye stream is rendered based on the estimated depth map and the 2D video sequence [4], and the original 2D video is used as the left-eye stream [5].

III. PERFORMANCE EVALUATION

The performance of our proposed 2D-to-3D conversion method is tested versus that of [6], using two 2D video sequences known as "Inter-view" and "Orbi". We chose these sequences since their true depth maps also exist (measured by 3D-depth range camera). In our experiment, the selected spatial, color and motion bandwidths for the mean-shift process were 6.5, 7, and 2, respectively [2]. We chose 256 depth layers in the non-linear model for scaling the $abs(MV_x)$s [1].

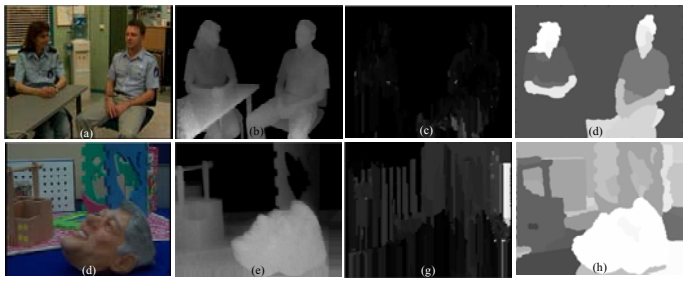

Fig.1 Two 2D video sequences (a&d); recorded depth map (b&e); estimated depth map by [6] (c&g); estimated depth map by our approach (d&h).

Fig. 1 shows a snapshot of the original 2D stream, the original depth map, and the estimated depth maps of [6] and of our approach. Note that our approach yields more realistic depth compared to [6]. Unlike [6], our method can approximate the depth information of an entire object even if only partial motion information of the object is available. This is due to our object-motion estimation procedure. As expected, both techniques fail to estimate depth map for static objects (e.g., table in Fig. 1c and Fig. 1d). Improvement on retrieving depth information of static objects may require integration of our approach with other depth cues, such as sharpness, and it is recommended path for future research. The subjective visual quality of the 3D streams was assessed against the original depth map based on the ITU-R Recommendation [7]. Eighteen people graded the videos from 1 to 10 in terms of 3D visual perception and visual picture quality. Table 1 illustrates the average scores of our subjective test.

TABLE 1 – AVERAGE SUBJECTIVE TEST SCORES FOR TEST STREAMS.

Subjective Score (out of 10)	3D Visual Perception			Visual Picture Quality		
	Actual depth	Our method	Existing method	Actual depth	Our method	Existing method
Interview	5.40	6.85	5.60	7.80	7.10	5.80
Orbi	7.00	7.90	7.20	6.70	6.55	5.00

The original stereoscopic video had the highest scores in terms of visual picture quality. Our method yielded the highest scores in terms of 3D visual perception.This visual 3D perception improvement is due to two factors: the perceptual depth enhancement step, and the prominence of the depth for moving-objects [1]. The use of the object-motion estimation procedure in our method is successful in generating a smooth depth map. Without it, the rendered stereoscopic image would have the binocular parallax effect only for parts of moving objects, as is the case with existing methods [1, 6]. In this case, some parts of an object are perceived in 3D format and the rest in 2D, something that may cause visual discomfort.

For a quantitative analysis, we compared the quality of the stereoscopic videos of our method and the method in [6] against the ones rendered from the actual (recorded) depth map. Table 2 shows the PSNR values of the right views generated by our method and by [6]. The right view generated based on the actual depth map is used as a reference. We observe that our method outperforms the proposed method in [6] by 1.93 dB to 2.2 dB.

TABLE 2 – AVERAGE PSNR COMPARISON

Average PSNR (dB)	Interview	Orbi
Right views based on our method vs right views rendered based on actual depth map	36.67	32.03
Right views based on existing method vs right views rendered based on actual depth map	34.47	30.1

In addition to the above comparisons, the percentage of bad matching pixels for the estimated depth was computed as:

$$\boldsymbol{B} = \frac{1}{\boldsymbol{N}} \sum_{(x,y)} (|\, \boldsymbol{D}(x,y) - \boldsymbol{D}_r(x,y) \,| > \boldsymbol{Th}) \quad (1)$$

where N is the number of pixels within the depth map, D is the estimated depth map, D_r is the recorded depth map and Th is a error tolerance. In our experiment we use Th=1. Our method yielded 55% correctly matched pixels for Interview and 52% for Orbi, while [6] resulted in 34% for Interview and 27% for Orbi. In other words, our method outperformed the existing method by an average of 23%.

IV. CONCLUSION

We present a new and efficient method that approximates the depth map of a 2D video sequence using H.264/AVC estimated motion information. To improve the quality and smoothness of the estimated depth, our algorithm utilizes motion-assisted color segmentation. Performance evaluations show that our approach outperforms the other existing technique by up to 2.2 dB PSNR and provides more realistic depth information of the scene. The superior subjective visual quality of our generated 3D stream was also confirmed by watching the resulting 3D streams on a stereoscopic display.

REFERENCES

[1] M. T. Pourazad, P. Nasiopoulos and R.K. Ward, "An H.264-based Scheme for 2D to 3D Video Conversion", IEEE Transactions on Consumer Electronics, May 2009.

[2] D. Comaniciu, P. Meer: Mean Shift: A Robust Approach toward Feature Space Analysis, IEEE Trans. Pattern Analysis Machine Intell., Vol. 24, No. 5, 603-619, 2002.

[3] Roman C. Kordasiewicz, M. D. Gallant, Shahram Shirani: Affine Motion Prediction Based on Translational Motion Vectors. IEEE Trans. Circuits Syst. Video Techn. 17(10): 1388-1394, 2007.

[4] W. J. Tam, F. Speranza, L. Zhang, R. Renaud, J. Chan, and C. Vazquez, "Depth image based rendering for multiview stereoscopic displays: Role of information at object boundaries", Three-Dimensional TV, Video, and Display IV, Vol. 6016, pp. 75-85, 2005.

[5] L. Zhang, "Stereoscopic image generation based on depth images for 3D TV," IEEE Trans. Broadcasting, vol. 51, no.2, pp191-199, 2005.

[6] I. Ideses, L. P. Yaroslavsky, and B. Fishbain, "Real-time 2D to 3D video conversion," Journal of Real-Time Image Processing, Vol. 2, no. 1, pp. 3-9, 2007.

[7] "Methodology for the subjective assessment of the quality of television pictures", ITU-R Recommendation BT.500-11.