LETTER *Special Section on Picture Coding and Image Media Processing*

Pixel-Wise Interframe Prediction based on Dense Three-Dimensional Motion Estimation for Depth Map Coding

Shota KASAI[†a)], ***Nonmember***, **Yusuke KAMEDA**[†], ***Member***, **Tomokazu ISHIKAWA**[†], ***Nonmember***, **Ichiro MATSUDA**[†], ***Senior Member***, ***and*** **Susumu ITOH**[†], ***Fellow***

SUMMARY We propose a method of interframe prediction in depth map coding that uses pixel-wise 3D motion estimated from encoded textures and depth maps. By using the 3D motion, an approximation of the depth map frame to be encoded is generated and used as a reference frame of block-wise motion compensation.

key words: *depth map coding, motion compensation, optical flow, 3DTV, MVD*

1. Introduction

Currently, frameworks for transmitting texture video and depth information for each pixel (called a depth map) are being studied with the objective of reducing the amount of data in multiview videos of free viewpoint TV (FTV) and 3DTV [1]. In these frameworks, the 3D video format is called the multiview video plus depth (MVD) format, which consists mainly of a set of a few texture videos of any view point and a corresponding depth map. Some of the transmitted texture videos and corresponding depth maps can be used to create a texture video of another view point by using depth-image-based rendering (DIBR) techniques at the decoder side. For view synthesis, the MVD format includes camera parameters, such as the maximum and minimum distances between the camera and the subject in the captured views and the focal length of the camera.

In general, block-wise motion compensation (MC) based on block matching (BM) is mainly performed to reduce the data amount during the coding of a 2D video as a depth map in interframe prediction. However, simple block-wise MC is not always suitable for motions such as local scaling, rotation, and deformation. Thus, to predict such motions, pixel-wise MC using a technique to estimate the apparent motion (optical flow) of each pixel through real-precision was proposed in [2]. In this method, pixel-wise motion vectors (MVs) are calculated using two coded frames, $t-2$ and $t-1$, and an approximate frame of the current frame is created using these vectors, assuming the linear uniform motion of the subject. An advantage of this method is that the pixel-wise MVs do not need to be transmitted, because they can be calculated at the decoder side by using the decoded frames. Finally, this method successfully improves interframe prediction accuracy by using the created approximate frame as a reference frame of BM.

In general, when coding an MVD format, the encoded texture video can be used for efficient depth-map coding, because each texture video frame is encoded before its corresponding depth map. In [3], an interframe prediction method for coding a depth map, in which the pixel-wise MVs are calculated from the texture video, was proposed. The coded pixels in the texture video frames at $t-1$ and t can be used to predict the current pixel in the depth map. Therefore, by utilizing both the intra- and interframe information this method allows an integrated prediction. However, the approach does not clearly present the method for calculating the pixel-wise MVs from the coded texture video. For example, whether the texture video being used is monochrome or color and the method for deriving an optimal solution of pixel-wise MVs are not stated.

Although both the aforementioned proposed methods constitute superior interframe prediction methods, they may reduce the interframe prediction accuracy when encoding depth map under certain conditions. The condition is a case that the subject in the corresponding texture video largely moves in the depth direction between frames. The depth value is determined based on the maximum and minimum distances between the camera and subject (these distances are determined for each frame). Thus, a considerable change in the distance between frames leads to a considerable change in the depth value of the subject. As the previous two interframe prediction methods do not consider the change in the movement of the subject in the depth direction, they cannot respond to the change in the depth value between frames.

To solve this problem, we propose pixel-wise MC and depth compensation (DC) techniques based on a pixel-wise motion estimation method [2] using coded texture video. In our proposed method, the coded texture video is considered to be a color video and the optimum pixel-wise MVs for pixel-wise MC in the depth map coding is calculated from the texture video. Furthermore, we calculate the depth-directional motion between frames by using the calculated MVs. In other words, our method performs interframe prediction in depth map coding to predict 3D (2D and depth-directional) motion between frames by utilizing the estimated pixel-wise dense 3D MVs.

Manuscript received December 16, 2016.
Manuscript revised April 14, 2017.
Manuscript publicized June 14, 2017.

[†]The authors are with Tokyo University of Science, Noda-shi, 278–8510 Japan.
a) E-mail: kasai@itohws01.ee.noda.tus.ac.jp

DOI: 10.1587/transinf.2016PCL0007

2. Pixel-Wise Motion Estimation

A texture video and its corresponding depth map can be regarded as one set. In general, such a set is coded in the order of texture video and depth map. Thus, the pixel-wise MVs estimated using the coded texture video can be utilized for depth map coding. In this case, the pixel-wise MVs do not need to be transmitted because they can be estimated at the decoder side by using the decoded texture video.

In the pixel-wise motion estimation performed using the texture video, the method proposed in our previous paper is applied [2]. In this method, the pixel-wise MVs (optical flow vectors) are estimated using two encoded successive frames of a monochrome moving image. By considering that the texture video is a color image, it is possible to use the color information of each pixel for the motion estimation. Therefore, we extended the previous motion estimation method such that it responds to color images.

An optical flow estimation method using the color information in a texture video was proposed in [4]–[6]. In this study, it was assumed that the color space of the texture video is YUV. Thus, we denote the luminance value Y, color difference value U, and V of the continuous video function as $Y(\boldsymbol{x}, t)$, $U(\boldsymbol{x}, t)$, and $V(\boldsymbol{x}, t)$, respectively, for an image domain Ω, where $\boldsymbol{x} = (x, y) \in \Omega \subset \mathbb{R}^2$ is the pixel position and t is the time. We assume a time-invariant condition for Y, U, and V. We define the following equation that indicates the error of the interframe corresponding point as a data term.

$$E_Y{}^2(\boldsymbol{x}, \boldsymbol{u}) = (Y(\boldsymbol{x}, t) - Y(\boldsymbol{x} - \boldsymbol{u}, t-1))^2 \quad (1)$$

$$E_U{}^2(\boldsymbol{x}, \boldsymbol{u}) = (U(\boldsymbol{x}, t) - U(\boldsymbol{x} - \boldsymbol{u}, t-1))^2 \quad (2)$$

$$E_V{}^2(\boldsymbol{x}, \boldsymbol{u}) = (V(\boldsymbol{x}, t) - V(\boldsymbol{x} - \boldsymbol{u}, t-1))^2, \quad (3)$$

where $\boldsymbol{u}(\boldsymbol{x}) = (u(\boldsymbol{x}), v(\boldsymbol{x}))^\top$ is the real-precision optical flow (apparent motion). The energy functional is defined as

$$J(\boldsymbol{u}, t) = \int_\Omega (E_Y{}^2 + E_U{}^2 + E_V{}^2 + E_S)d\boldsymbol{x}. \quad (4)$$

In this study, we defined the regularizer as

$$E_S(\boldsymbol{u}, x, t) = \lambda(|\nabla u|^2 + |\nabla v|^2), \quad (5)$$

where the positive real number λ is the weight of the regularizer and $\nabla = (\partial/\partial x, \partial/\partial y)^\top$.

After $\boldsymbol{u}$ is calculated, we use bilinear interpolation to compute the pixel-wise MC estimated frame of the depth map using $\boldsymbol{u}$. The proposed method searches for the value of $0 < \lambda \leq h$ that minimizes the squared error between the pixel-wise MC estimated frame and the original frame t and sends this optimal value of λ to the decoder. Here, in the subject's boundary region, as the estimation accuracy of $\boldsymbol{u}$ tends to decrease. This is due to the influence of E_S in the energy functional. Therefore, in order to limit the influence of E_S in the subject's boundary region, we use a cross bilateral filter (CBF) [7] that targets the estimated u. By referring to the pixel value on the corresponding texture video, this CBF smoothes the estimated u.

3. Pixel-Wise Motion and Depth Compensation

The depth values (as 8-bit intensity values, in general) in a depth map are determined based on the maximum and minimum distance values between the camera and the subject. Thus, if the distance between the frames changes considerably, the depth value is also considerably changed. In such a case, a conventional MC prediction method, such as the BM method, cannot respond to the change in the depth value between frames. To solve this problem, we propose a method in which not only ordinary MC prediction but also DC prediction are performed simultaneously by using pixel-wise MVs calculated using the technique of pixel-wise motion estimation.

The 3D locations of subjects can be reconstructed by transforming depth values d_v to distance values z. This transformation is defined by the following equation [1].

$$z = \left(\frac{d_v}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) + \frac{1}{Z_{\text{far}}}\right)^{-1}, \quad (6)$$

where Z_{far} and Z_{near} are the maximum and minimum distance values, respectively. In this paper, we call this transformation the depth-distance transform (DDT). To respond to the change in the depth value between frames, we perform interframe prediction using the distance value.

The steps of the proposed scheme are as follows. Figures 1 and 2 show the outline of the proposed method. First, we create two distance maps based on the DDT of two depth maps, encoded frames $t-2$ and $t-1$. These distance maps contain the distance values z for each pixel. Second, we estimate the pixel-wise MVs by using two texture videos, encoded frames $t-2$ and $t-1$. Next, we perform pixel-wise MC by applying the estimated MVs to the distance map of frame $t-2$ after the aforementioned transformation.

In this step, the 2D position (except the depth direction) of the subject between the frames of the distance map after the pixel-wise MC and the distance map of frame $t-1$ can

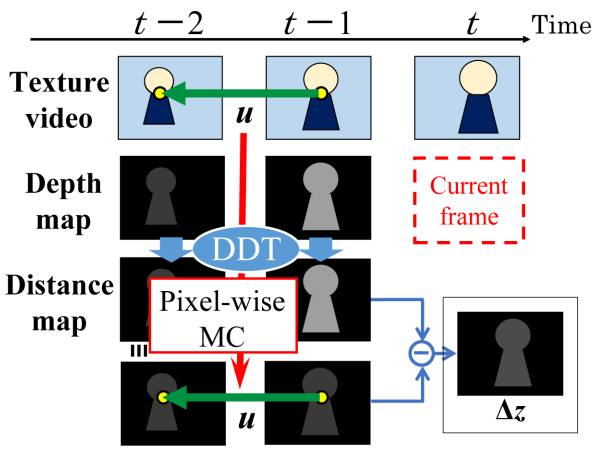


Fig. 1 Pixel-wise motion compensation and calculation of Δz.

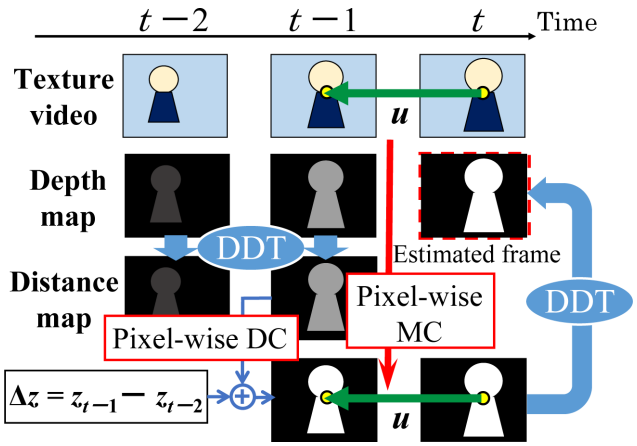


Fig. 2 Pixel-wise motion and depth compensation.

be considered to almost correspond. Thus, we calculate the difference Δz of the distance value by using these distance maps. Δz indicates the difference between two corresponding distance values, z_{t-1} and z_{t-2}, and the depth-directional motion between the frames.

However, the motions of each pixel estimated from the texture video are not always accurate. For example, owing to influences such as occlusion, regions exist where the motion cannot be accurately estimated. In such a case, pixel-wise MC causes considerable distortion. Thus, the region where the 2D position of the subject between the frames does not correspond increases, and therefore, it is difficult to calculate the appropriate Δz in the region. Therefore, in this method, filter processing is performed using a median filter (MF) to the calculated Δz. In other words, we remove as much as possible the region for which we cannot obtain the appropriate Δz.

Finally, we perform the pixel-wise MC and DC, as shown in Fig. 2. Specifically, the proposed method adds Δz to the distance map at $t-1$, assuming a linear uniform motion for depth-directional motion. We define this step as pixel-wise DC. Note that in fact we use Δz after MF processing. Then, we estimate the pixel-wise MVs by using two texture videos, encoded frames, $t-1$ and t. Furthermore, we perform pixel-wise MC by applying the estimated MVs to the distance map after pixel-wise DC. Finally, we create the depth map based on the DDT from the distance map after pixel-wise MC and DC.

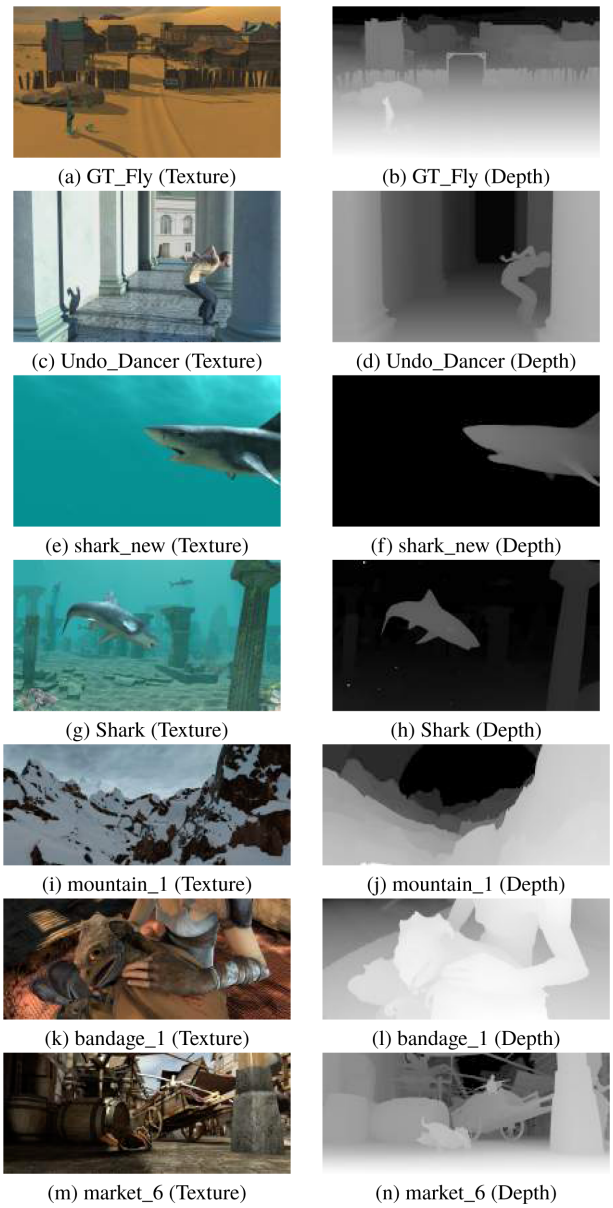
The created frame (estimated frame) in Fig. 2 is an approximation of the current depth frame to be encoded. We propose using the estimated frame as a reference frame of BM to improve the interframe prediction.

4. Experimental Results

We evaluated the performance of the proposed method by using depth map sequences, as shown in Table 1. Figure 3 shows the test sequences of depth maps and their corresponding texture videos. In this study, we used the "GT_Fly," "Undo_Dancer," "shark_new" and "Shark" depth maps, their original depth map [8], [9], and the "mountain_1," "bandage_1" and "market_6" depth maps, transformed their original depth data [10] into depth values based on DDT, and clipped their upper and lower two pixels. We evaluated the quality of the estimated and predicted frames by using the peak signal-to-noise ratio (PSNR) calculated as

Table 1 Information of test sequences

Sequence	Pixels	View
GT_Fly	1920 × 1088	3
Undo_Dancer	1920 × 1088	3
shark_new	1920 × 1088	10
Shark	1920 × 1088	5
mountain_1	1024 × 432	left
bandage_1	1024 × 432	left
market_6	1024 × 432	left



(a) GT_Fly (Texture) (b) GT_Fly (Depth)
(c) Undo_Dancer (Texture) (d) Undo_Dancer (Depth)
(e) shark_new (Texture) (f) shark_new (Depth)
(g) Shark (Texture) (h) Shark (Depth)
(i) mountain_1 (Texture) (j) mountain_1 (Depth)
(k) bandage_1 (Texture) (l) bandage_1 (Depth)
(m) market_6 (Texture) (n) market_6 (Depth)

Fig. 3 First frame of test sequences.

Table 2 Peak signal-to-noise ratio of the estimated frames before block-wise MC

Sequence	3D motion	2D motion	$t-1$
GT_Fly	38.9	31.2	27.1
Undo_Dancer	36.1	39.5	31.7
shark_new	49.1	43.4	26.0
Shark	34.4	32.2	25.1
mountain_1	39.7	26.8	25.8
bandage_1	38.3	35.8	29.0
market_6	23.9	22.0	17.2

Table 3 Peak signal-to-noise ratio of the predicted frames after block-wise MC

Sequence	3D motion	2D motion	$t-1$
GT_Fly	45.5	32.6	31.7
Undo_Dancer	46.9	48.8	38.8
shark_new	53.1	46.6	34.6
Shark	38.5	33.4	32.5
mountain_1	44.6	27.5	27.6
bandage_1	41.6	40.2	40.5
market_6	32.5	25.2	21.9

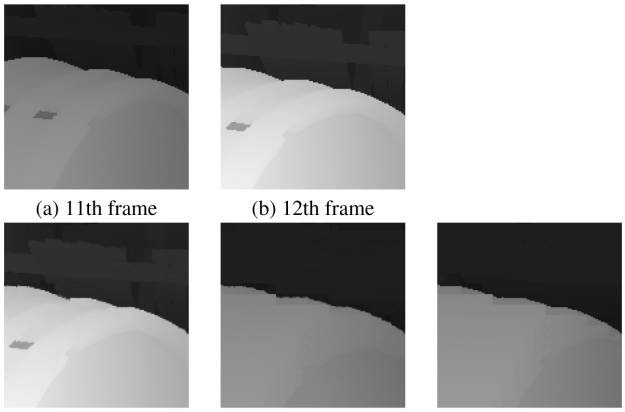
$$\mathrm{PSNR} = 10\log_{10}(255^2/\mathrm{MSE})(\mathrm{dB}). \tag{7}$$

Here, MSE denotes the mean squared error of the pixel values of both the current frame and the estimated frame or the predicted frame. The greater the PSNR value, the higher is the inter-prediction accuracy. The candidates of an estimated frame are generated according to Eq. (4) by changing λ between 0.003 and 0.192 in increments of 0.003. The candidate having the highest PSNR is then used as an estimated frame. The computational time for the pixel-wise MVs estimation between two frames of Full HD (1920×1088) video for certain λ is about 8 seconds on a desktop computer with Intel Core i7-6700K CPU (4 cores) and 16 GB memory.

Tables 2 and 3 show the average PSNR values of 13 depth map frames estimated and predicted using the following methods. In Table 2, "3D motion" is the method that estimates the current frame by performing pixel-wise MC and DC, "2D motion" is the method that estimates the current frame by performing pixel-wise MC from the depth map of frame $t-1$, and "$t-1$" is the method that sets the frame $t-1$ as the estimated frame. Table 3 shows three methods that apply BM to the estimated frame that is created by the "3D motion," "2D motion," and "$t-1$" methods. However, we assume that lossless encoding is performed; therefore, we used an original image as the encoded frame to be used for prediction. In the BM method, the number of reference frames is one (fixed), the MC block size is 8×8 pixels, and the search range is 15×15 pixels.

From the values shown in Tables 2 and 3, we can infer that the proposed method "3D motion" succeeds in improving the interframe prediction accuracy by introducing pixel-wise MC and DC based on pixel-wise 3D motion estimation as compared to "2D motion" and "$t-1$," which do not compensate for the motion in the depth direction. Further, even if we do not use the BM method, we can infer a highly accurate prediction of "3D motion."

On the other hand, the proposed method is not so good for sequences with many occlusions such as "Undo_Dancer." Since the proposed method cannot properly estimate pixel-wise MVs in the occlusion regions, the estimation accuracy of Δz also decreases. The result of "Undo_Dancer" in Table 3 shows that. For the occlusion regions, we can improve the performance of the proposed method by using multiple reference frame MC and block adaptive prediction selection, which are adopted in modern coding schemes such as H.265/HEVC [11].



(a) 11th frame (b) 12th frame

(c) 3D motion (d) 2D motion (e) $t-1$

Fig. 4 Original and predicted images.

The proposed method indicates higher performance even when a frame contains multiple objects with similar textures and the corresponding depth map has considerable distance differences, such as "mountain_1" sequence. The reason for this is that the pixel-wise MVs estimation of the proposed method does not use simple pixel value correspondence such as BM but uses multiple resolution analysis and variational method as in [2]. However, when interframe corresponding points cannot be obtained accurately due to intersection of moving objects with similar textures, the estimation accuracy decreases as in the case of the occlusion described above.

Figure 4 shows an enlarged view of a part (a lying barrel) of the 12th frame in "market_6" predicted by each method in Table 3. The figure shows that, while the previous methods, "2D motion" and "$t-1$," cannot predict the depth values (brighter pixel values) of the original 12th frame, our proposed method "3D motion" favorably predicts the depth value of the original 12th frame.

5. Conclusion

In this paper, we proposed an interframe prediction method for depth map coding that uses pixel-wise dense 3D motion. In the proposed method, pixel-wise 2D motion is estimated using the coded texture video and the motion of the depth direction is estimated using the depth value, that is, the distance value. The estimated frame is created using the estimated pixel-wise 3D motion, and it is possible to predict the realistic motion of the subject between the frames more accurately by utilizing the reference frame of BM. The experi-

mental results showed that the proposed method achieves an effective interframe prediction. In the future, we will evaluate the coding rate in lossless and lossy encoding of the proposed method.

References

[1] Y. Chen, G. Tech, K. Wegner, and S. Yea, "Test Model 11 of 3D-HEVC and MV-HEVC," JCT-3V and Video Subgroup, Feb. 2015.
[2] Y. Kameda, J. Takeichi, M. Ishibashi, I. Matsuda, and S. Itoh, "Two Stage Inter-frame Prediction using Pixel- and Block-wise Motion Compensation," Proc. of IWSSIP, pp.145–148, 2015.
[3] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-Based Inter Prediction in Coded Texture Assisted Depth Coding," IEEE Signal Process. Lett., vol.21, no.1, pp.74–78, 2014.
[4] P. Golland and A.M. Bruckstein, "Motion from Color," CVIU, vol.68 no.3, pp.346–362, 1997.
[5] R.J. Andrews and B.C. Lovell, "Color Optical Flow," Proc. of APRS Workshop on Digital Image Computing, vol.1, no.1, pp.135–139, 2003.
[6] K.R.T. Aires, A.M. Santana, and A.A.D. Medeiros, "Optical flow using color information: preliminary results," Proc. of ACM SAC, pp.1607–1611, 2008.
[7] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," ACM Transactions on Graphics, vol.23, no.3, pp.673–678, 2004.
[8] Nokia Corporation. http://mpeg3dv.nokiaresearch.com/
[9] National Institute of Information and Communications Technology (NICT). http://www.nict.go.jp/
[10] D.J. Butler, J. Wulffand, G.B. Stanley, and M.J. Black, "A naturalistic open source movie for optical flow evaluation," Proc. of ECCV, vol.7577, pp.611–625, 2012.
[11] G.J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. on Circuits Syst. Video Technol., vol.22, no.12, pp.1649–1668, 2012.