Kwon and Lee *EURASIP Journal on Image and Video Processing* (2020) 2020:11
https://doi.org/10.1186/s13640-020-00499-2

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Zoom motion estimation for color and depth videos using depth information

Soon-kak Kwon* and Dong-seok Lee

Abstract

In this paper, two methods of zoom motion estimation for color and depth videos by using depth information are proposed. Color and depth videos are independently estimated for zoom motion. Zoom for color video is scaled by spatial domain, and depth video is scaled by both spatial and depth domains. For color video, instead of existing methods of zoom motion estimation that apply all of possible zoom ratios for a current block, the zoom ratio of the proposed method is determined as the ratio of the average depth values of the current and reference blocks. Then, the reference block is resized by multiplying the zoom ratio and the reference block is mapped to the current block. For depth video, the reference block is first scaled in the spatial direction by the same methodology used for color video and then scaled by a distance ratio from a camera to the objects. Compared to the conventional motion estimation method, the proposed method reduces MSE by up to about 30% for the color video and up to about 85% for the depth video.

Keywords: Zoom motion Estimation, Inter prediction, Depth video, Depth video coding

1 Introduction

Intelligent surveillance systems for monitoring the behavior of objects are operated in various places for public safety. These systems can use not only conventional RGB videos but also infrared and depth videos to acquire new information. In order to operate the intelligent surveillance systems by transmitting the videos, an efficient encoding method is required for the various types of the videos.

In video coding standards such as H.264/AVC [1–4] and H.265/HEVC [5, 6], various methods for removing redundancies are used to compress color video. The temporal direction is one type of the redundancies of the video. The temporal redundancy is efficiently removed by motion estimation for objects in frames. The block matching algorithm (BMA) [7, 8] has been embraced as a method of motion estimation in the video coding standards. BMA estimates object motion accurately when the object size among frames is fixed. However, conventional motion estimation methods through BMA have a limitation that it estimates object motion inaccurately when the object size is changed because the size of the reference block is equal to the size of a current block.

In order to estimate various types of object motion including zoom, whose size is changed, the object motion models such as affine [9–11], perspective [12], polynomial [13], or elastic [14] can be applied. However, motion estimation methods through the motion models have high computational complexity because they need computation of model factor for each object. An improved affine model that the number of parameters is reduced from 6 to 4 has been introduced to solve this problem [15, 16]. Instead of computing the model parameters, a method of introducing a zoom ratio into the conventional BMA [17] has been proposed. However, there is a need to limit searching range of zoom ratios since the possible zoom ratios are infinite. To reduce the searching complexity of the zoom ratio, a diamond search method has been introduced to zoom ratio search [18]. Methods [19–21] for determining the zoom ratio instead of searching a zoom ratio have also been researched as follows. Superiori [19]

*Correspondence: skkwon@deu.ac.kr
Department of Computer Software Engineering, Dong-eui University, 47340 Busan, Korea

observes that directions of motion vectors (MVs) tend to align with a direction from the border to the center of the object when the object has zoom motion. Takada et al. [20] proposes a method of improving coding efficiency by calculating zoom ratios by analyzing MVs in the coded video and re-coding the video. This method has a limitation that it can only be applied in the coded video. Shukla et al. [21] proposes a method of finding warping vectors in the vertical and horizontal directions instead of the conventional BMA. Shen et al. [22] proposed a motion estimation method for extracting and matching scale-invariant feature transform (SIFT) features that are robust for rotating and scaling. Luo et al. [23] proposes a motion compensation method to detect feature points through the speeded-up robust features (SURF) algorithm in reference and current frames and find corresponding image projections by the perspective-n-point method. Qi et al. [24] proposes a 3D motion estimation method by predicting a future scene based on the 3D motion decomposition. Wu et al. [25] introduces a K-means clustering algorithm to improve a performance of motion estimation.

In this paper, a zoom estimation method for color video is first proposed by using depth information. Each pixel value in the depth video represents some distance from a depth camera to the objects. Applications of depth video have been researched in various fields such as face recognition [26–28], simultaneous localization and mapping [29, 30], object tracking [31–35], and people tracking [36–38]. The proposed method determines the zoom ratio as the ratio of the representative depth values of a current block to a reference block. The representative depth value is set to an average of depth values in each block. Then, a reference block size is determined by multiplying the current block size and the zoom ratio. The reference block is scaled to the current block size by spatial interpolation, and two blocks are compared in order to find an optimal reference block.

A method of motion estimation for depth video is also proposed in this paper. In depth video coding, studies for intra-prediction have been conducted [39–43], but studies for interprediction are insufficient. When an object in depth video has zoom motion, not only the size but also depth values of the object are scaled to a zoom rate. In order to accurately estimate the zoom motion for the depth video, we propose a 3D scaling method that is simultaneously scaling 2D spatial size and depth values of the reference block. The spatial scaling is similar to the method for the color video. After the spatial scaling, the depth values in the reference block are also scaled by multiplying the zoom ratio.

Contributions of the proposed method are as follows. The proposed method for color video encoding reduces a computational complexity for determining a zoom ratio through calculating the ratios of depth values. The proposed method for depth video encoding improves the accuracy of motion estimation through considering changes of pixels in the depth video when the object has zoom motion.

This paper is organized as follows. The proposed method is described in Section 2. In Section 3, we present the simulation results to show the improvement of motion estimation accuracy using the proposed method. Finally, we describe a conclusion for this paper in Section 4.

2 Proposed method

2.1 Relationship between depth values and object size

The size of an object and the distance from a camera appear to be inversely proportional. To clarify the relationship between the object size and the depth value of the depth frame, object widths in captured pictures are measured while moving a diamond-shaped object at intervals of 0.5 m from 1 m to 4 m as shown in Fig. 1.

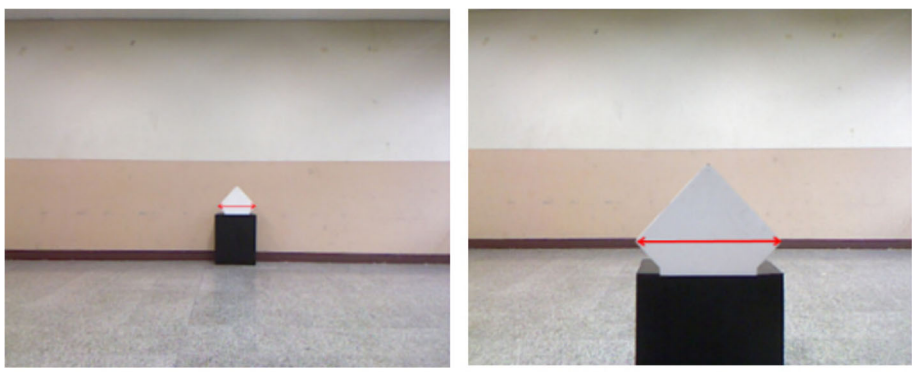


Fig. 1 Measurement of relationship between distance and width of object. **a** 4 m and **b** 1 m

The relationship between the width and distance of the object is described as shown in Fig. 2. The measured relationship can be approximated with a fitting equation as follows:

$$P = \frac{\beta}{d^{\alpha}}, \tag{1}$$

where P means the number of pixels of the object width shown in red arrow in Fig. 1, d means the distance from the camera, and α and β mean constant values. In the case of Fig. 2, α and β are measured as 0.965 and 214.59, respectively.

2.2 Relationship between depth values and object size

When the zoom motion of an object occurs between the current and reference picture, a size of the object is zoomed as the distance moved toward the camera. Therefore, the size of the reference block should be determined through the distance in order to estimate the object motion which has zooming. The depth information has distances from the camera at each pixel. Therefore, the zoom ratio between the current and reference blocks can be calculated through the depth information. The averages of the depth values in the current and reference blocks are assumed as distances of each block. If the zoom ratio s is defined as the ratio of the number of the pixels between the current and reference blocks, s is calculated by substituting the number of pixels of the current and reference blocks into Eq. (1) as follows:

$$s = \frac{P_{\text{ref}}}{P_{\text{cur}}} = \frac{\beta}{(\overline{d_{\text{ref}}})^{\alpha}} \Big/ \frac{\beta}{(\overline{d_{\text{cur}}})^{\alpha}}, \tag{2}$$

where $\overline{d_{\text{cur}}}$ and $\overline{d_{\text{ref}}}$ mean the representative depth values of the current and reference blocks, respectively, and P_{cur} and P_{ref} mean the number of pixels of the current and reference blocks, respectively. A simplified expression of Eq. (2) is as follows:

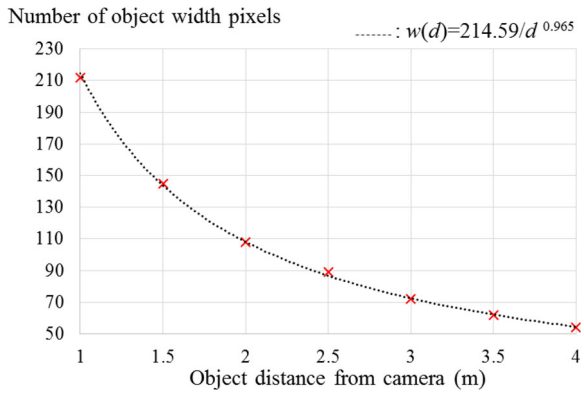


Fig. 2 Approximation of relationship between distance and width of object

$$s = \left(\frac{\overline{d_{\text{cur}}}}{\overline{d_{\text{ref}}}} \right). \tag{3}$$

When a size of the current block is assumed as $m \times n$, the size of the reference block is determined as $sm \times sn$. The reference block is scaled by interpolation so that the size of the reference block is equal to the size of the current block. Figure 3 shows a flowchart of the proposed zoom motion estimation for the color video and Fig. 4 shows processes of the proposed method.

Figure 5 shows an example of zoom motion estimation for the color video. Areas surrounded by the red rectangle in Fig. 5 a and b are the 8 × 8 current and reference blocks, respectively, and Fig. 5 c and d show depth values in each blocks. $\overline{d_{\text{cur}}}$ and $\overline{d_{\text{ref}}}$ of 8 × 8 current and reference blocks in depth pictures are about 2322.312 and 2469.523, respectively, so s is calculated as about 0.940 if α is set to 0.965. Therefore, the size of the reference color block is determined to 7 × 7. Then, a 7 × 7 reference color block is scaled so that the reference block size is equal to the current block size. The mean square errors (MSEs) of conventional and proposed motion estimation methods are about 169.734 and 74.609, respectively. These results shows the proposed zoom motion estimation method is more accurate when the object in the video has zoom motion.

2.3 Zoom motion estimation for depth video

In depth video, the distance of an object from the depth camera is changed when the object has zoom motion, so the depth values of the object are changed as shown in Fig. 6. Therefore, not only the size but also depth values of the object should be considered for the zoom motion estimation for depth video.

A method of 3D scaling is introduced for the zoom motion estimation for depth video. 3D scaling means that depth axis scaling has been added to the 2D spatial scaling that scales the block size. The flowchart of 3D scaling is shown in Fig. 7.

In 3D scaling, the zoom ratio calculation and the size determination of a reference block are the same as the processes of zoom motion estimation for previous color video. Then, the depth values of the size-scaled reference block are scaled by the following equation:

$$R_i(i,j) = s \times R(i,j), \tag{4}$$

where $R(i,j)$ and $R_i(i,j)$ mean original and scaled depth values in position (i,j), respectively.

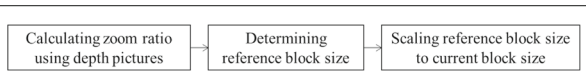


Fig. 3 Flowchart of proposed method for color video

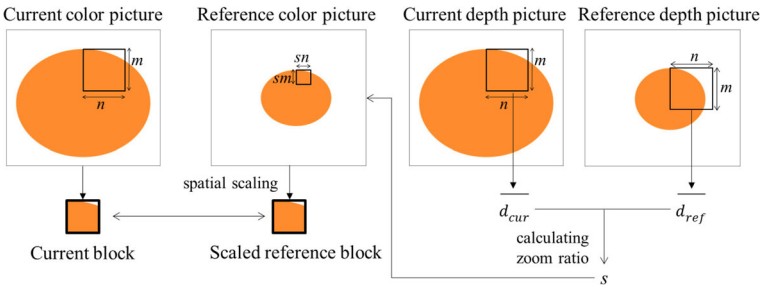
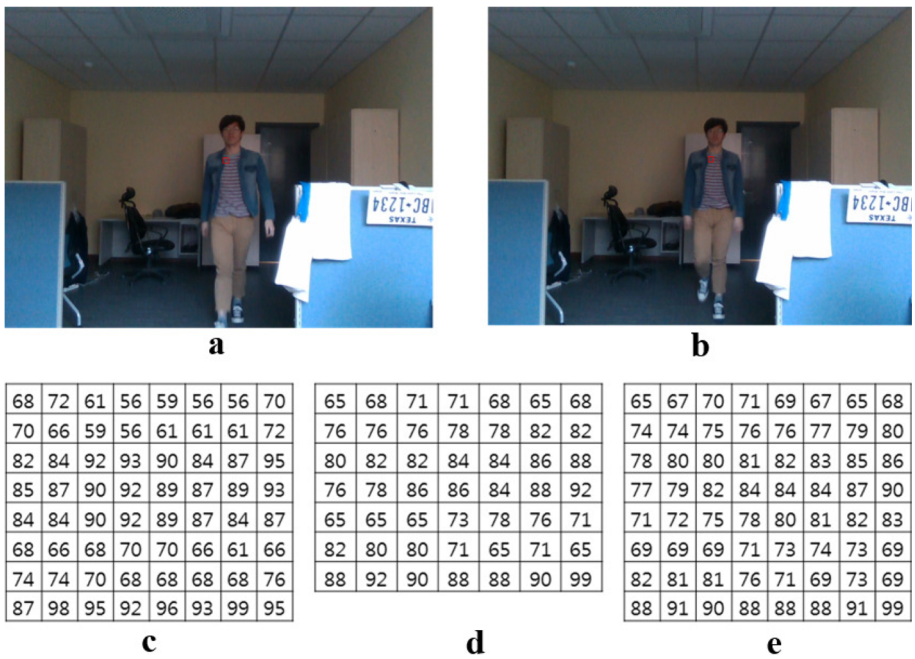


Fig. 4 Processes of proposed method for color video



a

b

68	72	61	56	59	56	56	70
70	66	59	56	61	61	61	72
82	84	92	93	90	84	87	95
85	87	90	92	89	87	89	93
84	84	90	92	89	87	84	87
68	66	68	70	70	66	61	66
74	74	70	68	68	68	68	76
87	98	95	92	96	93	99	95

c

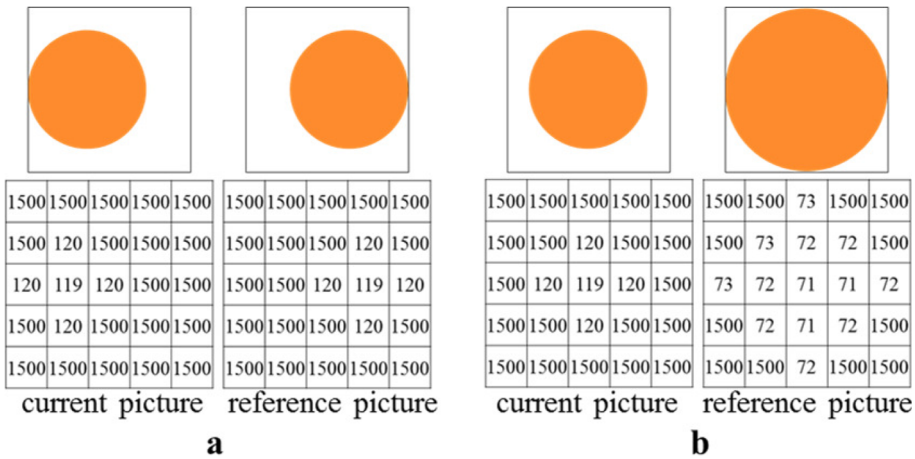
65	68	71	71	68	65	68
76	76	76	78	78	82	82
80	82	82	84	84	86	88
76	78	86	86	84	88	92
65	65	65	73	78	76	71
82	80	80	71	65	71	65
88	92	90	88	88	90	99

d

65	67	70	71	69	67	65	68
74	74	75	76	76	77	79	80
78	80	80	81	82	83	85	86
77	79	82	84	84	84	87	90
71	72	75	78	80	81	82	83
69	69	69	71	73	74	73	69
82	81	81	76	71	69	73	69
88	91	90	88	88	88	91	99

e

Fig. 5 Zoom motion estimation for color video. **a** Current picture, **b** reference picture, **c** current block, **d** 7 × 7 reference block, and **e** size-scaled reference block



1500	1500	1500	1500	1500	1500	1500	1500	1500	1500
1500	120	1500	1500	1500	1500	1500	1500	120	1500
120	119	120	1500	1500	1500	1500	120	119	120
1500	120	1500	1500	1500	1500	1500	1500	120	1500
1500	1500	1500	1500	1500	1500	1500	1500	1500	1500

current picture reference picture

a

1500	1500	1500	1500	1500	1500	1500	73	1500	1500
1500	1500	120	1500	1500	1500	73	72	72	1500
1500	120	119	120	1500	73	72	71	71	72
1500	1500	120	1500	1500	1500	72	71	72	1500
1500	1500	1500	1500	1500	1500	1500	72	1500	1500

current picture reference picture

b

Fig. 6 Pixel values in depth pictures including an object **a** when object moves in parallel and **b** when object has zoom motion

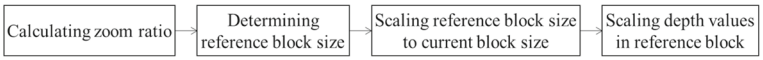


Fig. 7 Flowchart of proposed method for depth video

Figure 8 shows an example of zoom motion estimation for the depth video. Areas surrounded by the red rectangle in Fig. 8a and b are the 8 × 8 current and reference blocks, respectively, and Fig. 8 c and d show the depth values in each block. $\overline{d_{\text{cur}}}$ and $\overline{d_{\text{ref}}}$ for each 8 × 8 block are about 679.625 and 776.969, respectively, so s is calculated as about 0.874. If α is set to 0.965, the reference block size is determined as 7 × 7 as shown in Fig. 8 e when the current block size is 8 × 8. Then, a 7 × 7 reference block is scaled by the spatial scaling so that the reference block size is equal to the current block size. After that, depth values in 2D scaled reference block is scaled as shown in Fig. 8 g. MSEs of conventional and proposed methods are about 9482.97 and 3.48, respectively. These results show that the 3D scaling improves an accuracy of the motion estimation for the depth video.

2.4 Zoom motion estimation for variable-size block

The video coding standard provides the variable-size block that groups blocks which have similar MVs in order to reduce the number of coding blocks. In the motion estimation of H.264/AVC [1–4], the size of variable-size block is allowed to be 16 × 16, 16 × 8, 8 × 16, and 8 × 8 when the macroblock size is 16 × 16 and 8 × 8, 8 × 4, 4 × 8, and 4 × 4 when the macroblock size is 16 × 16. Figure 9 shows the division of a macroblock in the variable-size block. The modes of variable-size block are determined by comparing sum of absolute errors (SAEs) or sum of square errors (SSEs) of each variable-size block.

In addition, an introduction of the variable-size block can solve a problem that is difficult to determine the representative depth value of a mixed block having foreground object and background. For the mixed block, the

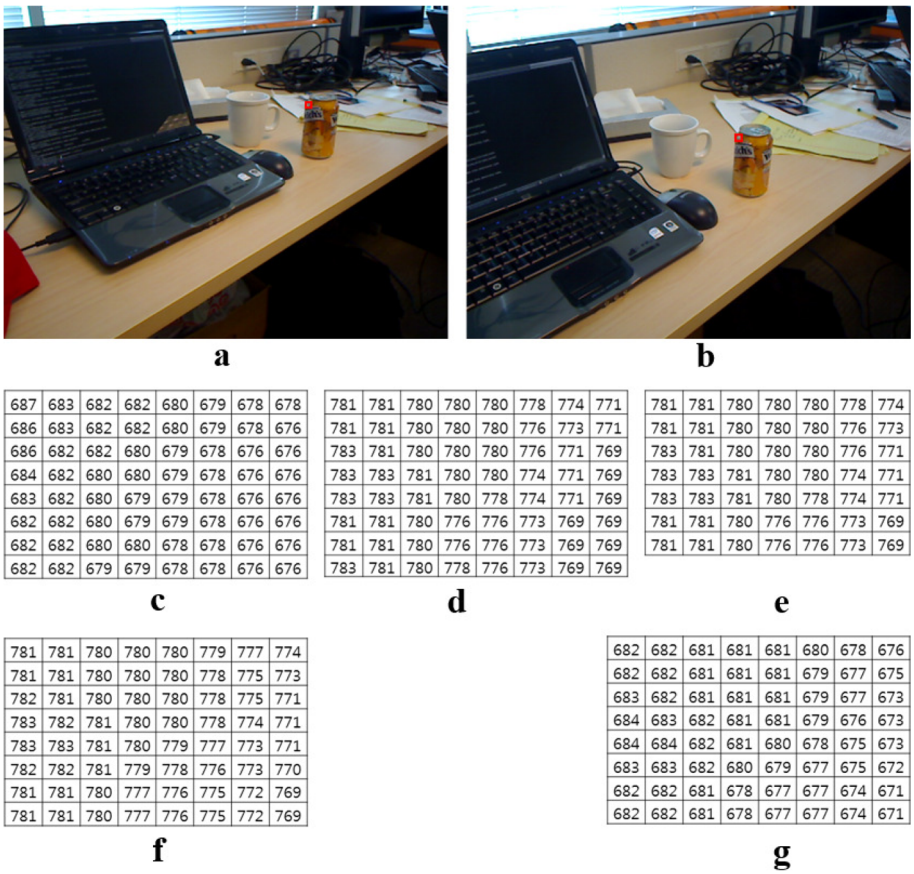


Fig. 8 3D scaling in proposed method for depth video. **a** Color current picture, **b** color reference picture, **c** current block, **d** reference block, **e** 7 × 7 reference block, **f** size-scaled block, and **g** value-scaled block

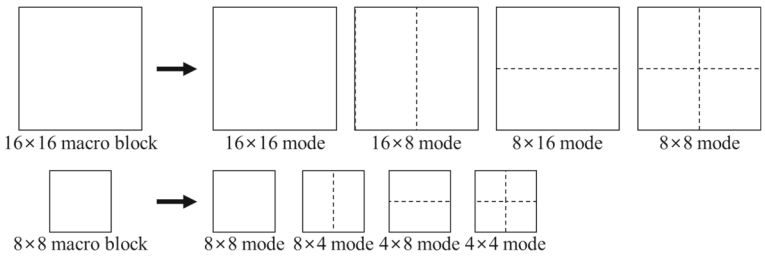


Fig. 9 Variable-size block in H.264/AVC

representative depth value is determined as an average value of the depth values of background and foreground, and then this causes the inaccurately zoom ratio. This problem can be solved by dividing the block into smaller size blocks so that each block has only background or foreground object.

The proposed method can provide estimation for variable-size block. The variable-size block is applied independently to both color and depth videos. When the size of sample block is 16 × 16, SAE for the original block and sums of SAEs for partitioned block are 16 × 16, 16 × 8, 8 × 16, and 8 × 8. In motion estimation for partitioned block, coding of each MVs for partitioned block should also be considered. In the case of comparing between 16 × 16 and 16 × 8 variable-size blocks, the equation for comparing SAEs is as follows:

$$\mathrm{SAE}_{16\times16} \geq \sum \mathrm{SAE}_{16\times8} + T_{16\times8}, \tag{5}$$

where $\mathrm{SAE}_{16\times16}$ and $\mathrm{SAE}_{16\times8}$ mean SAEs for original block and partitioned block as 16 × 8, respectively, and $T_{16\times8}$ means a threshold considering MVs. If the 16 × 16 sample block satisfies Eq. (5), this block can be partitioned into 16 × 8.

3 Results and discussion

In order to measure motion estimation accuracies of the proposed zoom motion estimation, we use the depth video datasets [44] that the camera moves forth or back as shown in Fig. 10 a and b, and we capture videos in which 1 or 2 people move back and forth while the position of the camera is fixed as shown in Fig. 10 c and d. The videos in Fig. 10 a and b are captured by Microsoft Kinect, and the videos in Fig. 10 c and d are captured by Intel Realsense D435. The resolutions of color and depth videos are specified as 640 × 480. We used 30 consecutive frames that has the most prominent zoom motion in each video. The reference picture basically has a picture gap from the current picture. The full-search method is applied as the search method for BMA. The search range is set to ± 15 while the sizes of the sample block are set to 8 × 8 and 16 × 16. α in Eq. (3) is set to 0.965. In the color videos, only a gray channel is used. The searching pixel unit is limited as 1/2 pixel in the case of the color video and 1 pixel in the case of the depth video.

In the proposed method, the RD optimization method can be used to determine the motion estimation mode. However, this paper does not discuss the coding method of depth video. Therefore, the estimation mode for each block is selected by following equation:

Fig. 10 Color and depth videos for simulation. **a** Bedroom, **b** basement, **c** a man, and **d** two men

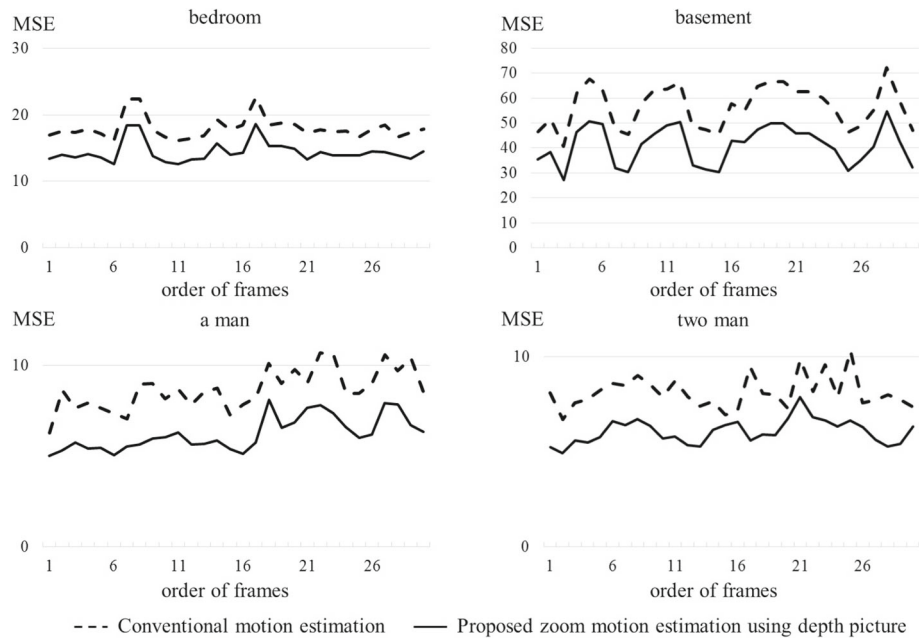


Fig. 11 Comparison of MSEs between conventional and proposed methods for color videos (8 × 8 block size)

$$\mathrm{SSE_{ME}} > \mathrm{SSE_{ZME}} + T_{\mathrm{mode}}, \tag{6}$$

where $\mathrm{SSE_{ME}}$ and $\mathrm{SSE_{ZME}}$ mean SSE for the conventional and proposed methods. If a block satisfies Eq. (6), then the motion estimation mode of this block is selected as the zoom motion estimation. In this simulation, T_{mode} is determined as the following equation:

$$T_{\mathrm{mode}} = 2mn, \tag{7}$$

where m and n mean the height and width of a current block, respectively.

Figures 11 and 12 show MSEs of motion estimation for the color videos through the conventional and proposed methods. A picture gap between the current picture and the reference picture is 1. The accuracies of motion estimation by the proposed method are improved.

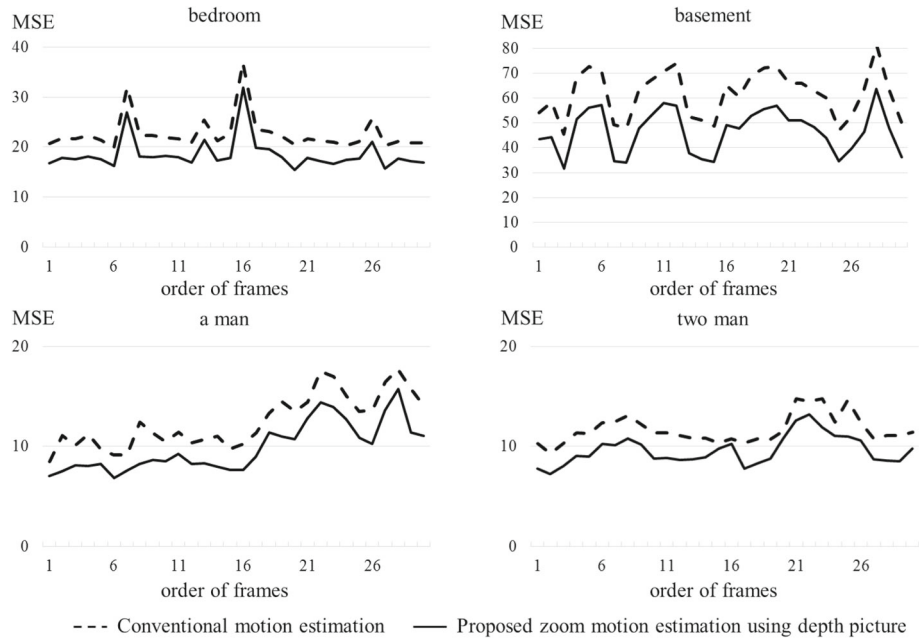


Fig. 12 Comparison of MSEs between conventional and proposed methods for color videos (16 × 16 block size)

Table 1 Averages of MSEs in color video according to frame gap in 8 × 8 block size

Video name	Picture gap	$\overline{\text{MSE}_{\text{ME}}}$	$\overline{\text{MSE}_{\text{ZME}}}$	$\Delta\overline{\text{MSE}}$	$\Delta\overline{\text{MSE}}/\ \overline{\text{MSE}_{\text{ME}}}$	Selected rate of zoom estimation mode (%)
	1	15.883	11.918	3.965	24.96%	10.5
Bedroom	2	18.971	15.062	3.909	20.61%	13.9
	3	20.258	16.177	4.081	20.15%	14.7
	1	53.952	38.521	15.431	28.60%	13.7
Basement	2	62.147	48.331	13.816	22.23%	16.2
	3	62.505	50.052	12.453	19.92%	16.0
	1	8.671	6.235	2.436	28.09%	2.47
A man	2	14.115	11.441	2.674	18.94%	3.97
	3	17.600	14.735	2.865	16.28%	4.85
	1	7.469	5.310	2.159	28.91%	2.25
Two men	2	10.806	8.211	2.595	24.01%	4.05
	3	13.489	11.091	2.398	17.78%	6.31

Tables 1 and 2 show the average MSEs according to the frame gap between the current picture and the reference picture. In Tables 1 and 2, $\overline{\text{MSE}_{\text{ME}}}$ and $\overline{\text{MSE}_{\text{ZME}}}$ mean averages of MSEs for conventional and proposed motion estimation methods and $\Delta\overline{\text{MSE}}$ means improved MSE by the proposed zoom motion estimation. The picture gap between the current and reference pictures is farther, and the number of selected block as the zoom estimation mode is larger. In color image, blocks including the object boundary region are mainly selected as the zoom motion estimation mode. This means that when the color video has the zoom motion, regions of the object boundaries are particularly affected in conventional motion estimation method.

Figures 13 and 14 shows MSEs of motion estimation in the depth videos through the conventional and the proposed methods. A picture gap between the current picture and the reference picture is 1. The accuracies of motion estimation by the proposed method are more improved than in the case of the color videos. Figure 15 shows zoom ratios in the proposed zoom motion estimation for depth videos. The zoom motion estimation mode is selected for almost all the areas where the zoom motion occurs.

Tables 3 and 4 show the average MSEs according to the picture gap between the current picture and the reference picture. Similar to the case of color images, the picture gap between the current and reference pictures is farther, and the number of selected block as the zoom estimation mode is larger.

Estimation accuracies and reduction in the number of MVs through the variable-size block are measured in Tables 5, 6, 7, and 8. Thresholds of the block partition in Eq. (6) are set as follows: $T_{16\times8}$ and $T_{8\times16}$ are set to $16^2/2$, $T_{8\times8}$ is set to 16^2, $T_{8\times4}$ and $T_{4\times8}$ are set to $8^2/2$, and $T_{4\times4}$ is set to 8^2. Tables 5, 6, 7, and 8 show MSEs and a number of each block size in a variable-size block allowing block

Table 2 Averages of MSEs in color video according to frame gap in 16 × 16 block size

Video name	Picture gap	$\overline{\text{MSE}_{\text{ME}}}$	$\overline{\text{MSE}_{\text{ZME}}}$	$\Delta\overline{\text{MSE}}$	$\Delta\overline{\text{MSE}}/\ \overline{\text{MSE}_{\text{ME}}}$	Selected rate of zoom estimation mode (%)
	1	19.945	15.619	4.326	21.69%	10.5
Bedroom	2	25.486	21.030	4.456	17.48%	13.9
	3	28.575	23.781	4.794	16.78%	14.7
	1	58.634	43.794	14.840	25.31%	13.7
Basement	2	69.201	56.263	12.938	18.70%	16.2
	3	70.680	58.833	11.847	16.76%	16.0
	1	12.475	9.900	2.575	20.64%	2.47
A man	2	21.677	18.676	3.001	13.84%	3.97
	3	27.437	24.087	3.350	12.21%	4.85
	1	10.567	8.426	2.141	20.26%	2.25
Two men	2	16.590	14.175	2.415	14.56%	4.05
	3	21.907	19.367	2.540	11.59%	6.31

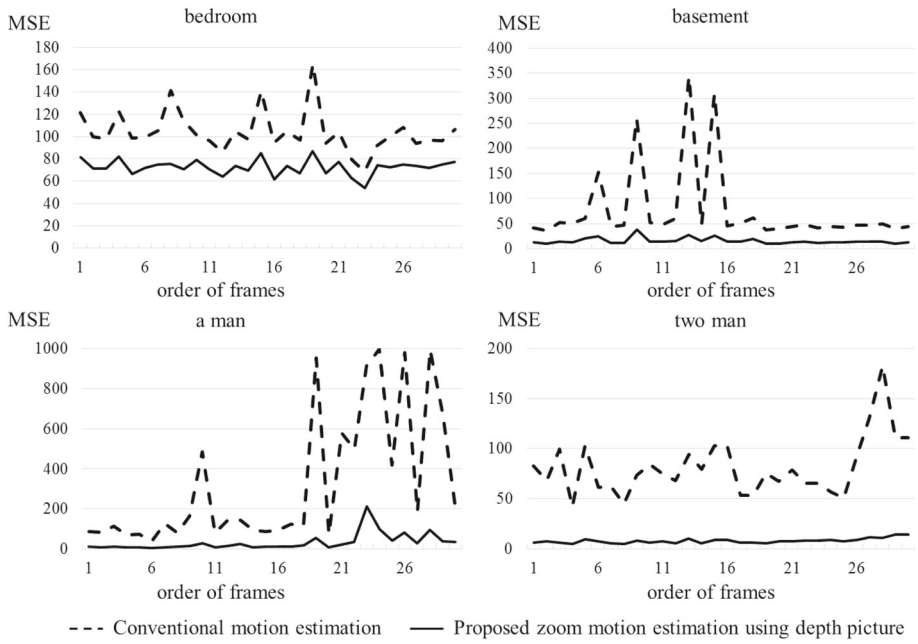


Fig. 13 Comparison of MSEs between conventional and proposed methods for depth videos (8 × 8 block size)

sizes of 16 × 16, 16 × 8, 8 × 16, and 8 × 8, and in a variable-size block allowing block sizes of 8 × 8, 8 × 4, 4 × 8, and 4 × 4. In Tables 5 and 6, MSE_{VB} means MSEs of the variable-size block and $MSE_{16\times16}$, $MSE_{8\times8}$, and $MSE_{4\times4}$ means MSEs of the fixed-size block. In Tables 7 and 8, notations such as $MV_{16\times16}$ and $MV_{16\times8}$ mean the number of MVs in the variable-size block, $MV_{fixed(8\times8)}$ means the number of MVs in the fixed-size block, and $\sum MSE_{VB}$ means the sum of the number of MVs in the variable-size block. MSEs in the variable-size block are similar to the fixed-size block whose the block size is equal to the smallest size in allowed size. The number of MVs is greatly

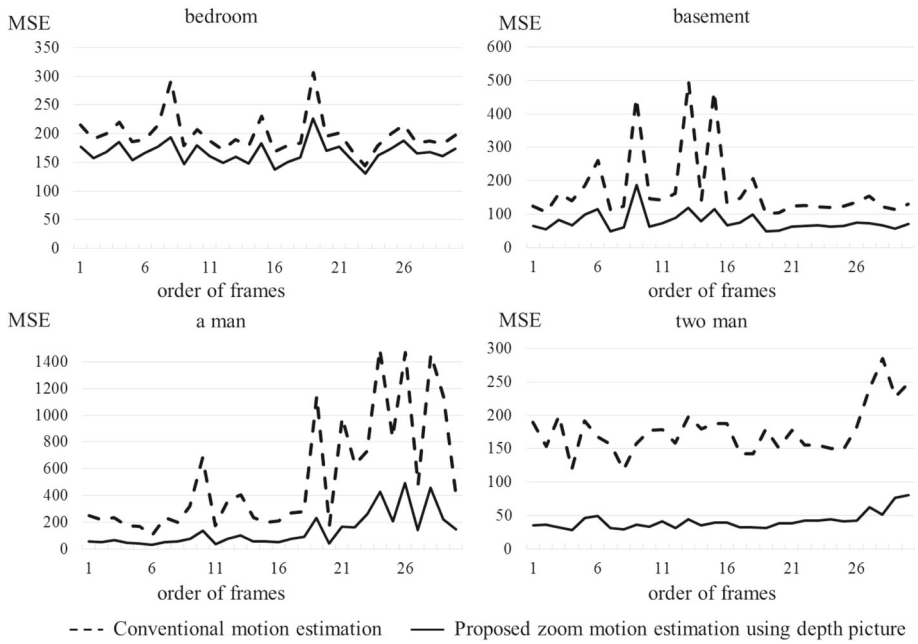


Fig. 14 Comparison of MSEs between conventional and proposed methods for depth videos (16 × 16 block size)

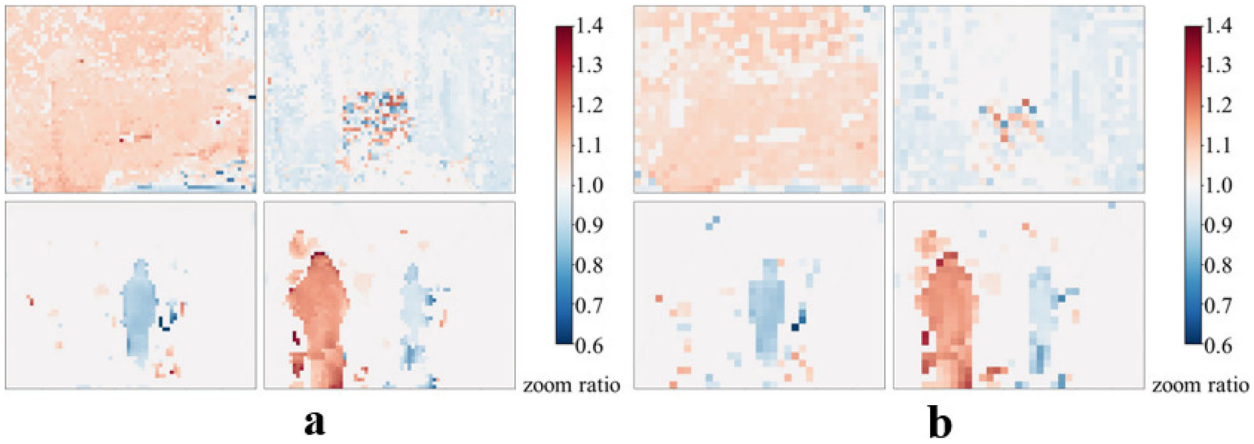


Fig. 15 Zoom ratios for simulation depth videos. **a** 8 × 8 block size and **b** 16 × 16 block size

Table 3 Averages of MSEs in depth video according to frame gap in 8 × 8 block size

Video name	Picture gap	$\overline{MSE_{ME}}$	$\overline{MSE_{ZME}}$	$\Delta\overline{MSE}$	$\Delta\overline{MSE}/\overline{MSE_{ME}}$	Selected rate of zoom estimation mode (%)
Bedroom	1	104.263	72.647	31.616	30.32%	10.5
	2	131.821	80.936	50.885	38.60%	13.9
	3	174.051	85.729	88.322	50.74%	14.7
Basement	1	76.351	15.685	60.666	79.46%	13.7
	2	98.926	18.464	80.462	81.34%	16.2
	3	120.149	19.115	101.034	84.09%	16.0
A man	1	542.273	32.968	509.305	93.92%	2.47
	2	1651.302	61.946	1589.356	96.25%	3.97
	3	3088.303	95.173	2993.130	96.92%	4.85
Two men	1	81.302	8.143	73.159	89.98%	2.25
	2	194.368	10.847	183.521	94.42%	4.05
	3	502.936	14.415	488.521	97.13%	6.31

Table 4 Averages of MSEs in depth video according to frame gap in 16 × 16 block size

Video name	Picture gap	$\overline{MSE_{ME}}$	$\overline{MSE_{ZME}}$	$\Delta\overline{MSE}$	$\Delta\overline{MSE}/\overline{MSE_{ME}}$	Selected rate of zoom estimation mode (%)
Bedroom	1	198.284	166.766	31.518	15.90%	10.5
	2	257.755	192.514	65.241	25.31%	13.9
	3	332.117	206.987	125.13	37.68%	14.7
Basement	1	172.938	77.911	95.027	54.95%	13.7
	2	225.602	93.993	131.609	58.34%	16.2
	3	277.677	99.45	178.227	64.19%	16.0
A man	1	647.179	139.144	508.035	78.50%	2.47
	2	2048.382	277.636	1770.746	86.45%	3.97
	3	3601.99	411.556	3190.434	88.57%	4.85
Two men	1	176.645	41.904	134.741	76.28%	2.25
	2	334.603	53.456	281.147	84.02%	4.05
	3	707.025	69.863	637.162	90.12%	6.31

Table 5 Comparison of MSEs between variable- and fixed-size blocks (16 × 16, 16 × 8, 8 × 16, and 8 × 8)

Frame order	$MSE_{16\times16}$	$MSE_{8\times8}$	MSE_{VB}	$MSE_{VB} - MSE_{8\times8}$
1	39.555	7.909	7.969	0.059
2	35.941	6.311	6.376	0.065
3	48.032	9.497	9.553	0.056
4	34.747	6.318	6.384	0.066
5	29.623	5.627	5.701	0.074
6	20.695	3.677	3.741	0.064
7	33.325	6.600	6.675	0.076
8	38.762	7.683	7.756	0.073
9	48.284	9.819	9.883	0.064
10	93.755	19.189	19.256	0.067

Table 6 Comparison of MSEs between variable- and fixed-size blocks (8 × 8, 8 × 4, 4 × 8, and 4 × 4)

Frame order	$MSE_{8\times8}$	$MSE_{4\times4}$	MSE_{VB}	$MSE_{VB} - MSE_{4\times4}$
1	7.909	1.999	2.071	0.072
2	6.311	1.588	1.657	0.068
3	9.497	2.286	2.367	0.081
4	6.318	1.599	1.668	0.068
5	5.627	1.498	1.573	0.074
6	3.677	1.092	1.164	0.072
7	6.600	1.741	1.812	0.072
8	7.683	1.865	1.943	0.078
9	9.819	2.372	2.454	0.082
10	19.189	4.615	4.683	0.068

Table 7 Comparison of a number of MVs between variable- and fixed-size blocks (16 × 16, 16 × 8, 8 × 16, and 8 × 8)

Frame order	$MV_{16\times16}$	$MV_{16\times8}$	$MV_{8\times16}$	$MV_{8\times8}$	$\sum MV_{VB}$	$MV_{fixed(8\times8)}$	$1 - \frac{\sum MV_{VB}}{MV_{fixed(8\times8)}}$
1	141	152	176	2420	2889	4800	39.8%
2	134	190	162	2400	2886	4800	39.9%
3	129	176	148	2476	2929	4800	39.0%
4	134	176	178	2396	2884	4800	39.9%
5	151	164	200	2308	2823	4800	41.2%
6	119	180	194	2416	2909	4800	39.4%
7	146	204	182	2284	2816	4800	41.3%
8	150	180	190	2300	2820	4800	41.3%
9	137	136	166	2488	2927	4800	39.0%
10	150	144	158	2436	2888	4800	39.8%

Table 8 Comparison of a number of MVs between variable- and fixed-size blocks (8 × 8, 8 × 4, 4 × 8, and 4 × 4)

Frame order	$MV_{8\times8}$	$MV_{8\times4}$	$MV_{4\times8}$	$MV_{4\times4}$	$\sum MV_{VB}$	$MV_{fixed(4\times4)}$	$1-\frac{\sum MV_{VB}}{MV_{fixed(4\times4)}}$
1	1613	1178	1408	2936	7135	19,200	62.8%
2	1657	1216	1262	2976	7111	19,200	63.0%
3	1622	1292	1298	2892	7104	19,200	63.0%
4	1635	1290	1206	3028	7159	19,200	62.7%
5	1655	1216	1364	2780	7015	19,200	63.5%
6	1693	1262	1396	2472	6823	19,200	64.5%
7	1653	1276	1250	2896	7075	19,200	63.2%
8	1675	1192	1262	2952	7081	19,200	63.1%
9	1630	1100	1300	3240	7270	19,200	62.1%
10	1656	1086	1248	3268	7258	19,200	62.2%

reduced to up to about 40% compared to the fixed-size block.

4 Conclusions

In this paper, we proposed a method of calculating the zoom ratio for the zoom motion estimation of color video by using the depth information. We also proposed a method of the zoom motion estimation for the depth video. We measured the improvement of MSEs when the proposed method was separately applied to the color and depth videos. The simulation results showed that MSE is reduced up to about 30% for the color video and 85% for the depth video. Furthermore, zoom motion estimation for variable-size block reduces a lot of the number of motion vectors.

Some of the conventional methods for zoom motion estimation determine the zoom ratio by extracting and matching object features which are robust against zooming. There are also methods for determining the zoom ratio through searching the pattern of zoom motion from the direction and size of MVs. In the other method, an optimal zoom ratio can be found through scaling a reference block in the range of possible zoom ratios. However, these conventional methods of determining the zoom ratio have a limitation of high computational complexity. On the other hand, a computation of the zoom ratio is simplified in the proposed method, since the determination of the zoom ratio is required only in the calculation of a ratio of depth values between reference and current blocks.

The motion estimation method proposed in this paper is expected to be applicable to the video coding standard. Also, a method to encode the zoom motion vector is to be studied more in the future. Further research to obtain optimal coding efficiency by considering both the number of bits for additional transmission of the zoom motion vector and the coding gain according to the reduced motion estimation difference signal is also required.

Abbreviations

BMA: Block matching algorithm; MSE: Mean square error; MV: Motion vector; SAE: Sum of absolute error; SIFT: Scale-invariant feature transform; SSE: Sum of square error; SURF: Speeded-up robust features

Acknowledgements

Not applicable.

Authors' contributions

All authors took part in the discussion of the work described in this paper. The authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The dataset used during the current study is the NYU Depth Dataset V2 [44] and is available at `https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html`.

Competing interests

The authors declare that they have no competing interests.

Received: 5 June 2019 Accepted: 3 March 2020
Published online: 16 March 2020

References

1. H.264: Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264. https://www.itu.int/rec/T-REC-H.264/en. Accessed 1 June 2019
2. I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. (Wiley, NJ, 2003)
3. T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the h.264/AVC video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
4. S. K. Kwon, A. Tamhankar, K. R. Rao, Overview of h.264/MPEG-4 part 10. J. Vis. Commun. Image Represent. **17**(2), 186–216 (2006)
5. G. J. Sullivan, J. R. Ohm, W. J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)
6. D. Patel, T. Lad, D. Shah, Review on intra-prediction in high efficiency video coding (HEVC) standard. Int. J. Comput. Appl. **132**(13), 26–29 (2015)
7. H. G. Musmann, P. Pirsch, H. Grallert, Advances in picture coding. Proc. IEEE. **73**(4), 523–548 (1985)
8. J. Jain, A. Jain, Displacement measurement and its application in interframe image coding. IEEE Trans. Commun. **29**(12), 1799–1808 (1981)

9. H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, H. Watanabe, Two-stage motion compensation using adaptive global mc and local affine mc. IEEE Trans. Circ. Syst. Video Technol. **7**(1), 75–85 (1997)
10. T. Wiegand, E. Steinbach, B. Girod, Affine multipicture motion-compensated prediction. IEEE Trans. Circ. Syst. Video Technol. **15**(2), 197–209 (2005)
11. R. C. Kordasiewicz, M. D. Gallant, S. Shirani, Affine motion prediction based on translational motion vectors. IEEE Trans. Circ. Syst. Video Technol. **17**(10), 1388–1394 (2007)
12. Y. Nakaya, H. Harashima, Motion compensation based on spatial transformations. IEEE Trans. Circ. Syst. Video Technol. **4**(3), 339–356 (1994)
13. M. Karczewicz, J. Nieweglowski, J. Lainema, O. Kalevo, in *Proceedings of First International Workshop on Wireless Image/Video Communications*. Video coding using motion compensation with polynomial motion vector fields, (1996), pp. 26–31. https://doi.org/10.1109/wivc.1996.624638
14. M. R. Pickering, M. R. Frater, J. F. Arnold, in *2006 International Conference on Image Processing*. Enhanced motion compensation using elastic image registration, (2006), pp. 1061–1064. https://doi.org/10.1109/icip.2006.312738
15. L. Li, H. Li, D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, F. Wu, An efficient four-parameter affine motion model for video coding. IEEE Trans. Circ. Syst. Video Technol. **28**(8), 1934–1948 (2018). https://doi.org/10.1109/TCSVT.2017.2699919
16. N. Zhang, X. Fan, D. Zhao, W. Gao, Merge mode for deformable block motion information derivation. IEEE Trans. Circ. Syst. Video Technol. **27**(11), 2437–2449 (2017). https://doi.org/10.1109/TCSVT.2016.2589818
17. L. Po, K. Wong, K. Cheung, K. Ng, Subsampled block-matching for zoom motion compensated prediction. IEEE Trans. Circ. Syst. Video Technol. **20**(11), 1625–1637 (2010)
18. H. S. Kim, J. H. Lee, C. K. Kim, B. G. Kim, Zoom motion estimation using block-based fast local area scaling. IEEE Trans. Circ. Syst. Video Technol. **22**(9), 1280–1291 (2012)
19. L. Superiori, M. Rupp, in *2009 10th Workshop on Image Analysis for Multimedia Interactive Services*. Detection of pan and zoom in soccer sequences based on H.264/AVC motion information, (2009), pp. 41–44. https://doi.org/10.1109/wiamis.2009.5031427
20. R. Takada, S. Orihashi, Y. Matsuo, J. Katto, in *2015 IEEE International Conference on Consumer Electronics (ICCE)*. Improvement of 8k UHDTV picture quality for H.265/HVEC by global zoom estimation, (2015), pp. 58–59. https://doi.org/10.1109/icce.2015.7066317
21. D. Shukla, R. K. Jha, A. Ojha, Unsteady camera zoom stabilization using slope estimation over interest warping vectors. Pattern Recogn. Lett. **68**, 197–204 (2015)
22. X. Shen, J. Wang, Q. Yang, P. Chen, F. Liang, in *2017 IEEE Visual Communications and Image Processing (VCIP)*. Feature based inter prediction optimization for non-translational video coding in cloud, (2017), pp. 1–4. https://doi.org/10.1109/vcip.2017.8305066
23. G. Luo, Y. Zhu, Z. Weng, Z. Li, A disocclusion inpainting framework for depth-based view synthesis. Trans. Pattern Anal. Mach. Intell. (Early Access), IEEE, 1–14 (2019). https://doi.org/10.1109/tpami.2019.2899837
24. X. Qi, Z. Liu, Q. Chen, J. Jia, in *2019 IEEE Conference on Computer Vision and Pattern Recognition*. 3D motion decomposition for RGBD future dynamic scene synthesis, (2019), pp. 7673–7682. https://doi.org/10.1109/cvpr.2019.00786
25. M. Wu, X. Li, C. Liu, M. Liu, N. Zhao, J. Wang, X. Wan, Z. Rao, L. Zhu, Robust global motion estimation for video security based on improved k-means clustering. J. Ambient Intell. Humanized Comput. **10**(2), 439–448 (2019)
26. G. Fanelli, M. Dantone, L. Van Gool, in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. Real time 3d face alignment with random forests-based active appearance models, (2013), pp. 1–8. https://doi.org/10.1109/fg.2013.6553713
27. M. Dantone, J. Gall, G. Fanelli, L. Van Gool, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Real-time facial feature detection using conditional regression forests, (2012), pp. 2578–2585
28. R. Min, N. Kose, J. Dugelay, Kinectfacedb: A kinect database for face recognition. IEEE Trans. Syst. Man Cybernet. Syst. **44**(11), 1534–1548 (2014)
29. J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. A benchmark for the evaluation of rgb-d slam systems, (2012), pp. 573–580. https://doi.org/10.1109/iros.2012.6385773
30. F. Pomerleau, S. Magnenat, F. Colas, M. Liu, R. Siegwart, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tracking a depth camera: Parameter exploration for fast ICP, (2011), pp. 3824–3829. https://doi.org/10.1109/iros.2011.6094861
31. M. Siddiqui, G. Medioni, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. Human pose estimation from a single view point, real-time range sensor, (2010), pp. 1–8. https://doi.org/10.1109/cvprw.2010.5543618
32. R. Muñoz Salinas, R. Medina Carnicer, F. J. Madrid Cuevas, A. Carmona Poyato, Depth silhouettes for gesture recognition. Pattern Recogn. Lett. **29**(3), 319–329 (2008)
33. P. Suryanarayan, A. Subramanian, D. Mandalapu, in *2010 20th International Conference on Pattern Recognition*. Dynamic hand pose recognition using depth data, (2010), pp. 3105–3108. https://doi.org/10.1109/icpr.2010.760
34. J. Preis, M. Kessel, M. Werner, C. Linnhoff-Popien, in *1st International Workshop on Kinect in Pervasive Computing*. Gait recognition with kinect (New Castle, UK, 2012), pp. 1–4
35. S. Song, J. Xiao, in *2013 IEEE International Conference on Computer Vision*. Tracking revisited using RGBD camera: Unified benchmark and baselines, (2013), pp. 233–240. https://doi.org/10.1109/iccv.2013.36
36. J. Sung, C. Ponce, B. Selman, A. Saxena, in *Workshops at the Twenty-fifth AAAI Conference on Artificial Intelligence*. Human activity detection from RGBD images, (2011), pp. 47–55
37. L. Spinello, K. O. Arras, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. People detection in RGB-D data, (2011), pp. 3838–3843. https://doi.org/10.1109/iros.2011.6095074
38. M. Luber, L. Spinello, K. O. Arras, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. People tracking in RGB-D data with on-line boosted target models, (2011), pp. 3844–3849. https://doi.org/10.1109/iros.2011.6095075
39. K. Lai, L. Bo, X. Ren, D. Fox, in *2011 IEEE International Conference on Robotics and Automation*. A large-scale hierarchical multi-view RGB-D object dataset, (2011), pp. 1817–1824. https://doi.org/10.1109/icra.2011.5980382
40. S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Wåhslén, I. Orhan, T. Lindh, in *International Conference on ICT Innovations*. Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion (Springer, 2015), pp. 99–108. https://doi.org/10.1007/978-3-319-25733-4_11
41. P. Ammirato, P. Poirson, E. Park, J. Košecká, A. C. Berg, in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. A dataset for developing and benchmarking active vision, (2017), pp. 1378–1385. https://doi.org/10.1109/icra.2017.7989164
42. M. Kraft, M. Nowicki, A. Schmidt, M. Fularz, P. Skrzypczyński, Toward evaluation of visual navigation algorithms on RGB-D data from the first- and second-generation kinect. Mach. Vis. Appl. **28**(1-2), 61–74 (2016)
43. D. S. Lee, S. K. Kwon, Intra prediction of depth picture with plane modeling. Symmetry. **10**(12), 715 (2018)
44. N. Silberman, D. Hoiem, P. Kohli, R. Fergus, in *European Conference on Computer Vision*. Indoor segmentation and support inference from RGBD images (Springer, 2012), pp. 746–760. https://doi.org/10.1007/978-3-642-33715-4_54

Publisher's Note