Signal Processing: *Image Communication* 29 (2014) 864–874

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Texture aided depth frame interpolation

Yongbing Zhang [a,*], Jian Zhang [b], Qionghai Dai [c]

[a] *Graduate School at Shenzhen, Tsinghua University, Shenzhen, China*
[b] *Department of Computer Science, Harbin Institute of Technology, Harbin, China*
[c] *TNLIST and Department of Automation, Tsinghua University, Beijing, China*

ARTICLE INFO

Article history:
Received 12 October 2013
Received in revised form
25 May 2014
Accepted 25 May 2014
Available online 23 June 2014

Keywords:
Texture aided motion estimation
Texture aided motion compensation
Depth image based rendering
Geometric mapping
View synthesis

ABSTRACT

Recent development of depth acquiring technique has accelerated the progress of 3D video in the market. Utilizing the acquired depth, arbitrary view frames can be generated based on depth image based rendering (DIBR) technique in free viewpoint video system. Different from texture video, depth sequence is mainly utilized for virtual view generation rather than viewing. Inspired by this, a depth frame interpolation scheme using texture information is proposed in this paper. The proposed scheme consists of a texture aided motion estimation (TAME) and texture aided motion compensation (TAMC) to fully explore the correlation between depth and the accompanying textures. The optimal motion vectors in TAME and the best interpolation weights in TAMC are respectively selected taking the geometric mapping relationship between depth and the accompanying texture frames into consideration. The proposed scheme is able to not only maintain the temporal consistency among interpolated depth sequence but also improve the quality of virtual frames generated by interpolated depth. Besides, it can be easily applied to arbitrary motion compensation based frame interpolation scheme. Experimental results demonstrate that the proposed depth frame interpolation scheme is able to improve the quality of virtual view texture frames in both subjective and objective criterions compared with existing schemes.

1. Introduction

In recent years, 3D video is experiencing a rapid growth in a great number of areas including 3D cinema, 3DTV and Free viewpoint video (FVV) [1], since it is able to provide audiences an immersive feeling for a real world. Due to the desirable property of enabling users to freely select their favorite viewpoints, the application of FVV receives more and more attention. In FVV, the selection of arbitrary view requires multi-view video with dense camera setting for scene capturing, which will cause a vast amount of data to be stored or transmitted to the users. To reduce the data volume in FVV caused by a large number of views, a new data format multi-view video plus depth (MVD) is proposed to enable virtual view synthesis [2]. For simplicity, the color pictures and depth of MVD are called texture and depth images in the remainder of this paper. In MVD, dynamic scenes are captured by a limited number synchronized texture and depth cameras [3,4]. Arbitrary views can be synthesized by depth image based rendering (DIBR) method [5] utilizing the captured texture video and the associated depth videos.

As an important auxiliary information in MVD, the accuracy of depth plays a critical role for the quality of synthesized view frame. However, due to the computational and physical complexities, most methods for capturing depth sequence, such as stereoscopic and range sensors based systems, can only provide information at a low frame rate, which severely limits the application of MVD [6]. Therefore, depth frame interpolation, representing

* Corresponding author.
E-mail address: ybzhang@tsinghua.edu.cn (Y. Zhang).

http://dx.doi.org/10.1016/j.image.2014.05.005
0923-5965/

the process to improve the frame rate of depth sequence by interpolating the intermediate depth frame, has become a desirable solution to increase the temporal resolution of acquired depth sequence.

Similar to texture videos, there exists a high temporal consistency between adjacent depth frames, which means the pixel intensities of the same object exhibit a high similarity. Consequently, one direct way to perform depth frame interpolation is to use the motion compensated texture frame interpolation (MCTFI) methods, which interpolate the intermediate frame along the motion trajectory. To ensure high quality of interpolated frames, two issues need to be addressed in MCTFI. The first is how to obtain reliable motion trajectories (selecting the best motion vector for each block) between the frame to-be-interpolated and the adjacent frames. The second is how to generate the pixels within the to-be-interpolated frame. These issues are usually handled by motion estimation and motion compensation, of which the brief introduction will be provided as follows.

To improve the accuracy of motion vector in MCTFI, Hann et al. [7] proposed a 3D recursive search (3DRS) algorithm, which obtains the optimal motion vector from spatial and temporally neighboring blocks recursively. Choi et al. [8] proposed a bi-directional motion estimation based on the assumption that the motion vectors referred to the forward and backward reference frames are of the same amplitude with reverse directions. Huang et al. [9,10] proposed to first perform motion estimation for each block using a small block size, followed by merging the neighboring blocks with similar motion vectors into larger blocks and re-estimating the motions for the merged block. Kang et al. [11] proposed a method to enhance the motion vector accuracy by using the bidirectional and unidirectional matching ratios of blocks in the previous and following reference frames. Wang et al. [12] explicitly incorporated both the temporal and spatial smoothness of the motion field in motion estimation process. Besides, post-processing algorithms [13,14] are also proposed to correct unreliable motion vectors after the completion of motion estimation stage.

After assigning appropriate motion vector to each block, the to-be-interpolated block can be generated by motion compensation, where the reference blocks referred to by the motion vectors in the forward and backward reference frames are weighted averaged. However, serious blocking artifacts may be observed at the block boundaries due to the arbitrary shapes of the object. To relieve such artifacts, overlapped block motion compensation (OBMC) [15] can be introduced in MCTFI. For example, Zhai et al. [16] proposed a method to suppress the blocking artifacts by positioning overlapped blocks from the previous and following frames. Although OBMC is able to generate a much smoother interpolated frame, it may result in blurring or over-smoothing artifacts in case of non-consistent motion regions since fixed weights for neighboring blocks are assigned. To get rid of this problem, an adaptive OBMC [17] was proposed to tune the weights of different blocks according to the reliability of neighboring motion vectors. In addition, auto regressive model based MCTFI [18,19] was proposed to generate the intermediate frame by a linear combination of pixels in a square neighborhood in the reference frames. Recently, a region based global and local (GL) higher-order motion estimation [20] was proposed to derive more reliable motion vectors. Besides, a multiple hypotheses (MH) motion estimation [21] was proposed to obtain more accurate motion vectors.

The aforementioned MCTFI methods can be directly utilized to perform depth frame interpolation. However, it is difficult to obtain true motion vector for depth frame, since depth frame is much smoother and lacks sufficient textures to find reliably matched blocks. To overcome this problem, [22] and [23] proposed to share motion information from the corresponding texture frames to perform motion compensation for the to-be-interpolated depth frame. Utilizing the correlation between depth frame and the corresponding texture frame, this method is able to obtain interpolated frame with high quality. However, objects with the same depth values may have different motion vectors due to the differences of textures in the corresponding texture frames, which deteriorate the quality of interpolated depth frame. Actually, compared with texture frames, depth frames indicate the geometric mapping between textures of different views in view synthesis and they are not supposed to be viewed directly. Consequently, we should not only consider the temporal consistency between interpolated depth frames but the quality of synthesized view utilizing the interpolated depth frame. Inspired by this, we propose a texture aided motion estimation (TAME) and texture aided motion compensation (TAMC), where the accompanying texture video is utilized as auxiliary information to enable the synthesized virtual view frame with high quality. In TAME, the block wise matching criterion consists of two ingredients. The first ingredient is texture/depth discrepancy between the blocks referred to by the candidate motion vector within the forward and backward reference texture/depth frames. The second ingredient represents the texture discrepancy between pixels sharing the same viewpoint with the processed depth and the pixels mapped by interpolated depths resulting from the candidate motion vector. In TAMC, the best interpolation weights are selected from a predetermined set by minimizing the texture discrepancy between the pixels within the texture frame, having the same viewpoint with the processed depth, and the pixels mapped by interpolated depths deriving from the candidate interpolation weights.

The novelty of this paper is as follows. Firstly, the texture information is utilized to select the optimal motion vector in TAME. Secondly, the texture information is employed to select the most appropriate interpolation weights during motion compensation in TAMC. Thirdly, the proposed algorithms are compatible with any existing motion compensated frame interpolation schemes. Various experimental results demonstrate that the proposed depth frame interpolation is able to improve the quality of synthesized texture frames in both objective and subjective criterion compared with the competing methods.

The remainder of this paper is organized as follows. Section 2 provides a brief introduction of view projection

in DIBR. In Section 3, the proposed TAME is investigated in detail. Section 4 provides the detailed description of TAMC followed by the experimental results given in Section 5. Finally, the conclusions are given in Section 6.

2. View projection in DIBR

In DIBR, the virtual view frames are commonly synthesized by the texture and depth images of neighboring reference views. As illustrated in Fig. 1, the texture image I_l of the virtual view $View_l$ is synthesized by the texture image I_k of reference view $View_k$. It should be noted that more than one reference view textures can be utilized (for example both the left and right view texture frames) and here we take one reference view as an example to illustrate the view projection process.

Dented by $I_k(x_k, y_k)$ the reference pixel located at (x_k, y_k), the view projection can be expressed as

$$[x_l, y_l, z_l] = A_l R_l^{-1}[R_k A_k^{-1}[x_k, y_k, 1]^T Z(x_k, y_k) + T_k - T_l], \quad (1)$$

where A_k and A_l denote the intrinsic parameters, R_k and R_l denote the extrinsic rotation parameters, T_k and T_l denote the extrinsic translation parameters, and $Z(x_k, y_k)$ denotes the depth value located at (x_k, y_k) within depth frame Z. Then, the normalized coordinate of $[x_l, y_l, z_l]$ can be represented as $(x_l/z_l, y_l/z_l)$. The disparity $d(x_k, y_k)$ between texture pixel $I_k(x_k, y_k)$ and $I_l(x_l/z_l, y_l/z_l)$ can be obtained with the coordinates of these pixels. Especially, for the rectified FVV, the parameters of the multi-view cameras are normalized. The disparity $d(x_k, y_k)$ can be calculated as

$$d(x_k, y_k) = \frac{f \times c}{Z(x_k, y_k)}, \quad (2)$$

where f represents the rectified focal length of the camera and c represents the baseline distance between adjacent view cameras. Consequently, the DIBR operator $g(x_k, y_k, Z)$ can be expressed as a function of the coordinates and depth value of each pixel in reference views, which can be expressed as

$$g(x_k, y_k, Z) = I_l(x_k - d(x_k, y_k), y_k) = I_l\left(x_k - \frac{f \times c}{Z(x_k, y_k)}, y_k\right). \quad (3)$$

It should be mentioned that throughout this paper we assume the to-be-synthesized view has the same vertical position as the reference views and hence only x-coordinates are being modified by DIBR.

Apparently, depth error results in the synthesized pixels shifting in the virtual view images. In plain regions, texture frames have similar pixel intensity, which means that depth error will not cause serious distortion in the synthesized frame. However, for the edge regions, slight depth error may cause significant distortion in the synthesized frame, since there are remarkable intensity fluctuations across edges. Inspired by this, we develop a TAME and a TAMC employing hybrid pixel intensity discrepancy considering the synthesized distortion in virtual view frame in the next two Sections.

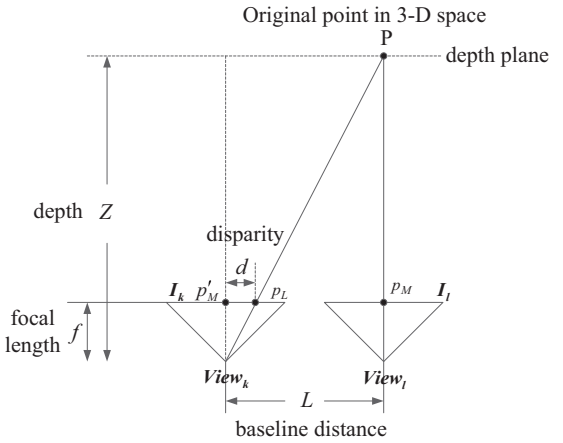

Fig. 1. View projection in DIBR.

3. Proposed TAME

The proposed depth frame interpolation scheme is illustrated in Fig. 2, which consists of forward TAME, backward TAME and TAMC. In the proposed scheme, both the reference and synthesized texture frames are available, while the corresponding depth frames have a lower frame rate compared to the associate texture frames. The goal of the proposed depth frame interpolation is to interpolate the missing depth frame to the same frame rate of the associate texture frames. To better capture the content varying property of depth frame, similar to MCTFI, the proposed depth frame interpolation is performed block by block. It should be noted that the proposed method can be also extended to region by region, similar to the work in [20]. For each block or region, both forward and backward TAME are performed to derive the forward and backward motion vectors, respectively, followed by TAMC to generate the interpolated depth frame.

To better understand the proposed TAME, we take forward TAME as an example in Fig. 3 to describe the

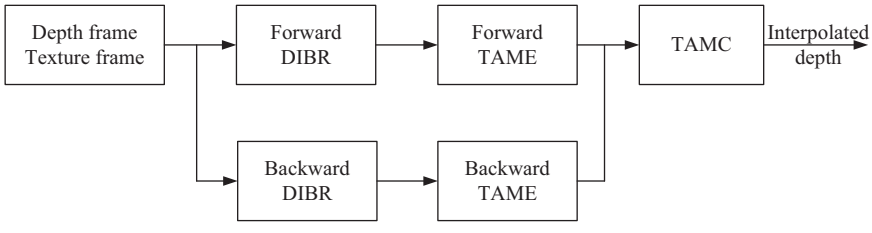

Fig. 2. Proposed depth frame interpolation.

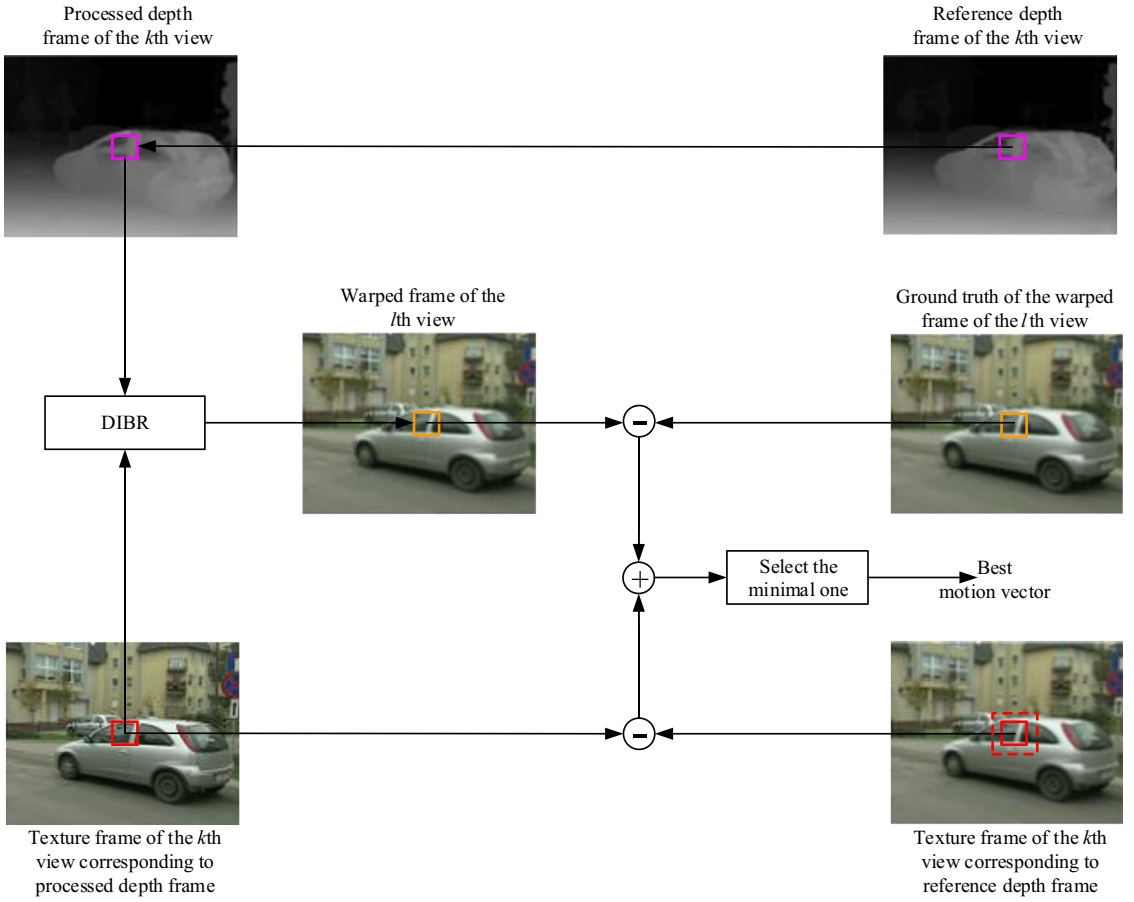

Fig. 3. Proposed forward TAME. Pink block represents the depth block being processed, red block represents the texture block corresponding to the processed depth block, the red dashed block represents the search region, and the yellow block represents pixel set of the warped pixels within the warped view. Warped pixel set can be arbitrary shapes, and here we use block shape for the sake of illustration convenience.

flowchart of TAME process. It should be noted that the process of backward TAME is quite similar by referring the backward reference frames. In order to select the best motion vector from the candidate set, in forward TAME, we introduce two distortion terms, including the distortion between corresponding texture block and the reference texture block as well as the distortion between pixel set of warped pixels and the ground truth of pixel set of warped pixels, as indicated in Fig. 3. It should be noted that the pixel set of warped pixels can be arbitrary shapes, and here we use block shape for the sake of illustration convenience. Denoted by Z_k the current processed depth block of the kth view point, as indicated the left pink block in Fig. 3, $Z_{k,f}$ the forward candidate depth block, as indicated the right pink block in Fig. 3, I_k the texture block of the kth view corresponding to processed depth block, as indicated left red block, $I_{k,f}$ the texture block of the kth view corresponding to the forward reference depth block, as indicated right red block, $\hat{I}_l$ and I_l pixel sets of the warped and ground truth of pixels within the lth view respectively, as indicated the yellow blocks in Fig. 3. Inspired by [22] and [23], we use the corresponding texture frame to search the optimal motion vector of the processed depth frame.

For each candidate motion vector $\vec{V}_c=(V_{cx},V_{cy})$, the distortion between the corresponding texture pixel and reference texture pixel can be represented as

$$D_Z(x,y,\vec{V}_c)=\|I_k(x,y)-I_{k,f}(x_f,y_f)\|^2 = \|I_k(x,y)-I_{k,f}(x+V_{cx},y+V_{cy})\|^2, \quad (4)$$

where I_k and $I_{k,f}$ represent the corresponding texture frame and forward reference texture frame of the kth viewpoint. It should be noted that other distortion norms can also be utilized in Eq. (4).

Assume the interpolated depth frames are utilized to synthesize the lth view frame, i.e., the interpolation result is fed into the view synthesis procedure as indicated in Eq. (3), the distortion between the pixel intensity within synthesized lth view frame and the corresponding pixel intensity within original lth view frame can be expressed as

$$D_I(x,y,\vec{V}_c)=\|g(x,y,Z_{k,f}(x+V_{cx},y+V_{cy}))-I_k(x,y)\|^2 = \|I_l\left(x-\frac{f\times c}{Z_{k,f}(x+V_{cx},y+V_{cy})},y\right)-I_k(x,y)\|^2, \quad (5)$$

where $Z_{k,f}(x+V_{cx},y+V_{cy})$ represents the matched depth pixel indicated by candidate motion vector $\vec{V}_c$ in the forward reference depth frame.

Taking consistency of interpolated depth frame (Eq. (4)) and the quality of synthesized view (Eq. (5)) into account,

the optimal candidate motion vector $\vec{V}_c$ should satisfy

$$\vec{V}=\arg\min_{\vec{V}_c\in\vec{R}}\left\{S(\vec{V}_c)+\sum_{(x,y)\in\Omega}D_Z(x,y,\vec{V}_c)\right\} s.t. \sum_{(x,y)\in\Omega}D_I(x,y,\vec{V}_c)<\tau, \tag{6}$$

where $\vec{R}$ represents the search range, Ω represents the domain of the processing block, and $S(\vec{V}_c)$ represents the spatial consistency penalty when selecting $\vec{V}_c$ as the best motion vector.

According to Eq. (6), the optimal $\vec{V}$ can be expressed as

$$\vec{V}=\arg\min_{\vec{V}_c\in\vec{R}}\left\{S(\vec{V}_c)+\sum_{(x,y)\in\Omega}[D_Z(x,y,\vec{V}_c)+\lambda D_I(x,y,\vec{V}_c)]\right\}, \tag{7}$$

where λ is a constant to make a tradeoff between the two error terms. Obviously, applying a proper parameter λ, Eq. (7) is able to select a motion vector which not only maintains the temporal consistency between interpolated depth frames but also ensures virtual frame with high quality.

4. Proposed TAMC

In traditional motion compensation method, the interpolated depth frame can be generated as

$$\begin{aligned}\hat{Z}_k(x,y)&=w_fZ_{k,f}(x_f,y_f)+w_bZ_{k,b}(x_b,y_b)\\&=w_fZ_{k,f}(x+V_{f,x},y+V_{f,y})+w_bZ_{k,b}(x+V_{b,x},y+V_{b,y})\\&=\frac{T_b}{T_f+T_b}Z_{k,f}(x+V_{f,x},y+V_{f,y})+\frac{T_f}{T_f+T_b}Z_{k,b}(x+V_{b,x},y+V_{b,y}),\end{aligned} \tag{8}$$

where w_f and w_b represent the forward and backward interpolation weights, $V_{f,x}$ and $V_{b,x}$ represent the best forward and backward motion vectors, T_f and T_b represent the temporal distance between the to-be-interpolated depth frame and the forward and backward reference frames. Such a method is able to achieve better performance when the object moves along one line in adjacent frames. However, the performance would get worse for the occlusion or dis-occlusion regions. To tackle this problem, a TAMC algorithm is proposed utilizing the view synthesis property of depth frame in this Section.

In this paper, the backward interpolation weight w_b is defined to be $w_b=1-w_f$. And it should be noted that other combinations of w_f and w_b can also be applied. Taking $w_b=1-w_f$ into Eq. (8), the depth interpolation can be expressed as

$$\begin{aligned}\hat{Z}_k(x,y,\vec{V},w_f)&=w_fZ_{k,f}(x+V_{f,x},y+V_{f,y})+w_bZ_{k,b}(x+V_{b,x},y+V_{b,y})\\&=w_fZ_{k,f}(x+V_{f,x},y+V_{f,y})+(1-w_f)Z_{k,b}(x+V_{b,x},y+V_{b,y}).\end{aligned} \tag{9}$$

Given a forward interpolation weight set W_s and taking the quality of virtual view frame into account, the optimal forward interpolation weight parameter under the optimal $\vec{V}$ can be selected as

$$w^*=\arg\min_{w_c\in W_s}\sum_{(x,y)\in\Omega}\|g(x,y,\hat{Z}_k(x,y,\vec{V},w_c))-I_k(x,y)\|^2. \tag{10}$$

Incorporating Eqs. (5) and (9) into Eq. (10), we have

$$\begin{aligned}w^*=\arg\min_{w_c\in W_s}\sum_{(x,y)\in\Omega}\|&I_l\\&\times\left(x-\frac{f\times c}{w_cZ_{k,f}(x+V_{f,x},y+V_{f,y})+(1-w_c)Z_{k,b}(x+V_{b,x},y+V_{b,y})},y\right)\\&-I_k(x,y)\|^2.\end{aligned} \tag{11}$$

Utilizing Eq. (11), we can obtain the optimal weight for each processing block in the depth interpolation based on the depth and texture frame contents. Giving a proper parameter set W_s, Eq. (11) is able to assign the best forward and backward interpolation weights for blocks with different characteristics adaptively. For example, Eq. (11) is able to select a relatively larger forward interpolation weight for covered regions, where the frame content is much similar to the forward reference frame. While a smaller forward interpolation weight can be selected automatically for uncovered regions, where the frame content is much similar to the backward reference frame.

5. Experimental results

In this section, various experiments are conducted to verify the superiority of the proposed depth frame interpolation. Standard test sequences in MPEG 3DV [24] are selected to carry out experiments. Every other frame of the first 100 frames in each reference view of the test sequences is dropped and then interpolated. Virtual views are synthesized by view synthesis reference software, version 3.5 (VSRS) [25] using the interpolated depth sequence and the associate texture videos. We will first give the effect of parameter λ in TAME and then the interpolation comparisons will be provided.

5.1. Effect of parameter λ in TAME

In this subsection, sequences Kendo (1024×768) and Poznan_street (1920×1088) are selected to illustrate the effect of parameter λ in the proposed TAME. In Kendo, we use view 1 and 5 to synthesize view 3, and in Poznan_street, we use view 3 and 5 to synthesize view 4. Here, we apply the similarity measure (SM) distortion [23] in proposed TAME and the search range is set to be 8. It should be noted in SM, the distortion in Eq. (4) can be rewritten as

$$D_Z(x,y,\vec{V}_c)=\frac{\min(I_k(x,y),I_{k,f}(x+V_{cx},y+V_{cy}))}{\max(I_k(x,y),I_{k,f}(x+V_{cx},y+V_{cy}))}. \tag{12}$$

and the distortion in Eq. (5) can be rewritten as

$$D_I\left(x,y,\vec{V}_c\right)=\frac{\min\left(I_k(x,y),I_l\left(x-\frac{f\times c}{Z_{k,f}(x+V_{cx},y+V_{cy})},y\right)\right)}{\max\left(I_k(x,y),I_l\left(x-\frac{f\times c}{Z_{k,f}(x+V_{cx},y+V_{cy})},y\right)\right)}. \tag{13}$$

Taking Eqs. (12) and (13) into consideration, the best motion vector can be selected as

$$\vec{V}=\arg\min_{\vec{V}_c\in\vec{R}}\left\{S\left(\vec{V}_c\right)+\sum_{(x,y)\in\Omega}D_Z\left(x,y,\vec{V}_c\right)+\lambda D_I\left(x,y,\vec{V}_c\right)\right\} \tag{14}$$

The interpolation weights of the forward and backward reference frames during motion compensation are both set to

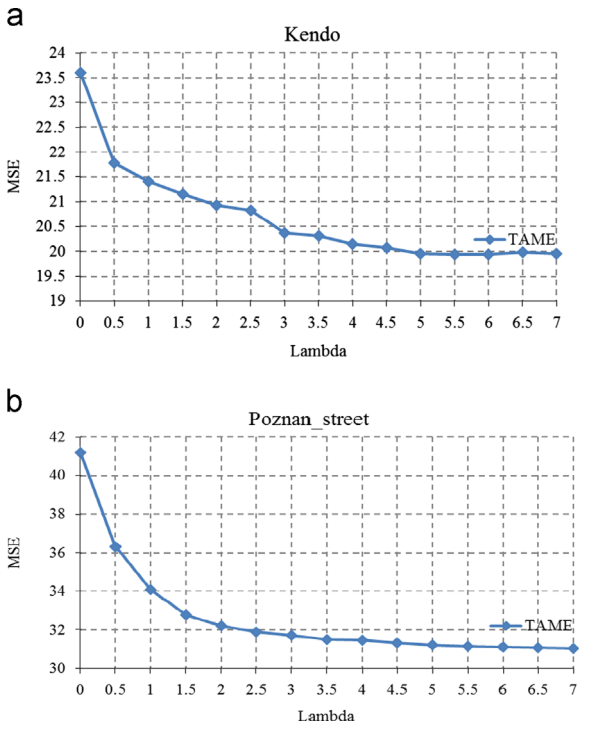

Fig. 4. The relation between parameter λ and the average MSE of synthesized frames utilizing the depth generated by the proposed method. (a)The relation between parameter λ and the average MSE of synthesized frame in Kendo utilizing the depth generated by the proposed TAME. (b)The relation between parameter λ and the average MSE of synthesized frame in Poznan_street utilizing the depth generated by the proposed TAME.

0.5. The interpolated depth of each reference view is then utilized to synthesize the virtual view using VSRS [25].

Fig. 4 depicts the relation between parameter λ and the average mean square error (MSE) of synthesized frames over each test sequence. Here, the MSE is computed between the original texture frame and the virtual one of the synthesized view. The virtual frame is synthesized by the interpolated depth frame, the accompanying texture frame as well as the corresponding camera parameters. In Fig. 4, it can be observed that the selection of λ plays a significant role on the quality of the synthesized frame. The MSE of the synthesized frame becomes smaller with the increase of parameter λ. However, when λ is larger than 5, the MSE has the tendency to be become a constant. This is because when λ is too small, the quality of synthesized texture frame plays a smaller role in the matching criterion as indicated in Eq. (7) in TAME. With the increase of λ, the proportion of synthesized frame quality in the matching criterion in Eq. (7) becomes converged. Based on such observation, λ is set to be 5 in the following experiment.

5.2. Interpolation comparisons

In this subsection, various experiments are conducted on four test sequences, whose reference views and synthesized views are depicted in Table 1. The proposed TAME and TAMC are applied to five typical frame interpolation frameworks: 3DRS [7], sum of square errors (SSE) criterion [8], the SM criterion [23], the region based global and local (GL) motion estimation [20], and the multiple hypothesis (MH) motion estimation [21], respectively. The parameter set W_s includes { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}.

Table 1
Reference and synthesized views for each test sequence.

Test sequence	Reference views	Synthesized view
Kendo 1024×768	1–5	3
Poznan_Street 1920×1088	3–5	4
Poznan_Hall 1920×1088	5–7	6
Café 1920×1080	1–5	3

The average PSNR values between the original depth frame and the interpolated one as well as the average PSNR values between the original texture frame and the synthesized one over each test sequence are tabulated in Tables 2 and 3, respectively. Here, ME represents the traditional motion estimation, of which the correlation between depth frame and the associate texture frame is neglected. It can be observed that in Table 2, there is a PSNR drop of TAME and the combination of TAME and TAMC, compared with traditional ME and motion compensation method. For example, the average PSNR drop between TAME and traditional ME can be up to 0.6 dB, 0.4 dB, 0.4 dB, 0.5 dB, and 0.4 dB, respectively over 3DRS, SSE, SM, GL and MH methods. This is because traditional ME only considers the fidelity between corresponding texture frames while neglects the quality of synthesized virtual frames.

In contrary, in Table 3, the average PSNR of TAME has a significant improvement compared with that of traditional ME method, e.g. the average PSNR gains can be up to 0.2 dB, 0.7 dB, 0.6 dB, 0.2 dB, and 0.4 dB over 3DRS, SSE, SM, GL and MH methods, respectively. Another observation is that the combination of TAME and TAMC achieves the highest performance. For example, compared with traditional ME method, the average PSNR gains can be up to 0.6 dB, 1.1 dB, 1.1 dB, 0.6 dB and 0.6 dB over 3DRS, SSE, SM, GL and MH method, respectively. Comparing the results of Tables 2 and 3, it can be concluded that the interpolated depth frames with higher PSNR values (compared with the ground truth) do not ensure the virtual frames with higher quality. This is because the traditional interpolation method neglects the geometric mapping effect of depth frame during view synthesis.

The frame by frame PSNR values between the original texture frame and synthesized one of the first 20 frames over Poznan_street and Kendo are illustrated in Fig. 5. The baseline ME method in traditional ME and TAME is 3DRS in Fig. 5. It can be observed that the combination of TAME and traditional MC outperforms the combination of traditional ME and traditional MC in terms of PSNR values for each synthesized frame. And the performance can be further improved significantly when combing TAME and TAMC. Especially for the 1st frame of Poznan_street, the highest PSNR gain of TAME+TAMC can be up to 0.7 dB and for the 5th frame of Kendo, the highest PSNR gain can be up to 0.9 dB when

Table 2
PSNR values of the original depth frame and interpolated one over each test sequence.

Sequence	ME					TAME					TAME+TAMC				
	3DRS	SSE	SM	GL	MH	3DRS	SSE	SM	GL	MH	3DRS	SSE	SM	GL	MH
Poznan_street	41.25	41.77	41.82	42.41	42.20	40.92	41.33	41.74	42.20	41.99	41.18	41.35	41.33	41.92	41.73
Kendo	32.48	32.44	32.54	32.92	32.70	31.86	31.98	32.06	32.50	32.36	31.87	31.98	31.99	32.48	32.30
Poznan_Hall	42.98	42.90	42.92	43.41	43.21	42.32	42.52	42.64	43.05	42.83	42.39	42.50	42.46	42.91	42.78
Cafe	36.66	37.24	37.65	38.09	37.80	35.92	36.76	36.86	37.25	37.08	35.77	36.66	36.74	37.17	36.92
Avg	**38.34**	**38.59**	**38.73**	**39.21**	**38.98**	**37.76**	**38.15**	**38.33**	**38.75**	**38.57**	**37.80**	**38.12**	**38.13**	**38.62**	**38.43**

Table 3
PSNR values of the original texture frame and synthesized one over each test sequence.

Sequence	ME					TAME					TAME+TAMC				
	3DRS	SSE	SM	GL	MH	3DRS	SSE	SM	GL	MH	3DRS	SSE	SM	GL	MH
Poznan_street	32.59	32.68	32.75	33.24	32.99	32.81	33.74	33.54	33.59	33.38	33.16	34.54	34.49	34.17	33.74
Kendo	34.09	34.30	34.64	35.19	34.95	34.22	35.09	35.18	35.47	35.27	34.75	35.38	35.49	35.83	35.58
Poznan_Hall	35.29	35.18	35.17	35.69	35.51	35.46	35.93	35.70	35.88	35.85	35.81	36.30	36.13	36.29	35.99
Cafe	29.17	28.72	29.24	29.84	29.74	29.34	28.93	29.63	29.78	30.10	29.62	29.17	30.01	30.13	30.10
Avg	**32.79**	**32.72**	**32.95**	**33.49**	**33.29**	**32.96**	**33.42**	**33.51**	**33.68**	**33.65**	**33.34**	**33.85**	**34.03**	**34.11**	**33.85**

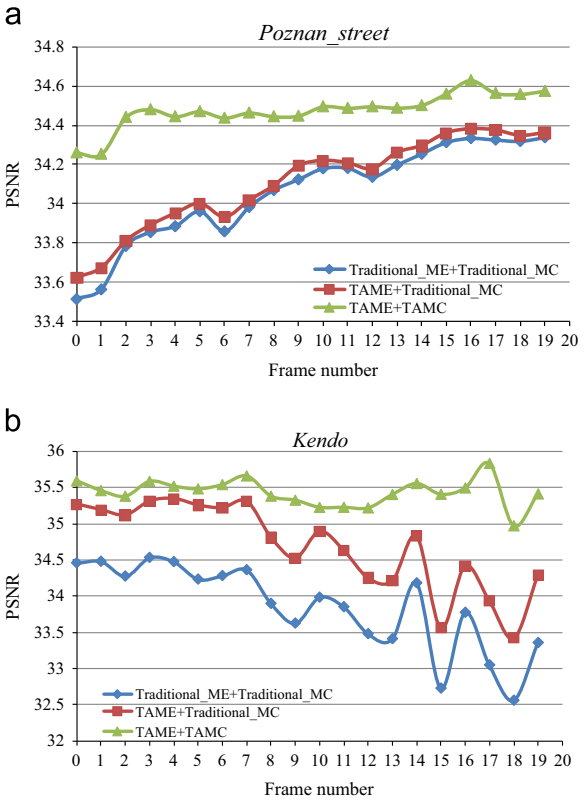

Fig. 5. The frame by frame PSNR values between the original texture frame and synthesized one of the first 20 frames over Poznan_street and Kendo. (a) The frame by frame PSNR values between the original texture frame and synthesized one of the first 20 frames over Poznan_street. (b) The frame by frame PSNR values between the original texture frame and synthesized one of the first 20 frames over Kendo.

compared with the combination of Traditional ME and Traditional MC.

Figs. 6 and 7 depict the visual comparisons between the synthesized texture frame and the interpolated depth image of the 2nd frame over Poznan_street and the 19th frame over Kendo, respectively. It should be noted that the baseline ME methods in traditional ME and TAME are SM and 3DRS schemes, respectively in Figs. 6 and 7. In Fig. 6(b) and (c), strong artifacts can be observed around the edge of windshield and significant ghost artifacts can be observed around the head of the car. In Fig. 6(d) and (e) both the blocking artifacts and ghost artifacts are removed to some extent, however they are still observable. While in Fig. 6(f) and (g), the blocking artifacts are greatly removed and the ghost artifacts cannot be observed. In Fig. 7(b), strong blocking artifacts can be observed and in Fig. 7(c) strong ghost artifacts can be observed around the edge of the wood sword. In Fig. 7(d) and (f), blocking artifacts are greatly removed and In Fig. 7(e), ghost artifacts are removed to some extent, but still observable. However, in Fig. 7(g), the ghost artifacts are removed thoroughly. It can be observed that the combination of TAME and TAMC achieves the highest visual quality among all the competing methods. This is because the proposed TAME is able to find better motion vector considering the geometric mapping effect of depth frame and the proposed TAMC can tackle the covered and uncovered regions by adaptively tuning the corresponding forward and backward interpolation weights.

Fig. 8 illustrates the regions with equal and unequal interpolation weights. It should be noted that the baseline ME method is SM in Fig. 8. Here, black regions represent the regions with equal interpolation weights (both forward and backward interpolation weights are 0.5) and white regions

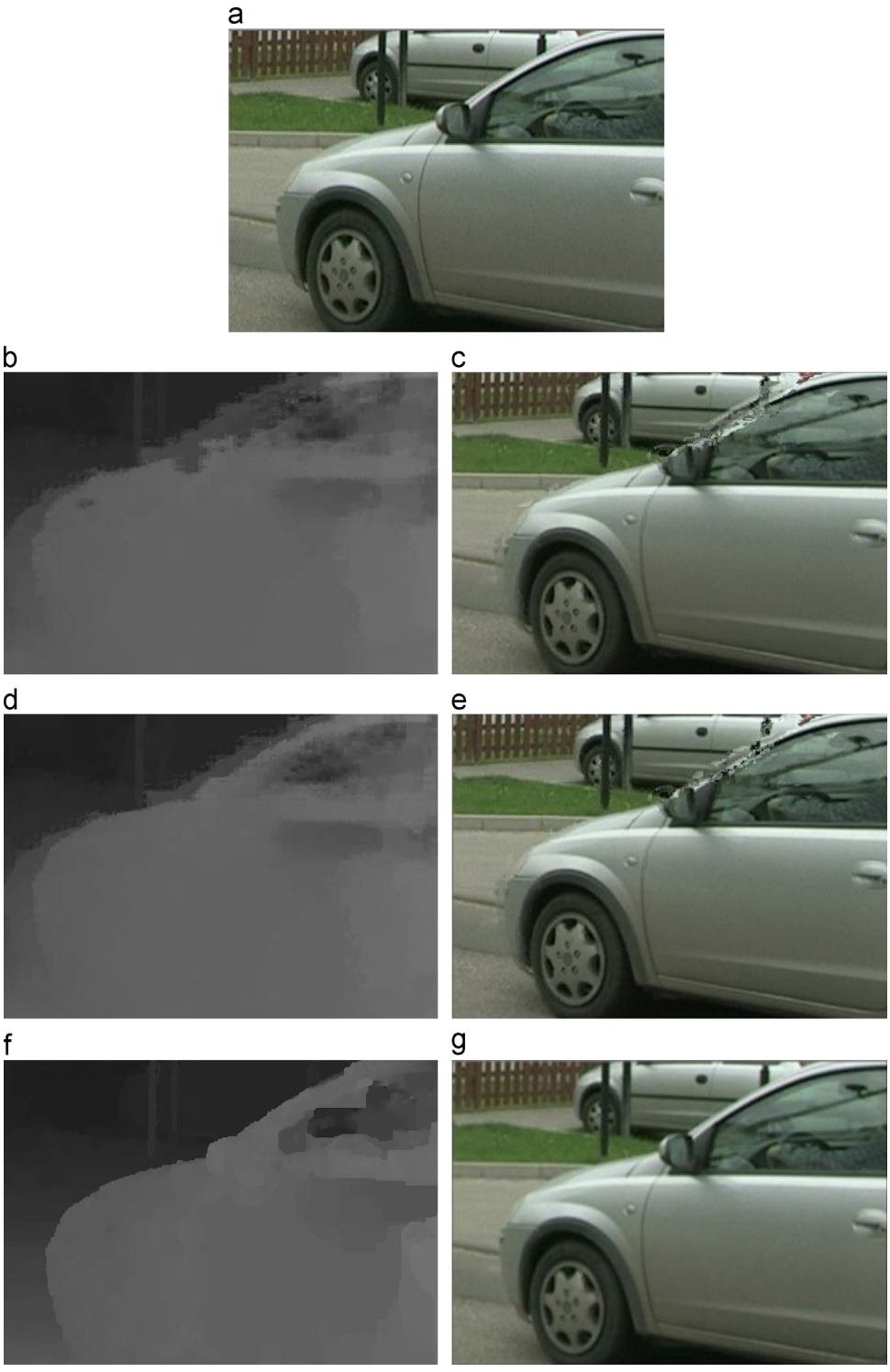

Fig. 6. Visual comparisons between the synthesized texture frame and the interpolated depth frames over Poznan_street. (a) Original texture frame, (b) Interpolated depth frame by Traditional ME + Traditional MC, (c) Synthesized result by Traditional ME+Traditional MC, (d) Interpolated depth frame by TAME+ Traditional MC, (e) Synthesized result by TAME + Traditional MC, (f) Interpolated depth frame by TAME+TAMC and (g) Synthesized result by TAME+TAMC.

represent the regions with unequal interpolation weights. As indicated in Fig. 8(b), the regions of unequal interpolation weights usually appear around the edges of objects. This is because motion compensation with equal interpolation weights may not achieve good performance around the edge regions, while the proposed TAMC is able to adaptively select

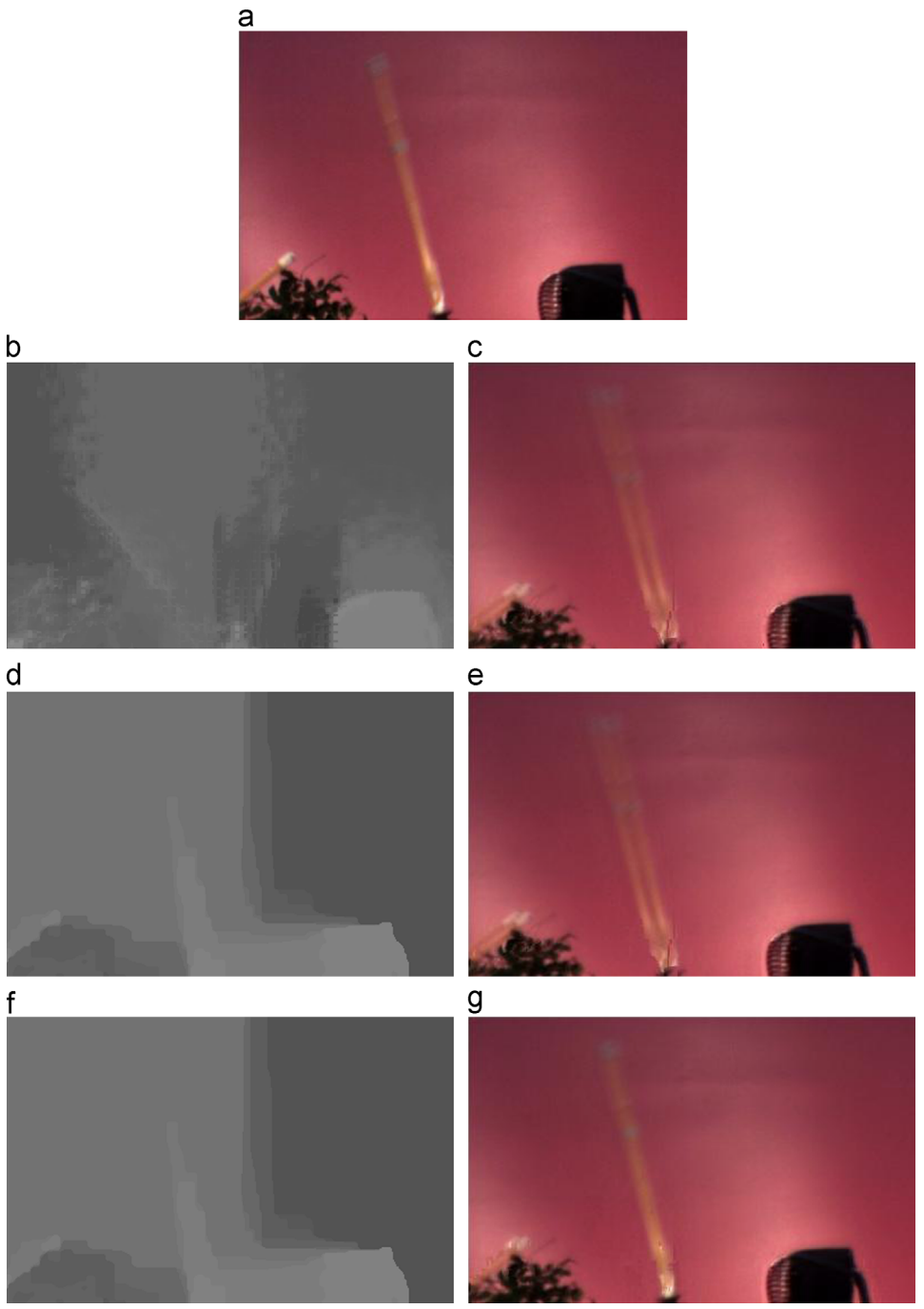

Fig. 7. Visual comparisons between the synthesized texture frame and the interpolated depth frames over Kendo. (a) Original texture frame, (b) Interpolated depth frame by Traditional ME + Traditional MC, (c) Synthesized result by Traditional ME+Traditional MC, (d) Interpolated depth frame by TAME+Traditional MC, (e) Synthesized result by TAME + Traditional MC, (f) Interpolated depth frame by TAME+TAMC and (g) Synthesized result by TAME+TAMC.

the optimal interpolation weights according to the contents of the corresponding texture regions.

5.3. Computational complexity analysis

The computational complexity of the proposed depth frame interpolation scheme mainly concentrates on the geometric mapping processing during TAME and TAMC. In TAME, for each candidate motion vector, every depth pixel within the matched depth block is first converted into a disparity vector, and then the distortion between the original and virtual texture pixel referred to by the converted disparity vector is calculated. In TAMC, for each forward and backward interpolation weight pair, the weighted depth pixel is first

Fig. 8. Illustration of interpolation regions with equal and unequal weights of the 2nd frame over Poznan_street. (a) Original texture frame. (b) Regions of equal interpolation weights and unequal interpolation weights.

Table 4
Average processing time (sec/frame) of different depth interpolation methods.

Sequence	ME			TAME			TAME+TAMC		
	3DRS	SSE	SM	3DRS	SSE	SM	3DRS	SSE	SM
Poznan_street	0.827	3.126	3.451	1.419	3.893	3.983	1.852	4.752	4.521
Kendo	0.462	1.982	2.912	1.073	2.773	3.401	1.441	3.287	4.298
Poznan_Hall	0.843	3.215	3.392	1.523	3.903	4.025	1.901	4.861	4.921
Café	0.853	3.428	3.617	1.427	4.025	4.218	1.842	5.193	5.227
Avg	**0.746**	**2.938**	**3.343**	**1.361**	**3.649**	**3.907**	**1.759**	**4.523**	**4.742**

converted into a disparity vector, and then distortion between the original texture pixel and virtual pixel is calculated.

Table 4 provides the average processing time of different depth interpolation methods, which are implemented in C/C++, on a typical computer (2.5 GHz Intel Dual Core, 4 GB Memory). It can be observed that traditional motion estimation and motion compensation method has the smallest average processing time. The average

processing time of TAME is larger than that of ME, since extra geometric mapping and distortion calculation processing is introduced. The combination of TAME and TAMC has the largest average processing time, since geometric mapping and distortion calculation processing is also implemented in TAMC. However, the average processing time can be lowered by looking for fast optimization algorithms or utilizing more and more powerful computers.

6. Conclusions

In this paper, we proposed a texture aided depth frame interpolation using texture information, based on the fact that depth frame is utilized for view synthesis rather than viewing. Considering the geometric mapping function of depth during view synthesis, a texture aided motion estimation (TAME) is devised. For each candidate motion vector, the matching criterion consists of two parts. The first part is the distortion between the current and forward/backward matched texture block referred to by candidate motion vector. The second one is the distortion between the reference texture frame and the synthesized texture frame resulted from matched depth block. In addition, to further improve the quality of interpolated depth, this paper proposes a texture aided motion compensation (TAMC), where optimal interpolation weights are selected taking into account the correlation between depth and the accompanying texture frames. The proposed TAME and TAMC are able to apply into any existing motion compensated texture frame interpolation. Experimental results verify the superiority of the proposed depth frame interpolation scheme.

Further research could look into depth frame interpolation algorithm incorporating the hole filing and boundary noise removing in view synthesis. As FVV receives more and more attention, we trust the proposed depth frame interpolation would be beneficial for the development of 3D video services.

Acknowledgment

This work was partially supported by National Natural Science Foundation of China (61170195), the Joint Funds of National Science Foundation of China (U1301257), the Upgrading Project of Shenzhen Key Laboratory (JCYJ20130402164013917).

References

[1] M. Tanimoto, Overview of free viewpoint television, Signal Process.: Image Commun. 21 (6) (2006) 454–461.
[2] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, T. Wiegand, Depth image-based rendering with advanced texture synthesis for 3-D video, IEEE Trans. Multimed. 13 (2011) 453–465.
[3] A. Smolic, P. Kauff, Interactive 3D video representation and coding technologies, in: Proceedings of IEEE Special Issue on Advances in Video Coding and Delivery, vol. 93, no. 1, January 2005, pp. 98–110.
[4] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, C. Zhang, Multiview imaging and 3DTV, IEEE Signal Process. Mag. 24 (6) (2007) 10–21.
[5] H. Sawhney, Y. Guo, K. Hanna, R. Kumar, Hybrid stereo camera: an IBR approach for synthesis of very high resolution stereoscopic image sequences, in: Proceedings of SIGGRAPH, 2001, pp. 451–460.
[6] Y. Li, L.I. Sun, T. Xue, Fast frame rate up-conversion of depth video via video coding, ACM Multimed. (2011) 1317–1320.
[7] G. Haan, P. Biezen, H. Huijgen, O. Ojo, True motion estimation with 3-D recursive search block matching, IEEE Trans. Circuits Syst. Video Technol. 3 (5) (1993) 368–379.
[8] B. Choi, S. Lee, S. Ko, New frame rate up-conversion using bi-directional motion estimation, IEEE Trans. Consumer Electron. 46 (3) (2000) 603–609.
[9] A. Huang, T. Nguyen, A multistage motion vector processing method for motion-compensated frame interpolation, IEEE Trans. Image Process. 17 (5) (2008) 694–708.
[10] A. Huang, T.Q. Nguyen, Correlation based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation, IEEE Trans. Image Process. 18 (4) (2009) 740–752.
[11] S. Kang, S. Yoo, Y. Kim, Dual motion estimation for frame rate up-conversion, IEEE Trans. Circuits Syst. Video Technol. 20 (12) (2010) 1909–1914.
[12] D. Wang, L. Zhang, A. Vincent, Motion-compensated frame rate up-conversion—Part I: fast multi-frame motion estimation, IEEE Trans. Broadcast. 56 (2) (2010) 133–141.
[13] R. Castagno, P. Haavisto, G. Ramponi, A method for motion adaptive frame rate up-conversion, IEEE Trans. Circuits Syst. Video Technol. 6 (5) (1996) 436–446.
[14] G. Dane, T. Nguyen, Motion vector processing for frame rate up conversion, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Quebec, Canada, 2004.
[15] M. Orchard, G. Sullivan, Overlapped block motion compensation: an estimation-theoretic approach, IEEE Trans. Image Process. 3 (5) (1994) 693–699.
[16] J. Zhai, K. Yu, J. Li, S. Li, A low complexity motion compensated frame interpolation method, in: Proceedings of ISCAS, vol. 5, May 2005, pp. 4927–4930.
[17] B. Choi, J. Han, C. Kim, S. Ko, Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation, IEEE Trans. Circuits Syst. Video Technol. 17 (7) (2007) 407–416.
[18] Y. Zhang, D. Zhao, X. Ji, R. Wang, W. Gao, A spatio-temporal auto-regressive model for frame rate up conversion, IEEE Trans. Circuits Syst. Video Technol. 19 (9) (2009) 1289–1301.
[19] Y. Zhang, D. Zhao, S. Ma, R. Wang, W. Gao, A motion-aligned auto-regressive model for frame rate up conversion, IEEE Trans. Image Process. 19 (5) (2010) 1248–1258.
[20] C. Qian, I. Bajic, Frame rate up-conversion using global and local higher-order motion, in: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), July 2013, pp. 1–6.
[21] H. Liu, R. Xiong, D. Zhao, S. Ma, W. Gao, Multiple hypotheses Bayesian frame rate up-conversion by adaptive fusion of motion compensated interpolations, IEEE Trans. Circuits Syst. Video Technol. 22 (8) (2012) 1188–1198.
[22] J. Choi, D. Min, B. Ham, K. Sohn, Spatial and temporal up-conversion technique for depth video, in: Proceedings of IEEE International Conference on Image Process. (ICIP), November 2009, pp. 3525–3528.
[23] H. Wang, C. Huang, J. Yang, Block-based depth maps interpolation for efficient multiview content generation, IEEE Trans. Circuits Syst. Video Technol. 21 (12) (2011) 1847–1858.
[24] Video and Requirements, Applications and Requirements on 3D Video Coding, N12035, ISO/IEC JTC1/SC29/WG11, Switzerland, Geneva, March 2011.
[25] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, Y. Mori, Reference Softwares for Depth Estimation and View Synthesis, ISO/IEC JTC1/SC29/WG11/M15377, April 2008.