Signal Processing: Image Communication 54 (2017) 11–22

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Two-stage filtering of compressed depth images with Markov Random Field

Lijun Zhao[a], Huihui Bai[a,*], Anhong Wang[b], Yao Zhao[a], Bing Zeng[c]

[a] Institute Information Science, Beijing Jiaotong University, No. 3 Shangyuancun, Haidian District, China
[b] Institute of Digital Media and Communication, Taiyuan University of Science and Technology, No. 66 Waliu Road, Wanbailin District, Taiyuan, China
[c] Institute of Image Processing, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

ARTICLE INFO

Keywords:
MRF
Binary segmentation
Filtering
Compression
Depth image
Distortion

ABSTRACT

Virtual view synthesis and image comprehension have become easier with the aid of depth information. However, when a depth image is compressed, severe distortions along boundaries may occur, thus leading to performance degradation. To solve this problem, we propose in this paper a two-stage filtering that consists of binary segmentation-based depth filtering and the reconstruction using a Markov Random Field (MRF) model. The MRF model adopted in our work consists of a data term and a smoothness term so as to preserve the boundary and maintain the smoothness simultaneously. We notice that directly applying the MRF model to a distorted depth image is usually unable to produce a satisfactory performance. Then, we propose that binary segmentation based depth filtering is used to remove artifacts over discontinuous regions in the distorted depth image. Experimental results show that, through our processing, the compressed depth image can render better quality for the synthesized images than many existing depth filtering methods.

1. Introduction

Multiview videos with associated depth map are a widely-adopted format for representing 3D video contents, owing to its ability of rendering virtual views, extracting foreground, understanding contents, etc [1]. With a limited band-width, videos need to be compressed, including the depth video. However, any coding inevitably degrades image's quality. In the 3D coding system, when a depth video is encoded by methods such as HEVC with relatively large quantization step-sizes, severe distortions over discontinuous regions often occur [2,3], such as blurring and some noisy ambiguity pixels, which may affect both accuracy of depth image itself and quality of the synthesized views in stereoscopic video applications. It is widely known that depth filtering techniques could improve the quality of coded depth images.

Recently, depth filtering has become a research issue in both depth measurement [4–8] acquired by a depth camera (such as Kinect and Time of Flight), and 3D video coding system [9–13]. Several novel filtering approaches have been explored to address different depth contaminations. Generally, filtering methods for compressed depth images can be linear versus non-linear and local versus global. It's well-known that linear filters have a drawback that edges in a depth image may get over-smoothed when filter's window-size becomes large. On the contrary, non-linear filtering can preserve edges through non-linear manipulations [14,15]. For instance, Silva et al. proposed an adaptive bilateral filter with adaptive filtering parameters for preserving edges by means of a non-linear combination of nearby pixel values based on both spatial distance and pixel similarity [9]. Similar to this method, Liu et al. proposed a trilateral filtering, which considers spatial correlation as well as luminance similarity of both depth images and color images [10]. In the meantime, Oh et al. proposed a depth boundary reconstruction filter and utilized it as an in-loop filter to code depth videos [11], and Xu et al. presented a low complexity adaptive depth truncation filter in which all edge pixels are replaced by a mean value in each block [12]. Although Xu's method can greatly reduce the artifacts of compressed depth image and achieve fast filtering, such a direct region-based replacement often leads to some distortions in non-flat regions, such as slop or curved surfaces. In our former work, we proposed a fast candidate values based boundary filtering method by utilizing spatial correlation and statistic property of local windows [13]. All these methods belong to the category of local filtering, which often corresponds to window-based operation with the correlation between window-centered pixels neglected. Different from these methods, a global method was proposed to solve the problem of generating high-resolution range images by Markov Random Field (MRF) models with low-resolution range images and registered high-resolution camera images [4,5]. Compared with local filtering methods without making filtered neighbouring pixels associated and consistency, the global method can usually provide better filtering performance because global optimization has been taken into consideration, which refers to mutual impacts between pixels during the filtering.

* Corresponding author.
E-mail address: hhbai@bjtu.edu.cn (H. Bai).

http://dx.doi.org/10.1016/j.image.2017.02.009
Received 23 September 2016; Received in revised form 22 February 2017; Accepted 23 February 2017
Available online 24 February 2017

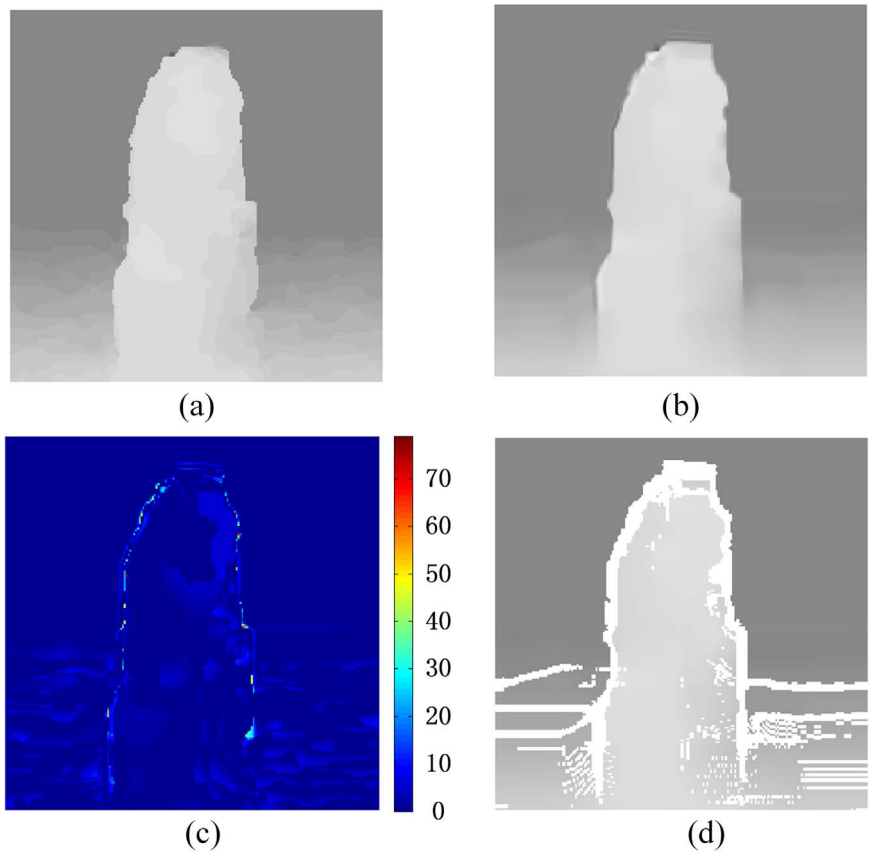

Fig. 1. (a) Part of the original depth image Book_Arrival; (b) compressed by HEVC with QP = 41; (c) coding errors; and (d) detected reliable and unreliable pixels from (b).

Because of MRF's superiority in producing more robust reconstruction against noises, such as coding artifacts or some ambiguity pixels, we attempt to use an MRF model in this paper to resolve the filtering problem of compressed depth images. According to the MRF model, reconstructed image can be obtained by taking the distorted image as prior information where pixels in smooth regions (regarded as reliable pixels) are used as measured depth values. However, directly applying the MRF model to a compressed depth image may lead to loss of some structural information in discontinuous regions in which pixels are usually regarded as un-reliable ones. To overcome this problem, we propose a two-stage filtering, denoted as TSF in this paper, where the distorted depth image is firstly dealt by binary segmentation-based depth filtering to get more reliable pixels. Note that this pre-filtering can well retrieve object boundaries with low-complexity. Then, an MRF-based reconstruction is carried out to maintain object surface smoothness and boundary sharp changes that are inherent characteristics of depth images, i.e., continuity within individual objects and discontinuity between adjacent objects. Although color guided filtering is meaningful, this paper concentrate on single depth filtering without color information, because color image couldn't always be provided in some cases.

2. Proposed method

2.1. Detection of reliable and unreliable regions

We employ an MRF model to filter the distorted depth image with binary segmentation based filtering as the pre-processing. First, we detect reliable and unreliable regions so that only unreliable pixels will be processed by binary segmentation-based filtering. To this end, a cross mask $\Psi(x, y)$ is formed around pixel $I(x, y)$: itself and 4 nearest neighbors. All five pixels are defined as reliable when the absolute difference between $I(x, y)$ and each neighbouring pixel is less than or equal to a threshold λ (empirically set to 1). After all reliable pixels are detected, we collect them to form reliable region $\Re$, whereas the rest forms the unreliable region $\Im$. One example is shown in Fig. 1, in which the white areas shown in Fig. 1(d) represent the unreliable region. It can be seen from Fig. 1 that the unreliable region occupies only a very small percentage. We propose to apply binary segmentation-based filtering only on pixels in the unreliable region so as to achieve an efficient pre-filtering.

2.2. Binary segmentation based depth filtering

The binary segmentation-based filtering (Denoted as BSF) is carried out over un-reliable pixels only, which can be regarded as a prefiltering to remove artifacts and ambiguity pixels, which is usually caused by quantization in the DCT-based frequency domain [3]. To this end, we first obtain the boundary through a binary classifier, in which three different methods including mean based classifier, median based classifier [12,13], and Otsu's method [16] are investigated. In order to compare with mean based binary segmentation strategy of method [13], the filtering results of three methods are all presented in the experimental section.

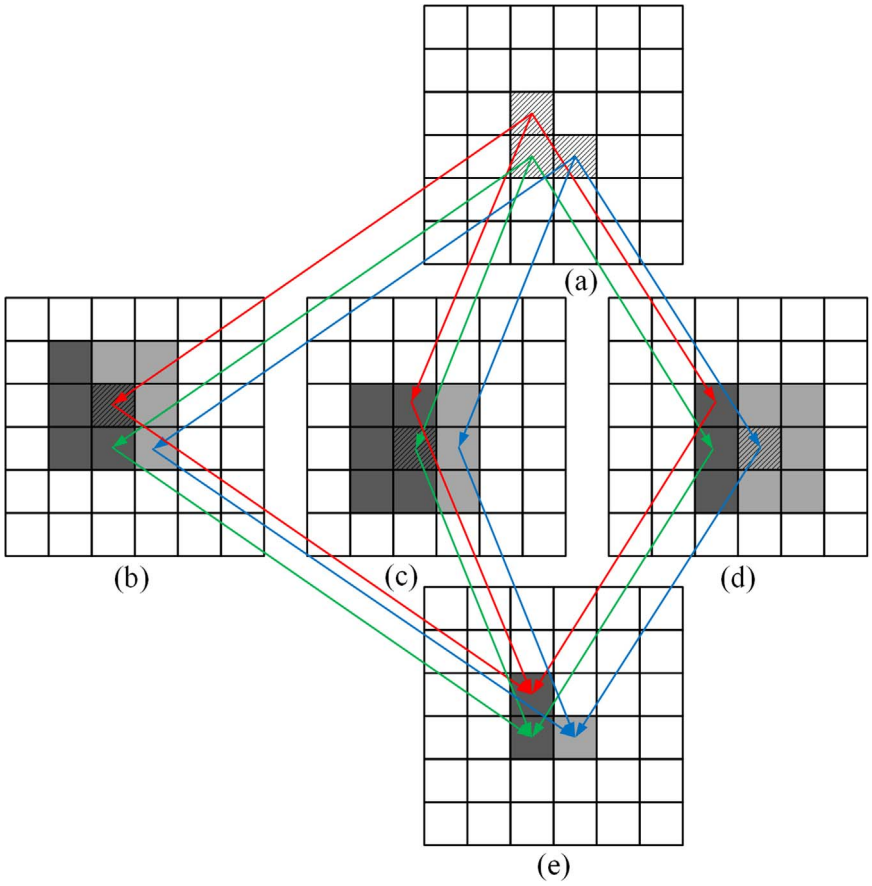

Fig. 2. Illustration of proposed BSF: (a) three unreliable pixels are detected and marked with diagonal lines while other pixels form the reliable region; (b) window w(3,3) is centered at the un-reliable pixel I(3,3); (c) window w(4,3) is centered at I(4,3); (d) window w(4,4) is centered at I(4,4); and (e) the filter output is computed by another average.

Here, binary segmentation based filtering with mean value is presented in the following as example. Specifically, a window is formed for each un-reliable pixel with its surrounding neighbors. The mean value is calculated and used for classification. An example is shown in Fig. 2. Here, three unreliable pixels, marked with diagonal lines in Fig. 2(a), are detected in the image $I$of size 6·6, while other pixels in Fig. 2(a) form reliable region $\Re$. For the un-reliable pixel $I(3, 3) \in \Im$ in Fig. 2(b), we form a window of size $(2L + 1)\cdot(2L + 1)$ centered at $I(3, 3)$ ($L=1$ in this example), denoted as $w(3, 3)$. Pixels in this window are classified into the foreground region F (with lighter filling) and the background region B (with darker filling) by comparisons with the mean value of this window. We then compute the mean values of F and B, denoted as m_F and m_B, respectively. We finally assign these two mean values to pixels in the corresponding regions, respectively, i.e., $I^{w(3,3)}(2, 3) = I^{w(3,3)}(2, 4) = I^{w(3,3)}(3, 4) = I^{w(3,3)}(4, 4) = m_F$ and $I^{w(3,3)}(2, 2) = I^{w(3,3)}(3, 2) = I^{w(3,3)}(3, 3) = I^{w(3,3)}(4, 2) = I^{w(3,3)}(4, 3) = m_B$.

Windows can be formed in the same way for the un-reliable pixels $I(4, 3)$ and $I(4, 4)$, as shown in Fig. 2(c) and (d), respectively. The corresponding classification, mean-value calculation and assignment can be performed accordingly. Note that multiple mean values will be generated for each un-reliable pixel in the end because the window moves by one pixel only at a time so that each pixel occurs in multiple windows. Finally, the filtering output at each un-reliable pixel is obtained by another average of these mean values.

The case of median based classifier for filtering with binary segmentation is analogous to the described above mean based case. For Ostu's case [16], this method assumes that image block with bi-modal histogram has two pixel's classes, which is foreground pixels set and background pixels set. Then, following the principle of intra-class variance is minimal and inter-class variance is maximal, the optimum threshold is calculated to classify pixels to be one of two classes. After Ostu's binary segmentation of block, we assign the respective class's median value of two classes as the pixel's value. For each unreliable pixels, multiple median values will be generated and the filtering output at each un-reliable pixel is obtained by average of these median values. To better discriminate three binary segmentation based filtering methods, the differences between these method would be given. The first difference between binary segmentation based filtering with Ostu's method and other binary segmentation based depth filtering is how to get the block binary segmentation, so the adaptive threshold for binary segmentation is the key. The second difference lies in whether mean value or median value of binary segmentation's corresponding region is chosen as the value of binary segmentation's corresponding region. Except these two aspects, the left operation for other binary segmentation based depth filtering is similar. For the convenience of latter discussion, the first method is binary segmentation based filtering with

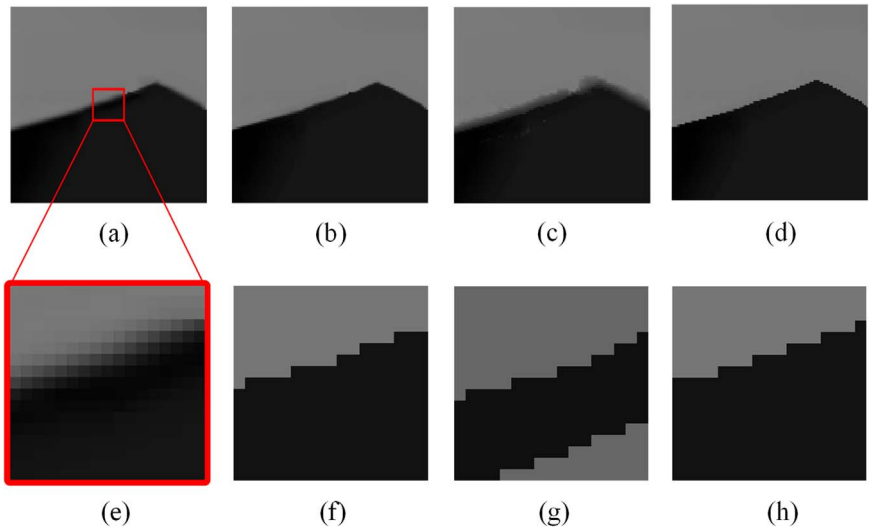

Fig. 3. The difference demonstration of three binary segmentation methods. (a) The compressed depth image, (b–d) the mean based classifier, median based classifier, and Ostu based classifier for the first stage filtering, (e) the enlargement of red box in (a), (f–h) the dealt images for (e) after assignment of foregrond's value and background's value.

mean value as segmentation threshold value and its corresponding mean value as its assigned value, denoted as BSF1. The second method is binary segmentation based filtering with mean value as segmentation threshold value and its corresponding median value as its assigned value, which is referred to BSF2. The third method is filtering with Otsu's method as segmentation method and its corresponding segmentation median value as its assigned value, which is named after BSF3.

But in fact, median based classifier is not suitable for the first stage filtering. The purpose of binary segmentation is to discriminate foreground region from background region. In Fig. 3, we give an example to explain the reasons for why not using median based classifier in the first stage. From Fig. 3 of (g), it can been seen that wrong foreground value assignment for background regions occur owing to wrong segmentation based on median, which finally leads to more blurring boundaries of depth images, while mean based classifier and Ostu based classifier don't have this problem in the most case and can well reduce the artifacts over discontinuous regions.

2.3. Depth reconstruction with MRF

The MRF model is an efficient way to connect measured values with the estimated ones. In this paper, all reliable pixels are treated as measured values and referred to as the given values. The reconstructed vector $f^* = (f_1, \ldots, f_n) \in \aleph$ where n is the total number of pixels of image F^*, is required to be estimated from the measured values. In addition, compressed depth image is regarded as prior information to constrain the reconstruction so as to maintain object's surface smoothness and preserve discontinuity between neighbouring objects. It is important to note that decoded depth image cannot be utilized directly as prior information because distortion along boundary is usually large. Therefore, the BSF presented above is firstly applied to the distorted depth image to retrieve the boundary information and remove some noisy pixels. The resulted depth image I^F is then used as prior information to get reliable region $\Re_1$. Note that this reliable region $\Re_1$ can be different from the one used earlier during BSF. Here, we set the threshold to 3 for reliable region detection to get MRF's measured values, because the reconstruction quality would usually become better with an increased number of measured values. As shown in Fig. 4, the influence of different detection thresholds of reliable region for $\Re_1$ is apparent, from which we can see that the threshold of 3 is a better value than 1. This comes from the statistic result of reliable pixel's percentage in the whole image, which will be discussed later. Another reason is that we need to ensure at least one reliable pixel in each object. Otherwise, if no reliable pixels are detected, some necessary structural information of the object would disappear totally so that the MRF model becomes inapplicable.

The probability distribution of the estimated depth image can be modeled with the following density function:

$$P(f^*|\Re_1, I^F) = \frac{\exp - E(f|\Re_1, I^F)}{\int_{\aleph} \exp - E(f|\Re_1, I^F) df}, \tag{1}$$

where f^* is the pixel to be reconstructed, and $E(f|\Re_1, I^F)$ is the energy function, consisting of a data term and a smooth term, defined as follows:

$$E(f|\Re_1, I^F) = E_{data}(f, \Re_1) + \alpha E_{smooth}(f, I^F). \tag{2}$$

The goal of data term is to regress the reconstructed depth pixels from the reliable pixels (defined by the given depth pixels), which is calculated as

$$E_{data}(f, \Re_1) = \sum_{i \in \Re_1} (f_i - R_i)^2 \tag{3}$$

$$R_i = \begin{cases} I^F(i); & if\ i \in \Re_1 \\ 0; & otherwise \end{cases} \tag{4}$$

whereas the smoothness term puts emphasis on the correlation between the pixel $I^F(i)$ and its neighbouring pixels, which is computed as:

$$E_{smooth}(f, I^F) = \sum_{i \in \Re_1} \sum_{j \in \Omega_i} \alpha \cdot w_{ij} \cdot \|(f_i - f_j)\|_2^2 \tag{5}$$

where

$$w_{ij} = \exp\left(-\frac{1}{2\sigma^2} \cdot \|I^F(i) - I^F(j)\|_2^2\right), \tag{6}$$

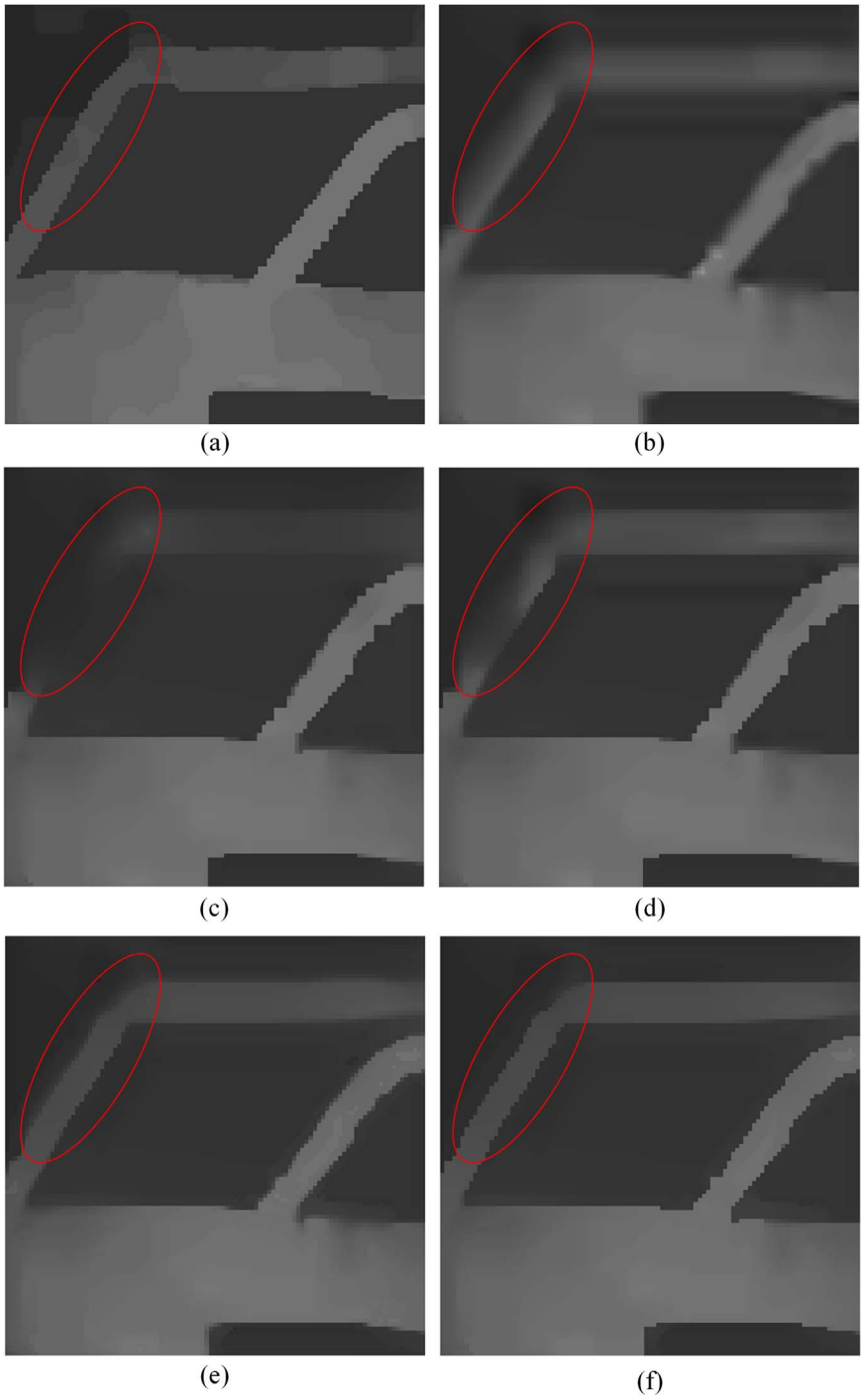

Fig. 4. (a) Part of a ground truth depth image; (b) compressed image, (c) reconstructed by MRF with λ set to 1; (d) reconstructed by MRF with λ set to 3; (e) BSF, and (f) TSF.

$\|\cdot\|_2$ is 2-norm, Ω_i denotes the neighbouring pixels of $I^F(i)$, α is a balance factor between the data term and the smoothness term, and σ^2 is the variance of the Gaussian function in Eq. (6), respectively.

To simplify, we rewrite Eq. (2) in the form of matrix as follows:

$$E(f|\Re_1, I^F) = f^T P^T P f - 2R^T P^T P f + R^T P^T P R + f^T W^T W f \tag{7}$$

$$W_{ij} = \alpha \cdot w_{ij} \tag{8}$$

$$P_{ii} = \begin{cases} 1; & if\ i \in \Re_1 \\ 0; & otherwise \end{cases} \tag{9}$$

To estimate the depth image F^* from I^F, we solve the maximum a posteriori probability (MAP) inference problem, which is equivalent to

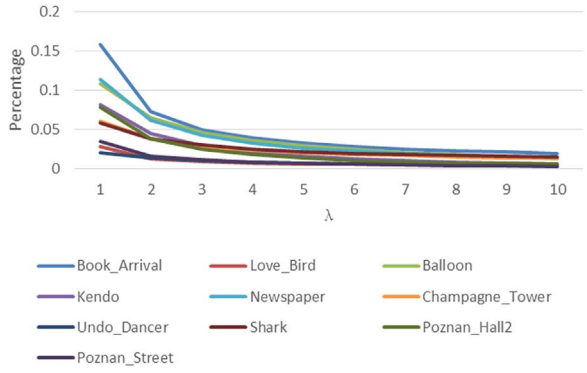

Fig. 5. The percentage of unreliable pixels in the whole image: The x-axis is the value λ ranging from 1 to 10; the y-axis is the percentage of unreliable pixels.

the minimization of energy function in Eq. (10).

$$f^* = \underset{f}{\operatorname{argmin}} E(f|\Re_1, I^F) \tag{10}$$

Since Eq. (7) is linear and convex, a global minimization can be achieved, with the closed-form solution given below:

$$f^* = A^{-1}b;\ A = P^TP + W^TW;\ b = P^TPR. \tag{11}$$

3. Experimental results and analysis

3.1. Experimental setting

Eight standard test sequences (the first 100 frames in each sequence) with the format of multi-view videos plus depth are selected in our experiments: Love_Bird (Denoted as L) (View 06 and View 08) from ETRI, Newspapers (N) (View 02 and View 04) from GIST [17] Book_Arrival (B) (View 08 and View 10) from HHI [18], Champagne_Tower (C) (View 37 and View 39) provided by Nagoya University [19], Undo_Dancer (U) (View 01 and View 05) from Nokia [20], Shark (S) from NICT [21] and Poznan_Hall2 (PH) and Poznan_Street (PS) from

Table 1
The objective quality measure of virtual view images synthesized with one/two given neighbouring views for different sequences when QP = 43.

	B	C	U	L	N	PH	PS	S	Gain
S/M(I)									
Coded	49.35	43.28	48.85	45.56	44.79	42.95	44.99	44.38	—
JTF [10]	49.51	43.25	49.35	45.74	44.88	43.12	45.22	44.56	0.18
DBRF [11]	49.50	43.53	50.01	46.39	44.63	42.87	45.35	44.61	0.34
ADTF [12]	49.93	43.58	50.08	46.23	45.05	42.93	45.25	45.82	0.59
CVBF [13]	50.24	43.74	50.35	46.57	45.24	43.02	45.30	46.08	0.80
TSF1(SLE)	50.24	43.73	49.90	46.35	45.21	42.92	45.03	45.37	0.57
TSF2(SLE)	50.46	43.55	49.71	46.16	45.07	43.02	45.18	45.92	0.61
TSF3(SLE)	50.66	43.85	50.46	46.70	45.35	43.08	45.37	46.64	0.99
TSF1(FGS)	50.37	43.60	50.22	46.46	45.15	42.98	45.20	45.68	0.69
TSF2(FGS)	50.58	43.78	50.37	46.60	45.24	43.05	45.18	46.25	0.86
TSF3(FGS)	50.63	43.82	50.61	46.79	45.29	43.10	45.32	46.66	**1.01**
S/M(II)									
Coded	50.59	45.88	49.10	48.16	47.28	41.93	47.95	47.25	—
JTF [10]	50.72	45.95	49.37	48.18	47.35	41.64	48.18	47.40	0.08
DBRF [11]	51.30	46.19	49.97	48.95	46.82	41.49	48.18	47.22	0.25
ADTF [12]	51.43	46.33	50.36	48.83	47.42	41.57	48.22	48.94	0.62
CVBF [13]	52.05	46.56	50.48	49.27	47.60	41.67	48.27	49.17	0.86
TSF1(SLE)	51.63	46.61	50.07	48.91	47.64	41.52	48.17	48.72	0.64
TSF2(SLE)	51.89	46.72	49.98	48.98	47.68	41.58	48.23	48.82	0.72
TSF3(SLE)	52.26	46.81	50.28	49.28	47.71	41.63	48.38	49.64	0.98
TSF1(FGS)	51.95	46.53	50.41	49.18	47.60	41.56	48.19	49.05	0.79
TSF2(FGS)	52.23	46.73	50.33	49.22	47.61	41.63	48.29	49.33	0.90
TSF3(FGS)	52.24	46.71	50.43	49.37	47.60	41.67	48.37	49.67	**0.99**

Table 2
The objective quality measure of virtual view images synthesized with one/two given neighbouring views for different sequences when QP = 41.

	B	C	U	L	N	PH	PS	S	Gain
S/M(I)									
Coded	50.17	44.09	49.56	46.47	45.44	43.30	45.54	45.40	—
JTF [10]	50.35	44.04	50.20	46.72	45.51	43.49	45.79	45.63	0.22
DBRF [11]	50.15	44.25	50.92	47.11	45.04	43.20	45.88	45.37	0.24
ADTF [12]	50.86	44.47	51.08	47.18	45.70	43.26	45.76	46.90	0.65
CVBF [13]	51.02	44.57	51.24	47.43	45.76	43.41	45.82	47.11	0.80
TSF1(SLE)	51.16	44.57	50.85	47.32	45.79	43.25	45.58	46.56	0.64
TSF2(SLE)	51.31	44.44	50.62	47.15	45.69	43.33	45.71	47.10	0.67
TSF3(SLE)	51.45	44.67	51.41	47.58	45.93	43.41	45.90	47.71	**1.01**
TSF1(FGS)	51.25	44.49	51.12	47.33	45.71	43.32	45.74	46.75	0.71
TSF2(FGS)	51.36	44.61	51.25	47.45	45.77	43.38	45.71	47.33	0.86
TSF3(FGS)	51.36	44.62	51.52	47.56	45.81	43.41	45.85	47.68	0.98
S/M(II)									
Coded	51.35	46.44	49.58	48.61	47.90	41.71	48.49	48.38	—
JTF [10]	51.55	46.49	49.87	48.68	47.98	41.89	48.77	48.52	0.16
DBRF [11]	51.80	46.81	50.59	49.19	47.20	41.84	48.69	47.88	0.19
ADTF [12]	52.32	47.11	50.96	49.28	48.02	41.81	48.76	49.80	0.70
CVBF [13]	52.68	47.25	51.08	49.51	48.14	41.92	48.82	50.00	0.87
TSF1(SLE)	52.55	47.33	50.54	49.33	48.20	41.74	48.70	49.57	0.69
TSF2(SLE)	52.72	47.38	50.51	49.38	48.20	41.80	48.76	49.87	0.77
TSF3(SLE)	52.89	47.43	50.99	49.54	48.20	41.85	48.90	50.40	**0.97**
TSF1(FGS)	52.74	47.28	50.98	49.48	48.08	41.83	48.76	49.92	0.82
TSF2(FGS)	52.88	47.37	50.97	49.49	48.10	41.90	48.85	50.14	0.90
TSF3(FGS)	52.84	47.31	51.13	49.54	48.12	41.94	48.91	50.41	0.96

Table 3
The objective quality measure of virtual view images synthesized with one/two given neighbouring views for different sequences when QP = 39.

	B	C	U	L	N	PH	PS	S	Gain
S/M(I)									
Coded	50.97	45.07	50.42	47.06	45.96	43.65	46.03	46.42	—
JTF [10]	51.17	45.01	51.14	47.29	46.04	43.85	46.33	46.68	0.24
DBRF [11]	50.71	44.94	51.77	47.48	45.41	43.40	46.33	46.06	0.06
ADTF [12]	51.71	45.36	52.09	47.71	46.29	43.51	46.28	47.94	0.66
CVBF [13]	51.87	45.40	52.23	47.74	46.28	43.72	46.36	48.15	0.77
TSF1(SLE)	51.98	45.41	51.86	47.74	46.24	43.57	46.11	47.67	0.62
TSF2(SLE)	52.08	45.34	51.65	47.64	46.15	43.64	46.22	48.12	0.66
TSF3(SLE)	52.06	45.47	52.39	47.91	46.34	43.71	46.39	48.65	**0.92**
TSF1(FGS)	52.01	45.40	52.14	47.75	46.21	43.63	46.26	47.81	0.70
TSF2(FGS)	51.98	45.45	52.24	47.80	46.22	43.70	46.22	48.29	0.79
TSF3(FGS)	51.90	45.41	52.48	47.89	46.27	43.72	46.36	48.58	0.88
S/M(II)									
Coded	52.16	47.68	50.13	49.43	48.51	41.93	48.96	49.24	—
JTF [10]	52.37	47.74	50.50	49.54	48.58	42.12	49.26	49.46	0.19
DBRF [11]	52.28	47.64	51.23	49.95	47.65	42.15	49.11	48.44	0.05
ADTF [12]	53.09	48.17	51.63	50.11	48.64	42.03	49.20	50.67	0.68
CVBF [13]	53.35	48.20	51.68	50.24	48.65	42.16	49.28	50.81	0.79
TSF1(SLE)	53.35	48.33	51.10	50.14	48.69	41.96	49.16	50.62	0.66
TSF2(SLE)	53.43	48.34	51.14	50.17	48.67	42.01	49.20	50.75	0.70
TSF3(SLE)	53.44	48.36	51.68	50.31	48.64	42.06	49.30	51.15	**0.86**
TSF1(FGS)	53.37	48.26	51.69	50.24	48.55	42.09	49.21	50.82	0.77
TSF2(FGS)	53.43	48.27	51.60	50.23	48.60	42.14	49.27	50.93	0.80
TSF3(FGS)	53.38	48.19	51.75	50.31	48.64	42.18	49.31	51.10	0.85

Table 4
The objective quality measure of virtual view images synthesized with one/two given neighbouring views for different sequences when QP = 36.

	B	C	U	L	N	PH	PS	S	Gain
S/M(I)									
Coded	52.35	46.42	52.06	48.00	46.71	44.20	46.67	48.01	—
JTF [10]	52.60	46.31	52.90	48.25	46.79	44.38	46.98	48.37	0.27
DBRF [11]	51.78	45.67	53.42	48.07	45.84	43.85	46.89	47.07	−0.23
ADTF [12]	53.09	46.54	53.73	48.52	46.99	43.91	46.97	49.33	0.59
CVBF [13]	52.99	46.43	53.97	48.62	46.91	44.18	47.04	49.55	0.66
TSF1(SLE)	53.20	46.35	53.56	48.63	46.71	43.99	46.78	49.18	0.50
TSF2(SLE)	53.25	46.37	53.44	48.58	46.70	44.05	46.87	49.37	0.53
TSF3(SLE)	53.16	46.38	54.09	48.75	46.87	44.08	47.01	49.80	**0.72**
TSF1(FGS)	53.11	46.43	53.94	48.55	46.86	44.03	46.93	49.28	0.59
TSF2(FGS)	53.14	46.36	53.94	48.53	46.81	44.07	47.02	49.50	0.62
TSF3(FGS)	53.03	46.27	54.13	48.60	46.83	44.08	47.11	49.67	0.67
S/M(II)									
Coded	53.45	48.90	51.20	50.50	49.39	42.27	49.68	50.55	—
JTF [10]	53.72	48.96	51.67	50.69	49.45	42.47	49.98	50.81	0.22
DBRF [11]	53.21	48.33	52.50	50.61	48.01	42.60	49.77	49.26	−0.05
ADTF [12]	54.32	49.19	52.83	51.14	49.42	42.36	49.91	51.67	0.54
CVBF [13]	54.28	49.12	52.97	51.11	49.32	42.54	50.02	51.86	**0.57**
TSF1(SLE)	54.41	49.27	52.40	51.11	49.32	42.26	49.83	51.71	0.46
TSF2(SLE)	54.43	49.23	52.42	51.14	49.23	42.31	49.86	51.56	0.46
TSF3(SLE)	54.32	49.21	53.00	51.15	49.16	42.35	49.95	51.97	0.54
TSF1(FGS)	54.35	49.18	53.07	51.19	49.04	42.48	49.94	51.87	0.55
TSF2(FGS)	54.34	49.14	52.95	51.17	49.16	42.51	49.96	51.86	0.55
TSF3(FGS)	54.22	49.01	53.12	51.15	49.28	42.53	49.96	51.87	0.56

Table 5
The objective quality measure of virtual view images synthesized with one/two given neighbouring views for different sequences when QP = 31.

	B	C	U	L	N	PH	PS	S	Gain
S/M(I)									
Coded	54.69	48.40	54.54	49.35	48.03	44.93	47.86	50.72	—
JTF [10]	55.00	48.11	55.56	49.65	48.25	45.09	48.11	51.25	0.32
DBRF [11]	54.64	48.05	55.79	49.60	47.96	45.19	48.14	50.70	0.20
ADTF [12]	54.83	48.24	55.97	49.88	48.28	44.79	48.01	51.48	0.38
CVBF [13]	54.96	48.29	56.32	49.69	48.55	45.07	48.10	51.82	**0.54**
TSF1(SLE)	55.20	48.23	55.66	49.70	47.98	44.70	47.76	51.04	0.22
TSF2(SLE)	55.21	48.20	55.47	49.69	47.93	44.78	47.85	51.31	0.25
TSF3(SLE)	55.13	48.35	56.05	49.82	48.30	44.86	48.01	51.61	0.46
TSF1(FGS)	55.13	48.31	56.21	49.96	48.40	44.88	47.97	51.10	0.44
TSF2(FGS)	54.78	48.30	56.30	49.89	48.33	44.93	48.04	51.22	0.41
TSF3(FGS)	55.01	48.23	56.42	49.89	48.48	45.01	48.14	51.30	0.50
S/M(II)									
Coded	55.68	50.65	52.92	51.94	50.92	42.83	50.79	52.84	—
JTF [10]	56.00	50.58	53.99	52.14	50.95	42.97	51.06	53.26	0.28
DBRF [11]	55.80	50.44	54.13	52.16	50.30	43.55	51.03	52.68	0.24
ADTF [12]	55.96	50.81	54.34	52.38	50.68	42.98	50.96	53.46	0.34
CVBF [13]	56.04	50.73	54.72	52.35	50.83	43.22	51.08	53.69	0.46
TSF1(SLE)	56.22	50.83	53.83	52.15	50.73	42.66	50.79	53.34	0.21
TSF2(SLE)	56.23	50.85	53.90	52.19	50.71	42.73	50.86	53.35	0.25
TSF3(SLE)	56.14	50.93	54.32	52.33	50.85	42.83	50.98	53.50	0.38
TSF1(FGS)	56.13	50.91	54.50	52.41	50.71	43.10	51.00	53.30	0.43
TSF2(FGS)	56.06	50.89	54.60	52.37	50.70	43.14	51.04	53.31	0.44
TSF3(FGS)	56.02	50.82	54.71	52.41	50.85	43.21	51.07	53.26	**0.48**

Poznan University [22,23]. Here, the intra mode of HEVC v9.0 [24] is employed to encode depth maps with quantization parameter (QP) set at 31, 36, 39, 41 and 43, respectively. In our experiment, we set λ to 1, which is used in [13]. It is common sense that non-linear filter results can be better with the increase of filtering window's size, but it can not be too large, because the larger window means more computational complexity. So suitable window's size is significant for good performance of filtering. We choose $L=8$ in the BSF step in order to tolerance greater distortions with larger quantization parameter, with one exception $L=4$ when $QP=31$, because depth image compressed with $QP=31$ has very small distortions, which usually include very few singular pixels. According to Eq. (6), $w_{i,j}$ becomes larger when σ^2 increases, thus leading to a more smoothed reconstruction. In the MRF model, α balances data term and smoothness term that measure the similarity between the reconstruction image F^* and the BSF-processed depth image I^F. In our experiments, σ^2 is set to 8 and α to 0.1, which are experimental values.

3.2. How to choose λ for detecting the reliable pixels

From Fig. 1, it can be noticed that reliable pixels of image belong to smoothing regions and only the discontinuous regions are detected for filtering, because the filtering of homogeneous regions doesn't make sense for the improvement of quality. But the usage of detecting reliable pixels of second stage filtering totally differs from the one of the first stage. As it is pointed out above that reliable pixels of the second stage are leveraged to be the given pixels in order to reconstruct the whole image in the MRF. Here, the percentages of unreliable pixels in whole depth image for all the tested sequences are counted out in Fig. 5, with λ ranging of detecting reliable pixels from 1 to 10. From the Fig. 5, it can been seen that the λ value of 3 tends to be a suitable and stable value from the perspective of unreliable pixels percentage statistically, because MRF reconstruction of two-stage filtering put emphasis on further artifacts removing rather than the skip-mode of homogeneous regions in the first stage. Besides, for proposed method thin objects can be well reconstructed if at least one pixel is detected as a reliable pixel. In most case, this condition can be satisfied, although an object alone discriminated from other objects with sharp edges has the possibility to be existed in the depth image. Thus, it has a risk of leading to an

Table 6
The time comparison for different filtering methods and the averaged coding time of each compressed depth image for various sequences.

M/S	B	C	U	L	N	PH	PS	S	Mean
Coding	6.32	9.46	15.73	5.99	6.31	16.14	15.94	16.58	11.56
JTF [10]	25.26	41.66	70.91	25.56	25.52	68.19	68.29	68.10	49.19
DBRF [11]	565.21	414.94	897.13	336.00	525.59	1097.77	1069.75	481.46	673.48
ADTF [12]	**0.41**	**1.04**	**1.81**	**0.27**	**0.41**	**2.24**	**1.20**	**1.80**	**1.15**
CVBF [13]	14.18	9.45	6.78	2.79	11.56	18.94	9.20	15.86	11.10
TSF1(SLE)	5.73	10.96	20.28	6.01	5.84	22.01	21.96	30.03	15.35
TSF2(SLE)	11.32	12.53	22.40	7.02	9.91	27.28	25.36	36.83	19.08
TSF3(SLE)	32.54	23.78	31.00	10.53	25.57	51.04	37.57	53.15	33.15
TSF1(FGS)	3.96	5.10	5.00	1.23	3.19	5.34	3.31	4.69	3.98
TSF2(FGS)	6.41	4.42	3.55	1.66	5.11	8.09	4.57	7.06	5.11
TSF3(FGS)	24.93	31.22	18.68	5.36	21.51	31.10	14.89	24.81	21.56

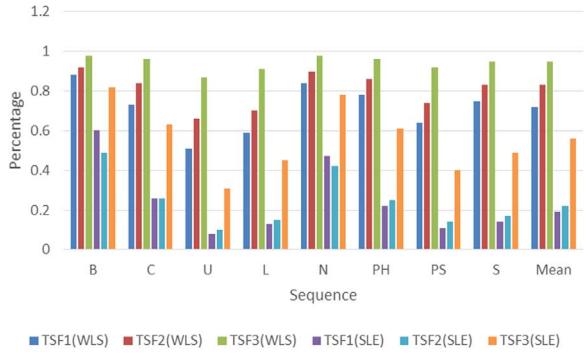

Fig. 6. The percentage of first stage filtering in proposed method: The x-axis is the testing sequence from B to S and the last one is the mean of all the sequences; the y-axis is the percentage of first stage filtering time in the two stage filtering time.

incorrect reconstruction when all the pixels of very small objects are labelled as un-reliable pixels.

3.3. Experimental comparison

We compare our methods, which include: BSF1/BSF2/BSF3 + MRF with the MATLAB backslash operator solving large linear equations (SLE) (which is respectively denoted as TSF1(SLE)/TSF2(SLE)/TSF3(SLE), and BSF1/ BSF2/BSF3 + MRF with Fast Global Smoothing (FGS) [25](denoted as TSF1(FGS)/TSF2(FGS)/TSF3(FGS) with JTF [10], DBRF [11], ADTF [12], and CVBF [13]. As we all known, the transmitted depth images are not to be displayed for viewers, but they are employed to render the virtual images. Thus, to evaluate its influence on virtual view rendering, we synthesize some virtual views and measure the resulted peak signal-to-noise ratio (PSNR). For given two views, the holes of occluded regions in one view can filled by another one, so left holes are very few, which has a good measurement for filtered depth image quality. But at special case, such as only one view RGB-D images are given, so the objective and visual measurement of the synthesized images with one given view is another way to measure depth quality [26]. To facilitate a more fair comparison, the virtual viewpoint synthesis software with 1D-fast mode of 3D-HEVC (HTM-DEV-2.0-dev3 version) [27] is adopted as the platform with the uncompressed texture images and the corresponding compressed depth images (filtered or not filtered). In the meanwhile, the performances of virtual images respectively rendered with one given view RGB-D images and two given views RGB-D images are tested. The YUV PSNR mean values with synthesized middle virtual view as the objective evaluation of synthesized images are presented in Tables 1–5, where S/M(I) denotes the Sequence/Method with one view to render the virtual view and S/M(II) denotes the Sequence/Method with two given view to render the virtual view. Tables 1–5 present the objective PSNR performance of synthesized image of middle virtual view rendered with two given neighbouring views with various QP, while it presents the objective PSNR performance of synthesized image of middle virtual view synthesized with one given neighbouring views. From these tables, it can be seen that, as compared to JTF, DBRF, the objective performance of ADTF, CVBF, TSF1 and TSF2 are similar, but the PSNR improvement of proposed TSF3 has more stable and better performance than others. As we all known, the JTF has leveraged the color information to improve the quality of depth, but it is sensitive to weak edges of color image. Thus it may make sharp boundaries of depth image to be blurry discontinuity. Although DBRF has well performance to make the depth's edge sharp, it tends to make image's strong edge left and slope surface to be smoothed. Compared with ADTF and CBVF, the priority of TSF3 maybe come from that the optimal threshold of block's binary segmentation is used and the robust median value of each class is assigned to the pixels belonging to the corresponding class. Meanwhile, proposed could well keep non-smoothing regions and remove the artifacts caused by compression without over-smoothing the depth images. It's worthy to compare the performance between TSF1/TSF2/TSF3(SLE) and TSF1/TSF2/TSF3(FGS), which is included in Tables 1–5. From these tables, SLE and FGS for solving the MRF's minimization problem have similar performance on the synthesized virtual images' PSNR, but the filtering time of FGS are less than SLE's and other methods except ADTF in Table 6. The ratio between filtering time of first stage and two-stage's time are given in Fig. 6. From this figure, it's interesting to find that the most time of TSF1(SLE)/TSF2(SLE)/TSF3(SLE) is spent on the second stage filtering while TSF1(WLS)/TSF2(WLS)/TSF3(WLS) major filtering time is costed on the first stage. Besides, the averaged coding time for each compressed the depth image including the encoding and decoding is given in the Table 6, from which it can be known that TSF1(FGS) and ADTF has more priority than others in view of time complexity. With the hardware's revolution, the hardware parallel acceleration of the first stage filtering could make proposed method more powerful.

To better demonstrate the efficiency of proposed method, the visual comparison is given, where both various depth images of two views (filtered or not filtered) and virtual view images rendered with these

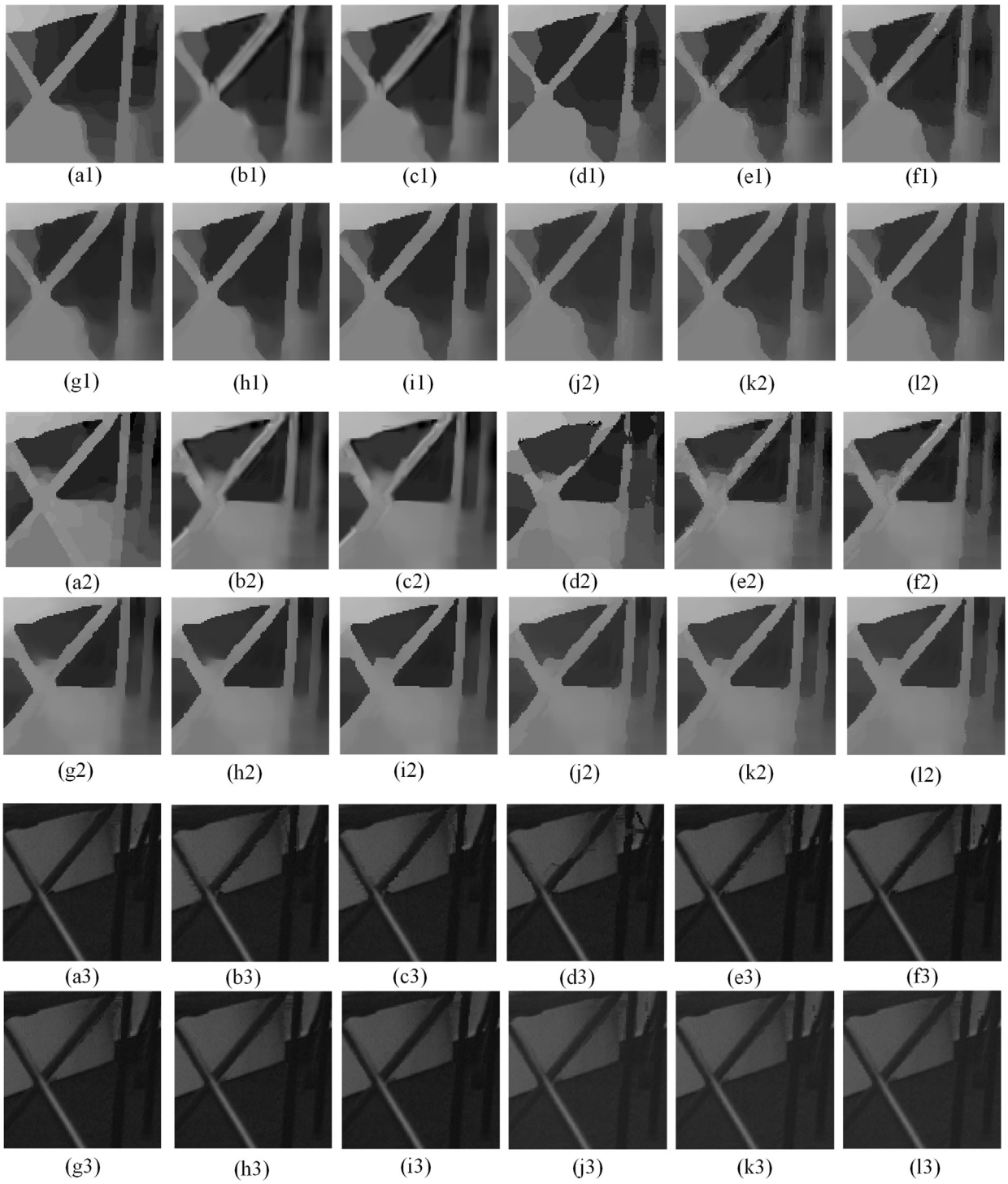

Fig. 7. (a1) Part of the original first frame of depth map for Book_Arrival in View 08; (b1) HEVC compressed depth map with QP = 43; (c1–l1) are filtered images from (b1) in order by JTF, DBRF, ADTF, CVBF, TSF1(SLE), TSF2(SLE), TSF3(SLE), TSF1(FGS), TSF2(FGS), TSF3(FGS); (a2)–(l2) is the part depth images in View 10 like (a1–l1); (a3)–(l3) Parts of synthesized image with two neighbouring views' depths from (a1)–(l1) and (a2)–(l2).

depths are presented in the Figs. 7–9. It can be seen clearly from these figures that our method can enhance structure information of HEVC-encoded depth images better than other methods, especially along discontinuity regions. At the same time, we can see that TSF-synthesized images have better visual quality than all other methods. In particular, our filtering preserves more discontinuity of the object with

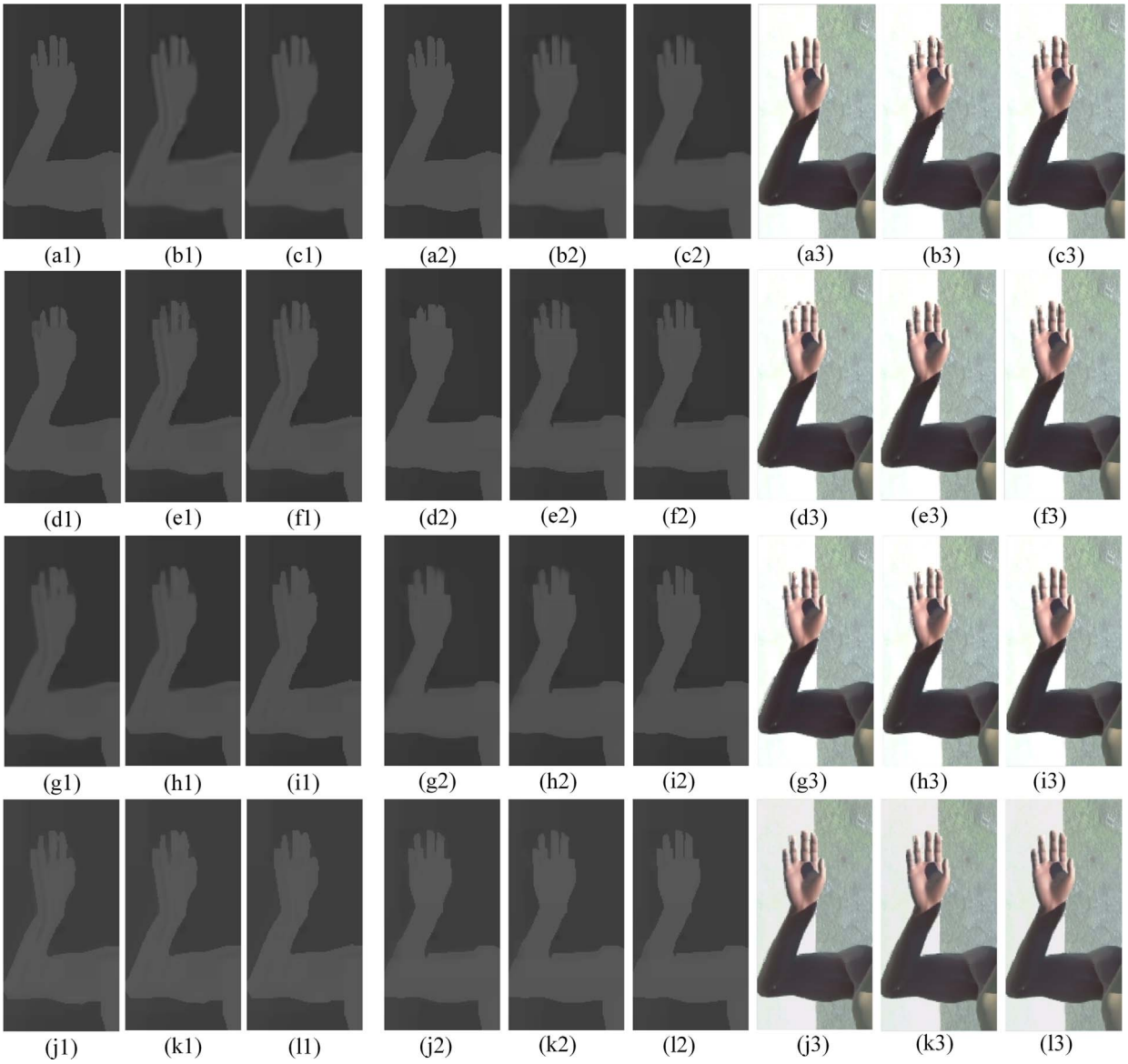

Fig. 8. (a1) Part of the original 31-th frame of depth map for Undo_Dancer in View 01; (b1) HEVC compressed depth map with QP = 36; (c1–l1) are filtered images from (b1) in order by JTF, DBRF, ADTF, CVBF, TSF1(SLE), TSF2(SLE), TSF3(SLE), TSF1(FGS), TSF2(FGS), TSF3(FGS); (a2)–(l2) is the part depth images in View 05 like (a1–l1); (a3)–(l3) Parts of synthesized image with two neighbouring views' depths from (a1)–(l1) and (a2)–(l2).

less deformed distortion in the filtered depth image and corresponding synthesized virtual images. One main reason for this improved performance is that the pre-processing BSF has done the pre-filtering to prepare for the next stage filtering. Meanwhile, boundary-preserving and coding-artifact removal benefits from the two terms of MRF, i.e., data term and smoothness term.

4. Conclusion and future works

A new two-stage depth filtering is explored in this paper by combining binary segmentation based depth filtering with MRF-based filtering. The first one handles contaminated depth boundary region caused by coding and the second maintains inherent continuity within object and discontinuity between adjacent objects. The experimental results clearly demonstrate that the performance of our method is superior over other methods, measured either by the quality of processed depth images or from the synthesized virtual views. Our future work will be focusing on the edge alignment between color video and depth video. The consensus of color-depth videos in the spatial-time domain will be investigated so as to produce even better depth videos and virtual views. Note that this paper mainly designs a filtering method for HEVC-compressed depth image. It can be applied into depth image coded by H.264 and any other standard codecs, although experimental results are provided only for HEVC compressed image.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61672087, 61672373), and CCF-Tencent Open Fund.

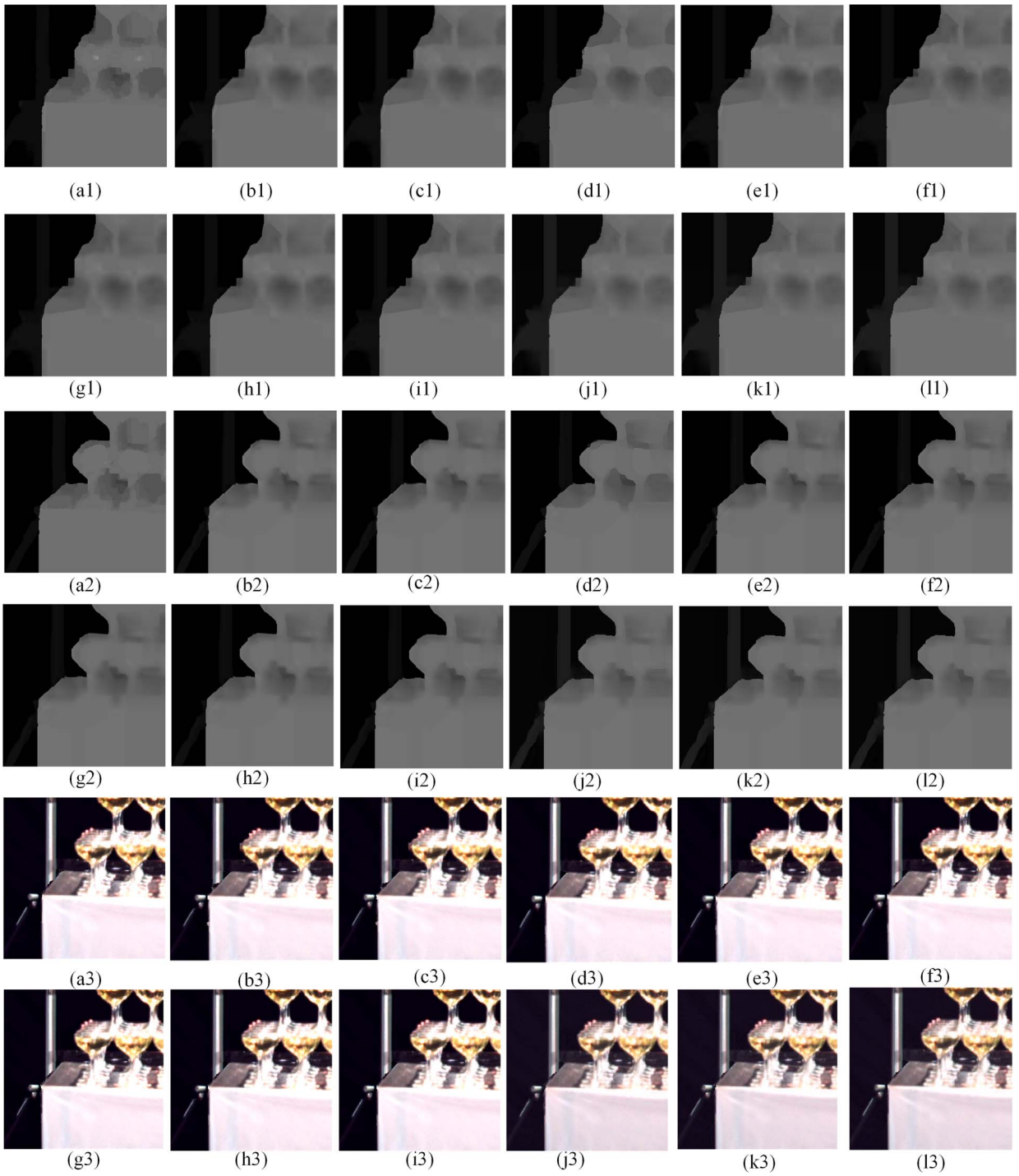

Fig. 9. (a1) Part of the original first frame of depth map for Champagne_Tower in View 37; (b1) HEVC compressed depth map WITH QP = 41; (c1–l1) are filtered images from (b1) in order by JTF, DBRF, ADTF, CVBF, TSF1(SLE), TSF2(SLE), TSF3(SLE), TSF1(FGS), TSF2(FGS), TSF3(FGS); (a2)–(l2) is the part depth images in View 39 like (a1–l1); (a3)–(l3) Parts of synthesized image with two neighbouring views' depths from (a1)–(l1) and (a2)–(l2).

References

[1] A. Singhal, J. Luo, W. Zhu, Probabilistic spatial context models for scene content understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, no. 235–241, http://dx.doi.org/10.1109/CVPR.2003.1211359.

[2] P. Merkle, A. Smolic, K. Muller, T. Wiegand, Multi-view video plus depth representation and coding, in: Proceedings of the IEEE International Conference on Image Processing, 2007, no. 201–204, http://dx.doi.org/10.1109/ICIP.2007.4378926.

[3] G. Sullivan, J. Ohm, W. Han, T. Wiegand, Overview of the high efficiency video coding (hevc) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668, http://dx.doi.org/10.1109/TCSVT.2012.2221191.

[4] J. Diebel, S. Thrun, An application of markov random fields to range sensing, in: Proceedings of the Conference on Neural Information Processing Systems, 2005, pp. 291–298, http://dx.doi.org/10.papers.nips.cc/paper/2837.

[5] H. Alastair, P. Newman, Image and sparse laser fusion for dense scene reconstruction, in: Field and Service Robotics, Springer, Berlin Heidelberg, 2010, http://dx.doi.org/10.1007/978-3-642-13408-1_20.

[6] Q. Yang, R. Yang, J. Davis, D. Nister, Spatial-depth super resolution for range images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8, http://dx.doi.org/10.1109/CVPR.2007.383211.

[7] M. Dongbo, L. Jiangbo, M. Do, Depth, video enhancement based on weighted mode filtering, IEEE Trans. Image Process. 21 (3) (2012) 1176–1190, http://dx.doi.org/10.1109/TIP.2011.2163164.

[8] J. Liu, X. Gong, J. Liu, Guided inpainting and filtering for kinect depth maps, in: Proceedings of the IEEE International Conference on Pattern Recognition, 2012, pp.

2055–2058, ⟨http://ieeexplore.ieee.org/document/6460564⟩.
[9] D. Silva, W. Fernando, H. Kodikaraarachchi, S. Worrall, A. Kondoz, Adaptive sharpening of depth maps for 3d-tv, Electron. Lett. 46 (23) (2010) 1546–1548, http://dx.doi.org/10.1049/el.2010.2320.
[10] S. Liu, P. Lai, D. Tian, C. Chen, Joint trilateral filtering for depth map compression, in: Proceedings of the SPIE 7744, Visual Communications and Image Processing 2010, 2010, http://dx.doi.org/10.1117/12.863341.
[11] K. Oh, A. Vetro, Y. Ho, Depth coding using a boundary reconstruction filter for 3-d video systems, IEEE Trans. Circuits Syst. Video Technol. 21 (3) (2011) 350–359, http://dx.doi.org/10.1109/TCSVT.2011.2116590.
[12] X. Xu, L. Po, C. Cheung, K. Cheung, L. Feng, Adaptive depth truncation filter for mvc based compressed depth image, Signal Process.: Image Commun. 29 (3) (2014) 316–331, http://dx.doi.org/10.1016/j.image.2013.12.005.
[13] L. Zhao, A.W.B. Zeng, Y. Wu, Candidate value-based boundary filtering for compressed depth images, Electron. Lett. 51 (3) (2015) 224–226, http://dx.doi.org/10.1049/el.2014.3912.
[14] J. Weijer, R. Boomgaard, Local mode filtering, in: Proceedings of the IEEE International Conference on Pattern Recognition, 2001, pp. 428–433, http://dx.doi.org/10.1109/CVPR.2001.990993.
[15] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2013) 1397–1409, http://dx.doi.org/10.1109/TPAMI.2012.213.
[16] N. Ohtsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66, http://dx.doi.org/10.1109/TSMC.1979.4310076.
[17] Electronic Telecommunication Res Institute (ETRI) and Gwangju Institute of Science and Technology (GIST) Korea, Apr, 2008, 3DV Sequence of ETRI and GIST, ⟨ftp://203.253.130.48⟩.
[18] Fraunhofer Heinrich Hertz Inst Berlin, Germany, Sep, 2013, 3DV Sequence of HHI, ⟨ftp://hhi.de/HHIMPEG3DV/⟩.
[19] Nagoya University Japan, 3DV Sequence of Nagoya University, ⟨http://www.tanimoto.nuee.nagoya-u.ac.jp/MPEG-FTVProject.html⟩.
[20] Nokia, Finnland, 3DV Sequence of Nokia, ⟨http://mpeg3dv.nokiaresearch.com/sequence.html⟩.
[21] National Institute of Information and Communication Technology (NICT) Japan, 3DV sequence of NICT, ⟨ftp://ftp.merl.com⟩.
[22] Poznan University of Technology, 3DV Sequence of Poznan University, ⟨ftp://multimedia.edu.pl/3DV/⟩.
[23] ISO/IEC/JTC1/SC29/WG11/MPEG2011/N12036, Geneva Switzerland, March 2011, Call for Proposals on 3D Video Coding Technology (VQEG_3DTV_2011_022_MPEG w12036(3DV CfP)FINAL.doc), ⟨ftp://vqeg.its.bldrdoc.gov/Documents/VQEG_Seoul_Jun11/MeetingFiles/3DTV/⟩.
[24] JCT-VC, HEVC Test Software [Online], ⟨https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-9.0/⟩.
[25] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, M. Do, Fast global image smoothing based on weighted least squares, IEEE Trans. Image Process. 23 (12) (2014) 5638–5653, http://dx.doi.org/10.1109/TIP.2014.2366600.
[26] D. Tian, P. Lai, P. Lopez, C. Gomila, View synthesis techniques for 3d video, Proceedings SPIE 7443, Applications of Digital Image Processing XXXII 7443, http://dx.doi.org/10.1117/12.829372.
[27] JCT-3V, 3D-HEVC Test Software (HTM) [Online], ⟨http://hevc.kw.bbc.co.uk/git/w/jctvc-3de.git/shortlog/refs/heads/HTM-DEV-2.0-dev3-Zhejiang⟩.