Information Sciences 320 (2015) 306–319

Contents lists available at ScienceDirect

Information Sciences

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

A bundled-optimization model of multiview dense depth map synthesis for dynamic scene reconstruction

You Yang [a,b], Xu Wang [c], Qiong Liu [a,*], Mingliang Xu [d], Li Yu [a]

[a] *Department of Electronics & Information Engineering, Huazhong University of Science & Technology, Wuhan, China*
[b] *Department of Automation, Tsinghua University, Beijing, China*
[c] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*
[d] *School of Information Engineering, Zhengzhou University, Zhengzhou, China*

ARTICLE INFO

Article history:
Received 15 June 2014
Received in revised form 26 October 2014
Accepted 6 November 2014
Available online 20 November 2014

Keywords:
Three-dimensional video
Optimization method
Depth image processing
Energy function optimization

ABSTRACT

Depth map is the basic requirement for all three-dimensional (3D) applications, but facing sensor noises, low frame-rate and low resolution in the procedure of data acquisition, especially in multiview cases. These problems bring obstacles to high quality 3D applications. Among the existing approaches, depth propagation is one of the promise approaches, and it can be utilized in temporal or spatial manner. However, propagation based algorithms process one aspect of the mentioned problems to pursuit local optimal solution. Actually, the process chain of depth map is from capture to application, and the optimization should be coupled instead of mutually independent. In this paper, we proposed a bundled-optimization scheme to process the thorough chain from capture to multiview dense depth map generation for the 3D applications. In this scheme, sensor noises in the captured low-resolution depth map are first detected and removed through a frequency-counting based non-linear filter. The filter refrains from the noise amplification in the procedure of depth map up-sampling. Low-pass blurring effect around high frequency areas is the by-product in up-sampling, and it is hard to detect in the depth map. We therefore propose a Blocklet based depth map optimization method for this blurring effect, and the accuracy of the high resolution depth map is then improved. Temporal depth propagation is then utilized on the optimized depth maps through the optical flow field rectified by temporal and spatial constrains. After that, a multi-set graph cut model is proposed to synthesize the multiview dense depth map. The experimental results indicate that our scheme can achieve at least 13.2575% PSNR gains when comparing to the benchmark depth map synthesis methods, and suggest the effectiveness of the proposed bundled-optimization method.

1. Introduction

Recently, three-dimensional (3D) video has witnessed a rapid development in both academia and industry, including 3D imaging [30,16,19], 3D video compression and communication [32,41,2], 3D modeling and retrieval [34,8,26,36,37,35,11,28], remote 3D medical treatment [43], and other fields. In these applications, the reconstructed 3D natural scene can provide realistic viewing experiences, and it has become an important extension for the traditional 2D vision applications. Among

[*] Corresponding author. Tel.: +86 27 87542667.
E-mail address: q.liu@hust.edu.cn (Q. Liu).

http://dx.doi.org/10.1016/j.ins.2014.11.014
0020-0255/© 2014 Elsevier Inc. All rights reserved.

all of the applications, depth map is critical because it records the 3D space coordinate information for every pixel corresponding to the captured color image. However, the generation of high resolution and rate depth map is not as convenient as color image, which can be easily captured by traditional cameras. These limitations bring challenges to 3D applications, for example, shutter glass based ultra-high Definition 3D movie with 120 Hz display frequency. So far, there is a trade-off between resolution and rate for depth map generation. As for high resolution, stereo matching methods were utilized to color image pairs. The requirement of heavy computation resources is the main hinder in high resolution depth image generation. On the other hand, as for high rate, RGB-D (e.g., Kinect) and ToF cameras are adopted for video-rate depth image capture but only low resolution (e.g., 320×240 pixels) can be obtained, and it is far away from practical usages [20,17,12,7]. Furthermore, multiview depth image capturing results in serious noise problem when multi-RGB-D or ToF cameras are utilized simultaneously [39]. In many applications, the inaccurate depth information is one of the limitations, such as multiple view analysis for 3D object retrieval [10,9,5]. High quality of multiview dense depth maps can lead to performance improvement in many applications. Therefore, multiview dense depth map synthesis methods were proposed to solve the resolution-rate problem.

Among the methods of dense depth map synthesis, propagation based schemes play an important role and they can be utilized in temporal or spatial manner. It is assumed in these schemes that depth and color images are different representations of the same scene, and thus they share the same temporal and spatial features [43,42,33,38,21,24]. Therefore in temporal schemes, motion vectors or optical flow among consequent color images which represents the temporal features is utilized for depth map synthesis. Specifically, the depth value for the region containing static object maintains un-changed in the consequent depth maps, while motion in consequent color images corresponds to variation of depth value in the corresponding region [33,38,42,21,24]. Therefore, the status (i.e., static or moved) of objects in consequent color images is utilized to identify the depth value variation in the depth map. Temporal depth propagation via motion vector or optical flow can solve the resolution-rate problem effectively in many cases, but the performance mainly depends on the fidelity of color images. For example, the performance drops down dramatically in the case of low bit-rate of 3D video communication [43]. In order to optimize the quality of depth map, side information such as motion vectors in compressed video bit-stream is utilized [43,23]. Besides that, the reconstructed depth map can be further rectified via localized quality improvement. For example, the block-partition parameter can be extracted from the bit-stream as the required side information [43]. The second type of depth propagation is in spatial manner, and constrains from inter-view affine transformation is the fundament of this scheme [22,18]. For example, cross-view iterative filtering is utilized for depth map spatial propagation at the decoder with the help of inter-lace sampling method adopted at the encoder [23]. However, spatial schemes are unable to solve the problem of rate. Therefore, up-sampling filter is usually accompanied with this scheme. So far, the problem of multiview dense depth map synthesis is still open for both temporal and spatial depth propagation.

In this paper, we propose a bundle-optimization scheme for multiview dense depth map synthesis. Noisy, low frame rate and low resolution depth maps are captured by depth sensors, and they should be optimized for better quality, higher resolution and higher frame rate. In this method therefore, we first solve the problem of low resolution for the key-time depth map. After that, flying pixel problem in the up-sampled depth map brings background–foreground ambiguous. A Blocklet based clustering method is therefore proposed for flying pixel removal in depth map. In this method, depth image is warped into 3D point cloud in the first. After that, a Blocklet region selection method is utilized to identify and rectify the flying pixels. Through the proposed scheme, the noises in original depth maps are removed, and flying pixel problem in up-sampled depth maps is cleared. Finally, in the procedure of multiview dense depth map synthesis, the multi-set graph cut model is built on the rectified candidates through optical flow. The experimental results show that our method can achieve at least 36.78% and 0.198 dB average gains on bad point ratio and PSNR for the test images, respectively. These results suggest the effectiveness of our bundled-optimization scheme in generating multiview dense depth maps.

The rest of this paper is organized as follows. The proposed scheme is discussed in details in Section 2. After that, experiments and discussions are presented in Section 3. Finally, we conclude our work in Section 4.

2. The proposed bundled-optimization scheme

As mentioned previously, we synthesize multiview dense depth maps for all non-key frames through the bundled-optimization methods. The processing procedure of the proposed scheme is given by Fig. 1. As described by this flowchart, we first solve the problem of low resolution for the key-time depth map. In this step, noises in the captured key-time depth map are detected and removed before up-sampling. After that, flying pixel problem is left in the up-sampled depth map and brings background–foreground ambiguous in the procedure of 3D content generation. In the second step therefore, the up-sampled depth map is optimized by Blocklet and clustering joint method to remove the flying pixels. Based on the high resolution key-time depth maps, we utilize a multi-set graph cut model to generate dense multiview depth maps for the non-key frames. We utilize the symbols and notations in Table 1 for following discussions.

2.1. Noise removal for original depth map

The noises in the captured depth map are the results of mis-detection over object convex surfaces or cross disturbance between depth sensors [14]. The noise distribution is content based and the stochastic model is usually unknown, and

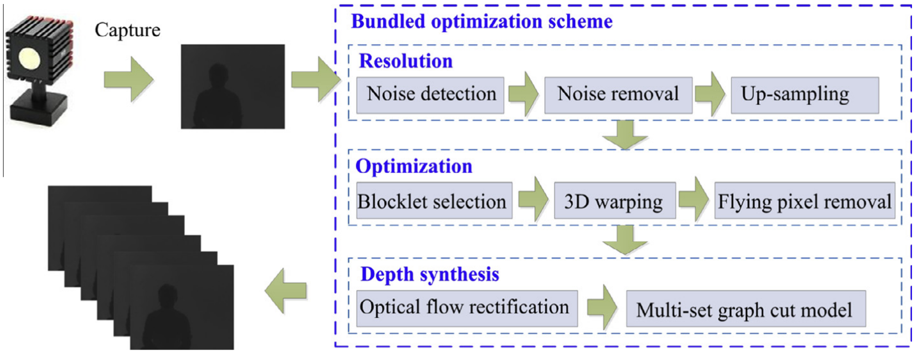

Fig. 1. The proposed bundled-optimization scheme for multiview dense depth synthesis.

Table 1
Symbols and notations utilized in discussions.

Symbols and notations	Descriptions
$W,\ W(i,j)$ and W_v	Window, pixel in window, and the value with The maximum occurrence
$\Omega_{W(i,j)}$	Neighborhood of $W(i,j)$
B and Θ	Block and mask image
$m,\ M$ and Φ	Point in 2D plane, its corresponding point In 3D space, and 3D point set
φ and γ	Block size and the threshold
d and s	Depth value measured in image space and physical World space
$l,\ L,\ p,\ P$	Label, label set, pixel and pixel set
$G,\ V,\ E,\ \Lambda$	Graph model, vertex, edge and weight set
C	Edge cut
$\Gamma(\cdot)$	Metric on labels

therefore it is hard to design a distribution model based filter for this kind of noises. In the proposed scheme, we modify the filter in [29] which is specialized for depth map reconstruction. Actually, noises in the captured depth map are singular values when comparing with surroundings. Therefore, we can select the depth value with the highest frequency in this region to replace the noise value. In order to solve this problem, a window W is centered by $W(i,j)$ with size $m \times n$, where both m and n are odd integers. Suppose W_v is the value with the maximum number of occurrence in W, we set $W(i,j) = W_v$. This filter is a frequent-counting non-linear filter, and the noises are detected through frequency counting within a specific window. After processed by this filter, noise pixel can be replaced by the existed depth value around this pixel and no new depth value is raised.

2.2. Flying pixel removal in up-sampled depth map

The blurred object boundaries in depth image bring challenges to 3D point clouds and models in the procedure of 3D content generation. We present a group of examples of blurred depth image in Fig. 2. In these examples, the standard test sequence *Undo_Dancer* of MPEG is selected [1], and the reference software VSRS_7.20 [15] is utilized for image rendering. The color images involved in rendering are used without distortion, and thus only the depth images are processed with Gaussian blurring filter. As shown in Fig. 2, the artifacts of boundaries in Fig. 2(d) is hard to be identified, although this figure is rendered by the blurred depth image (i.e., Fig. 2(b)). However, the artifacts turn out to be significant when the rendered images are represented by 3D point clouds, as shown in Fig. 2(f). The red[1] circles in these figures suggest that blurred object boundaries in depth image can result in flying pixels crossing background–foreground objects. Based on these examples, it can be found that the flying pixel is the key issue producing the aforementioned ambiguity in 3D applications, and brings challenges to 3D scene modeling.

Some related works have been proposed to remove the flying pixel. For example, non-local mean filter was proposed for the depth image that obtained by ToF camera [13,4]. The filter is very helpful for this kind of noisy depth image. However, it faces difficulties when depth image is generated by software or stereo matching. Therefore, a new flying pixel removal method is needed for practical 3D applications.

[1] For interpretation of color in Fig. 2, the reader is referred to the web version of this article.

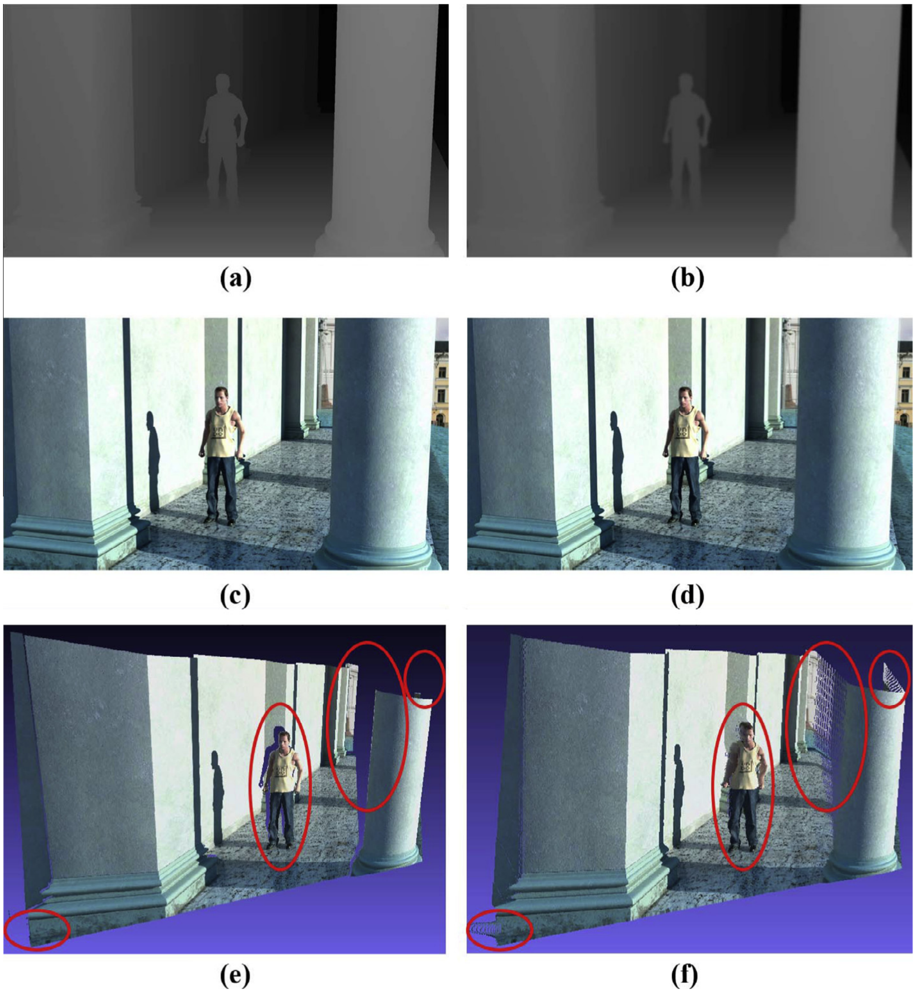

Fig. 2. Flying pixels caused by blurred boundaries. (a) The original depth image. (b) The depth image with blurred boundaries. (c) The rendered 2D image via (a). (d) The rendered 2D image via (b). (e) The rendered 3D point clouds via (a). (f) The rendered 3D point clouds via (b).

In order to rectify the flying pixels in the blurred depth map, we proposed a Blocklet based method, as shown in Fig. 3. In the first step, a Blocklet algorithm is utilized for local region selection around object boundaries. Our evidence lies in that flying pixels are local instead of global errors, subsequently they appear around object boundaries. After that, the depth image is warped into 3D point clouds, and the selected local region in depth image is also projected on the 3D clouds to locate the related 3D points. The 3D points involved are then clustered through the constraint of geometric consistency. During clustering, the flying pixels can be recognized and rectified to their proper positions. Finally, the optimized depth image is obtained from 3D point clouds through inversed warping transform.

We detail each step as below.

- Step 1: Blocklet based region selection

  The purpose of this step is to figure out a suitable point set for the subsequent clustering on the 3D point clouds. We first detect object boundaries in the depth image, in which the flying pixels reside. More specifically, we adopt edge detection operator, and the detected edges form a mask image for the subsequent processing.

  Based on the obtained mask image, a Blocklet region selection method is adopted, and the details of selection is given by Algorithm 1. In this algorithm, we figure out the number of edges in a block B, which is centered by edge pixel $W(i,j)$ in Θ. The block B should be split if more than one edge is inside. There therefore is only one edge in each block B after the process of Algorithm I, and then it is ready for the consequent clustering.

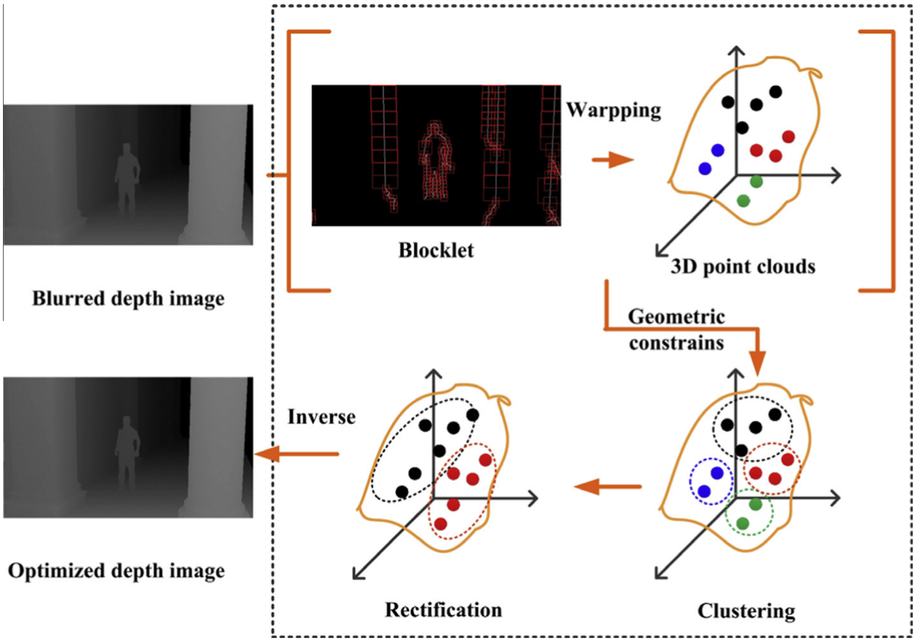

Fig. 3. The flowchart of the proposed optimization method for blurred depth image processing.

Algorithm 1. Blocklet selection.

```
Step 1: Input mask image $\Theta$, initialize block size $\varphi$.
Step 2: For each edge pixel $W(i,j)$ in $\Theta$, recursion
Step 2.1: if $\varphi <$ threshold $\gamma$, recursion terminated.
Step 2.2: identify the number of edges $\#e$ in the block $B$ centered at $(i,j)$ with size $\varphi \times \varphi$.
Step 2.3: if $\#e = 1$, all edge pixels in $B$ are marked as non-pixel symbol, store $B$ and recursion terminated.
else, $\varphi = \varphi/2$ and goto Step2.1 for deeper recursion.
Step 3: Output all block $B$, and then terminate.
```

After the process of Blocklet region selection, we warp the selected region in depth image into 3D space to obtain 3D point clouds. It is a transformation from image coordinate system to the 3D world coordinate system. For the 2D point $m = (x,y)^T$ with depth value d in the depth image, we determine its corresponding 3D point $M = (X,Y,Z)^T$ by the following model

$$s\tilde{m} = \mathbf{K}[\mathbf{Rt}]\widetilde{M} = \mathbf{P}\widetilde{M} \tag{1}$$

where s is the physical distance of the 3D point M, $[\mathbf{Rt}]$ is the extrinsic parameters matrix in the world coordinate system, $\mathbf{R}$ is the rotation matrix between the world coordinate system and the camera coordinate system, and $\mathbf{t}$ is the translation vector. In this model, $\mathbf{K}$ is the camera intrinsic matrix, $\widetilde{M} = (X,Y,Z,1)^T$ is the homogeneous coordinate of point M, $\tilde{m} = (x,y,1)^T$ is the homogeneous coordinate of 2D point m. In addition to this model, the relationship between depth value d and the physical distance s is given by

$$s = \frac{1}{d/255(1/z_{\min} - 1/z_{\max}) + 1/z_{\max}} \tag{2}$$

where $z_{\min}$ and $z_{\max}$ is the value of the nearest and farthest physical depth value of the scene, respectively.

Finally, we can obtained the coordinate of $\widetilde{M}$ as

$$\widetilde{M} = s\mathbf{P}^{-1}\tilde{m} \tag{3}$$

In the case of just one camera is involved, the camera coordinate system can be set as the same as the world coordinate system, and then $\mathbf{R}$ and $\mathbf{t}$ can be set as unit matrix and zero vector, respectively.

After the transformation process from image to the world coordinate, we can obtain 3D point clouds corresponding to the depth image, and group of corresponding point sets for the regions selected by Blocklet algorithm.

- Step 2: Clustering and rectification

  The purpose of this step is to figure out the flying pixels and then rectify them to their proper 3D position. With the help of Blocklet algorithm, a set of 3D points Φ in the point clouds corresponding to the region are selected. As indicated in Fig. 2, flying pixels are isolated from background and foreground objects. That said, a 3D point of one object plane keeps similar geometric consistency with its surrounding points, while the flying pixel breaks this consistency. Therefore, we utilize normal vector to measure the geometric consistency for the current 3D point $M_1(X_1, Y_1, Z_1)$ to its surrounding 8 points $M_i(X_i, Y_i, Z_i)(i = 2, 3, , 9)$, where $M_1(X_1, Y_1, Z_1)$ and $M_i(X_i, Y_i, Z_i)$ corresponds to $m_1(x, y))$ and 8-neighbors of $m_1(x, y)$ in depth image, respectively. The relationship of $M_1(X_1, Y_1, Z_1)$ and $M_i(X_i, Y_i, Z_i)$ is shown in Fig. 4. In this figure, 8 triangle meshes can be obtained, sharing the common center point $M_1(X_1, Y_1, Z_1)$, and normal vector $N = \{\delta, \beta, \chi\}$. These 8 meshes is the solution of the following equations

$$\begin{bmatrix} X_a - X_1 & Y_a - Y_1 & Z_a - Z_1 \\ X_b - X_1 & Y_b - Y_1 & Z_b - Z_1 \\ X_b - X_a & Y_b - Y_a & Z_b - Z_a \end{bmatrix} \begin{bmatrix} \delta \\ \beta \\ \chi \end{bmatrix} = 0 \tag{4}$$

  where (X_a, Y_a, Z_a) and (X_b, Y_b, Z_b) are coordinates of 3D points in $M_i(X_i, Y_i, Z_i)$, and M_1, M_a, M_b is the mesh depicted in Fig. 4. A K-mean cluster method is utilized on normal vectors of all 3D points in Φ. The 3D points in the top two clusters are treated as background–foreground pixels, while others are treated as flying pixels. For the process of rectification on flying pixel M_{fly}, we update its 3D coordinates according to its nearest cluster center iteratively until it is not the outlier for the top two clusters.
- Step 3: Inversed warping

  The rectified 3D point clouds in Step 2 is going to be converted into depth image in Step 3, and the process is an inversed warping from the world coordinate system to the image one. Therefore, the process is an inverse transformation of Eqs. (1)–(3).

2.3. Optical flow rectification

A high quality and high resolution key-time depth image is obtained in the process of depth optimization. After that, propagation in temporal and inter-view manner is performed. Pixel-wise vector (PMV) is crucial in the propagation because the depth information can be transit through this vector. However, PMV faults may occur around the occlusive or low textural regions because of the risk of mismatches. Therefore, these regions should be detected properly at first. The texture distribution is an important clue for the detection. Usually, the texture distribution is consistent with the variation of pixel value, so that it can be represented by standard deviation of pixel values.

In order to identify the texture or edge pixel $W(i,j)$ from color image, we propose a binary decision function $\zeta(\Omega_{W(i,j)})$ in form of the Heaviside step function to determine whether a region $\Omega_{W(i,j)}$ is the low textural region by

$$\zeta(\Omega_{W(i,j)}) = \lim_{k\to\infty}\left(1/2 + 1/\pi \tan^{-1} k\left(\sigma\left(I_{\forall W'} \in \Omega_{W(i,j)} - I_{W(i,j)}\right) - \varepsilon_\Omega\right)\right) \tag{5}$$

where $\Omega_{W(i,j)}$ is the neighboring pixel set centered by $W(i,j)$, W' is a pixel in $\Omega_{W(i,j)}$, $I_{W(i,j)}$ is the gray value for $W(i,j)$, $\sigma(\cdot)$ is standard deviation operator for a set of pixels, and ε_Ω is a threshold for texture detection. According to the definition of

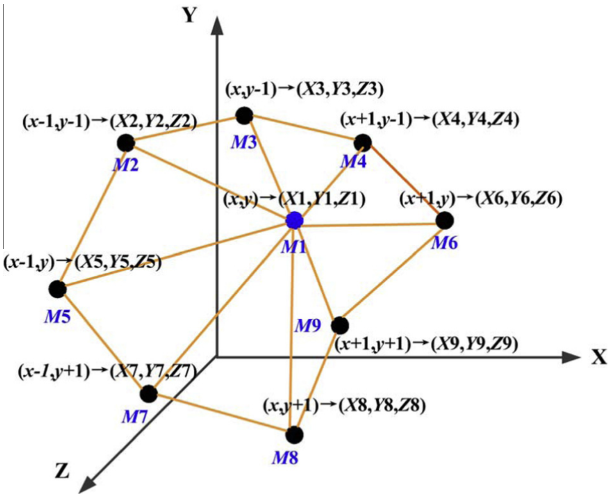

Fig. 4. 8-neighbor in depth image and the corresponding 8-neighbor 3D points mesh generation.

Heaviside step function, the value of $\zeta(\Omega_{W(i,j)})$ corresponds to a binary decision for textural region detection. $\zeta(\Omega_{W(i,j)}) = 0$ indicates the pixel $W(i,j)$ is surrounded by textures, but in low textural region when $\zeta(\Omega_{W(i,j)}) = 1$.

Similarly, we also propose a binary decision function $\rho(\mathbf{v})$ to determine whether a pixel $W(i,j)$ is occluded in consequent frames

$$\rho(\mathbf{v}) = \lim_{k\to\infty}\left(1/2 + 1/\pi\tan^{-1}k\left(\left(I_{W(i,j)+v} - I_{W(i,j)}\right)^2 - \varepsilon_I\right)\right) \tag{6}$$

where $\mathbf{v}$ is the PMV on $W(i,j)$, ε_I is a threshold for occlusion detection. According to this model, $\rho(\mathbf{v}) = 1$ is for the occluded pixel $W(i,j)$ and $\rho(\mathbf{v}) = 0$ is for the visible one. In the detection of occlusive or low textural region, smaller threshold results in higher accuracy of detection, and then it can have stable performances on different test materials. However, this may lead to heavy cost on computation resource which is unfavorable in practical cases. On the other hand, larger threshold can benefit the practical applications, but the accuracy of detection may drop down.

Based on $\zeta(\Omega_{W(i,j)})$ and $\rho(\mathbf{v})$, we obtain the status for both the pixel $W(i,j)$ and its surroundings, and the status can be utilized in making appropriate PMV rectifications. There are several possible cases aroused by $\zeta(\Omega_{W(i,j)})$ and $\rho(\mathbf{v})$. For the first case, the vector $\mathbf{v}$ for $W(i,j)$ is an error PMV when the pixel $W(i,j)$ is occluded (i.e., $\rho(\mathbf{v}) = 1$). The reason comes from the truth that no pixel correspondence can be found for $W(i,j)$ in consequent frames regardless $W(i,j)$ is surrounded by textures or not. In this case, rectification on this PMV for $\mathbf{v}$ will result in new error. Therefore, we mark $W(i,j)$ with a label $label(\mathbf{v})$ indicating unreliable and the process of $W(i,j)$ is postponed after the depth map has been reconstructed. Then for the second case, the pixel $W(i,j)$ is visible and surrounded by textures (i.e., $\rho(\mathbf{v}) = 0$ and $\zeta(\Omega_{W(i,j)}) = 0$). Texture is benefit for accurate optical flow calculation, and thus the vector $\mathbf{v}$ can be treated as reliable and accurate. For the last case, the pixel $W(i,j)$ is visible but in a low textural region (i.e., $\rho(\mathbf{v}) = 0$ and $\zeta(\Omega_{W(i,j)}) = 1$). Usually, low textural region is the reason of ambiguous vectors in optical flow calculation. These ambiguous PMVs are outliers and odd when comparing with neighboring PMVs. In this case, the ambiguous vectors are also unreliable, but they can be rectified by an average filtering with the neighboring vectors. We summarize the above processing as a condition function by follows

$$\Psi(\mathbf{v}) = \begin{cases} label(\mathbf{v}) & \rho(\mathbf{v}) = 1 \\ \mathbf{v} & \rho(\mathbf{v}) = 0\&\zeta\left(\Omega_{W(i,j)}\right) = 0 \\ avg\left(\underset{\mathbf{v}\,\in\,\Omega_{\overline{\mathbf{x}}}}{\mathbf{v}}\right) & \rho(\mathbf{v}) = 0\&\zeta\left(\Omega_{W(i,j)}\right) = 1 \end{cases} \tag{7}$$

where $label(\mathbf{v})$ is a mark on $W(i,j)$ reserved for postpone processing, $avg(\cdot)$ is an average operator on a set.

Based on the above occlusive and low textural region detection and rectification, most of the fault PMVs can be refined, and the remained PMVs of occlusive regions are reserved for later processing. Therefore, the accuracy of PMVs is improved.

2.4. Multi-set graph cut model and candidate selection

The accuracy of depth propagation in temporal dimension is improved by the above optical flow rectification. However, it is not enough because we need to have multi-view dense depth map. Therefore, depth propagation in both temporal and inter-view manner should be utilized. Here comes a new problem that ambiguous selection is aroused for depth value calculation on one pixel when the number of temporal and inter-view candidates increases. An optimization is needed to solve the problem of ambiguous candidate selection from both temporal and inter-view directions.

Generally, the optimization process of depth image synthesis is searching a proper label set L for pixel set P, where the selected L should be both smooth and consistent with the observed label value. The obtained label value $l \in L$ is assigned as the depth value for the current pixel p. The process is usually described by a minimum energy function

$$\min E(L) = E_{data}(L) + E_{smooth}(L). \tag{8}$$

In this energy function, E_{smooth} measures the smoothness of l when comparing to its neighbors, and E_{data} measures the difference between l and the observed data on $p \in P$. Actually, the energy function Eq. (8) is a global minimization problem because it will loop over P and L for the final result.

Two stages are arranged to handle the minimum energy function, including the definition of label set L and the energy minimization. Usually, a graph model is constructed to handle the minimum energy function, as depicted by Fig. 5(a). In this graph model, labels are designed as only one attribute. For example, all labels are pixel values and selected in one image. Specified in our problem, it is different from the above because the labels are selected from different images. A modified graph model is needed and proposed to optimize the minimum energy function specified in our problem. As indicated by Fig. 5(b), L is defined as correspondence obtained by the rectified PMV between non-key and key-time color images. In other words, one candidate in one reference is involved in L. Consequently, in stage two, an energy function is modeled to describe the candidate selection and handle the process of synthesis. After a loop over all non-key images, the synthesized dense multi-view depth image is obtained.

In order to improve the accuracy of L_i, the content similarity should be taken into account. Suppose a block B_i is centered by p_i, and select the best matches in reference image by

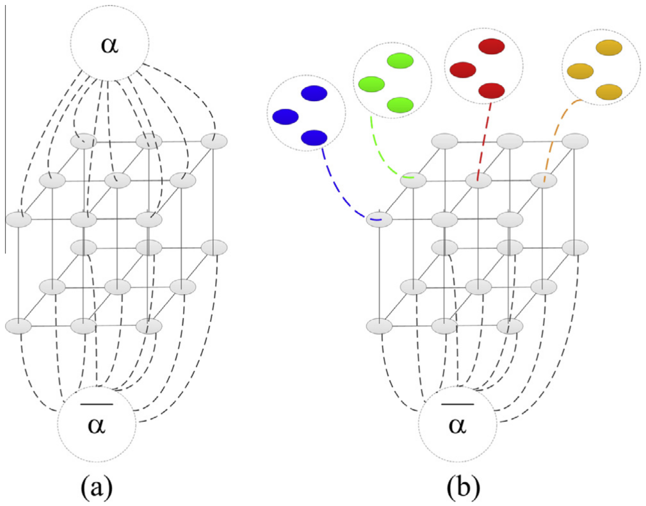

Fig. 5. Graph cut models for energy minimization.

$$B_r = \arg\max(||B_i - B_r||_\mu + \lambda\Psi(B_i - B_r)) \tag{9}$$

where $||\cdot||$ is a norm operator, $\Psi(\cdot)$ is a content similarity measurement, and λ is the Lagrange operator. B_i is a block in the current color image, B_r is a block in the reference. In the experiment section, we set $\mu = 2$ and $\Psi(\cdot)$ is

$$\Psi(B_i, B_r) = \frac{\sum_{j=1}^{n}(B_i^j - \overline{B_i})(B_r^j - \overline{B_r})}{\sqrt{\sum_{j=1}^{n}(B_i^j - \overline{B_i})^2 \cdot \sum_{j=1}^{n}(B_r^j - \overline{B_r})^2}} \tag{10}$$

Graph cut model is widely utilized for the energy minimization model Eq. (8) [3], and it is based on the definition of graph. Specifically, let $G = \{V, E, \Lambda\}$ be a weighted graph, where V, E and Λ corresponds to the vertex, edge and weight set. Vertices $v_i,\ v_j \in V$ are joined by edge $e_{ij} \in E$ and accompanied by a weight $\omega_{ij} \in \Lambda$. There are two distinguished vertices $v_{sink},\ v_{start} \in E$ called start point and sink point, respectively. A cut $C \subset E$ is a subset of edges such that the two vertices v_{sink} and v_{start} are cut by the induced graph $G(C) = \{V, E - C\}$. In this case, no proper subset of C can cut the vertices in $G(C)$. The cost of the cut C is the sum of its edge weight, and it is denoted by

$$|C_\Lambda| = \sum_{e_{ij}\ \in C} \omega_{ij}. \tag{11}$$

The minimum cut of C, which has the least cost among all cuts.

$$C = \arg\min C_\Lambda \tag{12}$$

Based on the definition above, a graph G_α is constructed to minimize the energy function Eq. (8) with the structure shown in Fig. 5(a). The graph G_α is built for a given label set $L = \{\alpha_1, \alpha_2, \ldots, \alpha_{||L||}\}$, and $\overline{\alpha_i} = L - \{\alpha_i\}$ is utilized for the complementary set of α_i. For any two labels $\alpha_i,\ \alpha_j \in L$, if they are in the same data-domain (e.g., $L \subset I^+$ is a set of positive integer labels), a metric $\Gamma(\alpha_i, \alpha_j)$ is defined with following features.

$$\begin{aligned}
&\Gamma(\alpha_i, \alpha_j) = 0 \Longleftrightarrow \alpha_i = \alpha_j\\
&\Gamma(\alpha_i, \alpha_j) = \Gamma(\alpha_j, \alpha_i)\\
&\Gamma(\alpha_i, \alpha_j) \leqslant \Gamma(\alpha_i, \alpha_k) + \Gamma(\alpha_k, \alpha_j)
\end{aligned} \tag{13}$$

Based on Eq. (13), the energy minimization problem can be processed by the optimal swap moves on the graph structure G_α [3]. The optimal swap algorithm is in fact a labeling process. Each pair of neighboring pixels $p, q \in V$ which are not cut by the partition C is connected by an $n - link$. Auxiliary vertex is the candidate label α, and $t - link$ connects the auxiliary vertex and the v_{sink}. $n - link$ and $t - link$ is utilized for the minimum cost computation.

However, different from the traditional energy minimization problem, the constructed graph structure G_α has difficulties in handling our energy minimization problem. The main reason lays on the fact that elements in label set L do not belong to one data-domain, and then the metric $\Gamma(\cdot)$ cannot be defined as usual. As we have mentioned previously, the label set L in our problem is composed of blocks rather than others. Furthermore, the block B_i has a correspondence to L_i, and it is represented by $f(B_i) = L_i$. Therefore, the label $l_j \in f(B_i)$ cannot be assigned to block $f(B_k)$, where $i \neq k$ and $j = 1, 2, \ldots, ||R||$, and $\Gamma(l_x, l_y)$ is meaningless when l_x and l_y belongs to different label subsets. In this case, each element in the energy minimization

Eq. (8) should be re-defined according to our problem. We modify the data term $E_{data}(L)$ to measure the difference between current label $l_j \in L_i$, as following

$$E_{data}(L) = \sum_{B_i \in I, l_x, l_y \in L_i} \Gamma_d(l_x, l_y) \tag{14}$$

where $\Gamma_d(l_x, l_y)$ is obtained by

$$\Gamma_d(l_x, l_y) = \frac{1}{||B_i||} \sum_{p \in B_i} (l_x(p) - l_y(p))^2 \tag{15}$$

After that, the smooth $E_{smooth}(L)$ measures the smoothness of L when comparing to its neighbors, i.e., smoothness of label $l_x \in L_i$ and $l_y \notin L_i$, and it can be calculated as

$$E_{smooth}(L) = \sum_{B_i \in, l_x \in L_i, l_y \notin L_i} \Gamma_s(l_x, l_y) \tag{16}$$

where $\Gamma_s(l_x, l_y)$ is calculated as

$$\Gamma_s(l_x, l_y) = \sum_{x \in B_i, y \in B_j, \{x,y\} \in N} \left(\frac{I(x) - I(y)}{D(x,y)} \right) \tag{17}$$

where $D(x,y)$ is the 2-order Minkowski distance between x and y. After that, the Eq. (8) can be re-organized as

$$E(L) = \sum_{B_i \in I, l_x, l_y \in L_i} \Gamma_d(l_x, l_y) + \sum_{B_i \in, l_x \in L_i, l_y \notin L_i} \Gamma_s(l_x, l_y) \tag{18}$$

The definition of $\Gamma_d(\cdot)$ and $\Gamma_s(\cdot)$ satisfies the requirement of Eq. (13).

After the definition of energy function Eq. (18), we utilize the graph cut model for minimization. As we discussed before, different blocks have their corresponding labels. This brings difficulties to the construction of G_α, because it is different to the traditional graph model where the label α in G_α is used to be shared by all vertices. In order to handle this problem, we construct a new graph structure $G_{\alpha-set}$ as shown in Fig. 5(b). In this graph structure $G_{\alpha-set}$, $n-link$ connects the block to its corresponding candidate label set, and there is no link connecting the block to label sets that belongs to other blocks. In this case, all candidate labels from all label set L_i other than the selected labels that related to $n-link$ are included in $\overline{\alpha}$. After the construction of $G_{\alpha-set}$, the remaining optimization process of $G_{\alpha-set}$ is similar to G_α, and the energy minimization can be obtained.

3. Experiments and discussions

In this section, we verify the performance of the proposed method by standard MPEG test sequences (namely *Breakdancer*, *Loverbird1* and *Undo_Dancer*) with different characteristics and difficulties. Difficulties in the selected test sequences are as following, for example, *Undo_Dancer* is generated by software indicating high quality and accuracy, *Breakdancer* and *Lovebird1* is an indoor and outdoor complex scene, respectively.

3.1. Flying pixel removal

In this section, we utilize our method to the up-sampled depth maps, and then render virtual view images and 3D point clouds for comparison. As suggested by [31], we selected bad point ratio (BPR) to evaluate the accuracy of depth image, and PSNR for the quality of virtual view image. We compare our method by two baselines. The first one is the non-local mean filter proposed in [13,4] which is specialized for flying pixels processing. Objective performance is utilized for this comparison. After that, the second baseline is the bilateral and trilateral filters that proposed in [25] for edge preservation depth image enhancement. Performances on BPR and PSNR will be presented for this comparison. Parameter tuning for Gaussian smooth filter is with window size 3 × 3, 5 × 5 and 7 × 7, and $\sigma = 0$, and it is utilized to obtain the blurred depth image and simulating the after-effect of image up-sampling. These tuning parameters are widely used for depth images before 3D content generation [27,6].

The experimental results for quantitative comparison are listed in Tables 2 and 3. In these tables, Tun. Para. is for the window size of Gaussian filter, GF stands for the Gaussian filter, BF and TF stands for bilateral and trilateral filter in [25], respectively. The average gains are calculated through the difference between the proposed and the corresponding baseline methods. As suggested by Table 2, our method reduces the BPR and optimizes the quality of blurred depth image significantly, and has average BPR gain 41.59% and 36.78% when comparing to BF and TF, respectively. The reason for these gains comes from the process around object boundaries. BF and TF methods process the flying pixel by assigning new values through calculation on its surroundings, and the value of object boundaries disturbs the accuracy of this assignment. As for the proposed method, the assignment is a selection in the surrounding with the highest possibility. Therefore, the assigned value is either background or foreground value, and the accuracy improvement is reasonable.

Table 2
Comparative results of bad point ratio (%).

Test image	Tun. Para.	BF in [25]	TF in [25]	Proposed
Undo_Dancer	3 × 3	0.7793	0.683	0.2778
	5 × 5	0.9385	0.8163	0.5516
	7 × 7	1.1766	1.0214	0.7372
Lovebird1	3 × 3	0.8130	0.5820	0.3099
	5 × 5	0.9474	0.7078	0.4125
	7 × 7	1.1878	0.9007	0.6379
Breakdancer	3 × 3	2.3115	2.6744	1.7058
	5 × 5	2.7377	3.1570	2.1572
	7 × 7	3.4017	3.7020	2.7439
Average gain (%)		41.59%	36.78%	–

Table 3
Comparative results for quality of virtual viewpoint image (dB).

Test image	Tun. Para.	GF	BF in [25]	TF in [25]	Proposed
Undo_Dancer	3 × 3	39.531	41.927	38.329	41.184
	5 × 5	37.922	40.118	37.368	39.467
	7 × 7	37.212	38.826	37.157	38.034
Lovebird1	3 × 3	40.788	38.747	39.207	40.118
	5 × 5	40.067	38.673	38.835	39.738
	7 × 7	39.155	38.32	38.539	39.133
Breakdancer	3 × 3	36.204	35.851	34.851	36.534
	5 × 5	35.661	35.671	34.753	35.83
	7 × 7	35.082	35.476	34.516	35.342
Average gain (dB)		0.418	0.198	1.314	–

After the flying pixels have been removed, we utilize the rectified depth image for virtual view image rendering, and the results are listed in Table 3. In this table, benchmarks include the results of GF, the depth image processed by GF optimized by BF and TF. The average gains are given in dB and calculated through the difference between the proposed and the corresponding benchmark methods. As can be found through these results, the quality of rendered virtual view image is improved by our method with average gain 0.418 dB when comparing to the benchmark GF method. Besides that, we have 0.198 and 1.314 dB average gains for BF and TF, respectively. The reason for these gains comes from the fact that disturbances around object boundaries are inevitable for these benchmark methods. Our method can process these boundary regions properly. Results in Tables 2 and 3 suggest that our method can improve the accuracy of depth image and the quality of generated 3D content simultaneously, and it thanks to the flying pixel identification and rectification in our method.

After the subjective evaluation of our method, we take *Undo_Dancer* as an example for further objective comparison, and the rendered 3D point clouds with texture information through different depth images are shown in Fig. 6. Specifically, Fig. 6(a) is rendered by the original depth image, and Fig. 6(b) is rendered by depth image that processed by Gaussian smooth filter. Flying pixels can be clearly identified around object boundaries. The blurred depth image utilized in Fig. 6(b) is processed by our method, and then the optimized depth image is utilized to render Fig. 6(c). The result suggests that most of the flying pixels are rectified while details of objects in the scene are remained. It thanks to that our method can identifies and rectifies the flying pixels around object boundaries instead of other regions. For comparative studies, we utilize non-local mean filter to the depth image in Fig. 6(b), and then the rendered 3D point clouds are given by Fig. 6(d). The results suggest that most of the flying pixels are removed in the scene, but the details of the objects are lost, especially the 3D geometric information.

3.2. Dense depth map synthesis

In this section of experiments, we synthesis dense depth maps for the test sequences *Undo_Dancer*, *Breakdancer* and *Lovebird1*. We compare our method with the state-of-the-art bi-directional depth propagation method in [40], and PSNR is the subjective quality parameter that utilized for performance comparison. In order to have a deep insight on the performance of our method, we arrange three experiment settings for performance comparison. The main difference between these settings is the distance between two key-time depth maps. In the *Setting 1*, the distance is set as 2, indicating the depth maps on t and $t+2$ are utilized as key-time depth maps in synthesizing the depth map on $t+1$. Similar as *Setting 1*, the distance for *Setting 2* and *Setting 3* is set as 4 and 8, respectively. In this case, 3 and 7 depth maps will be synthesized for *Setting 2* and *Setting 3*, respectively. The quantitative results are listed in Tables 4 and 5, and Fig. 7 provides the objective comparisons of test sequence *Undo_Dancer*.

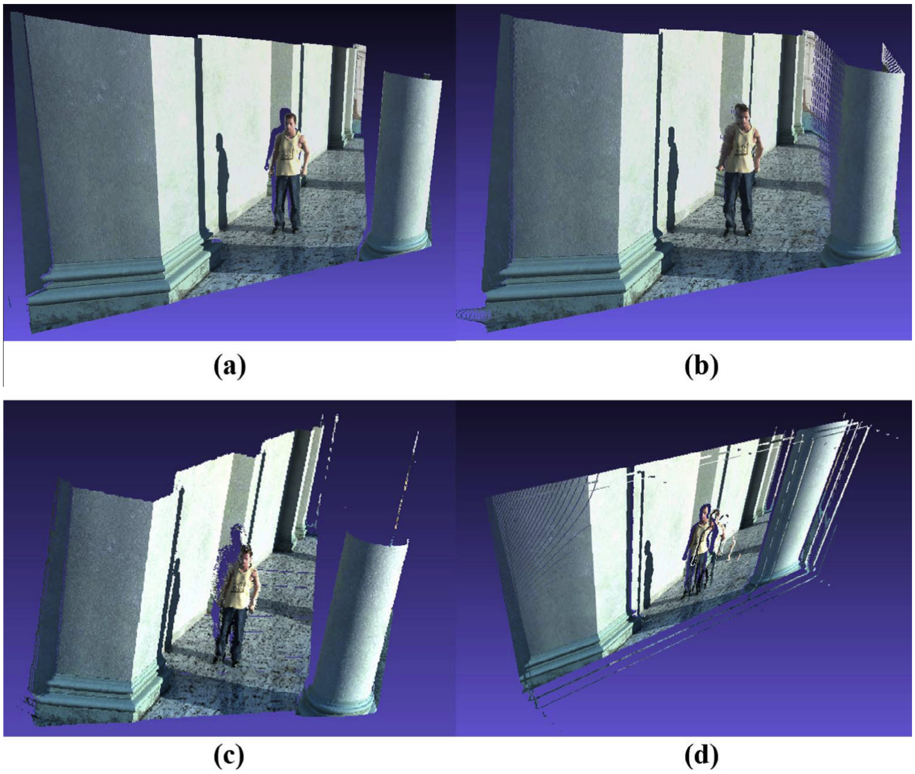

Fig. 6. Comparative results of rendered 3D images for Undo_Dancer through different depth images. (a) Rendered 3D image via the original depth image. (b) Rendered 3D image via depth image processed by Gaussian filter with 3 × 3 window size. (c) Rendered 3D point clouds via depth image processed by our proposed method. (d) Rendered 3D point clouds via depth image processed by non-local mean filter.

Table 4
The subjective quality results of synthesized dense depth maps for test sequence *Breakdancer*.

Settings	Breakdancer		
	The benchmark (dB)	The proposed (dB)	Gains (%)
Setting 1	19.2683	28.5380	32.4818
Setting 2	22.6908	28.3314	19.9094
	17.4435	27.1629	35.7817
	22.9159	26.6650	14.0600
Setting 3	24.2138	28.2067	14.1558
	20.4266	26.9096	24.0917
	17.3719	26.3974	34.1908
	14.4873	26.0616	44.4113
	16.3639	26.0237	37.1200
	19.3453	25.3096	23.5655
	23.9065	24.2261	1.3192
Average gain	25.5534		

As given by Tables 4, 5 and Fig. 7, the PSNR performance of the benchmark varies with the distance between key-time depth maps. We take the test sequence *Undo_Dancer* as an example for consequent discussion. The benchmark method is based on bi-directional propagation, and it is known that the quality performance is highly depended on the distance in propagation. Therefore, as indicated by Table 5 and Fig. 7, the PSNR performance of the benchmark method verifies the above assertion, which is not steady in the procedure of propagation. The parameter of PSNR is high when the depth image is synthesized nearby the key-time frame. As for the proposed method, the depth value of each pixel is determined by the optimal solution of multi-set graph cut model, and thus the PSNR performance keeps stable and independent from the distance between two key-time depth maps. The steady performance on objective and subjective quality of the proposed method is indicated by Fig. 7.

Table 5
The subjective quality results of synthesized dense depth maps for test sequence *Undo_Dancer*.

Settings	*Undo_Dancer*		
	The benchmark (dB)	The proposed (dB)	Gains (%)
Setting 1	29.6312	34.3336	13.6962
Setting 2	30.0450	34.1855	12.1120
	26.7632	33.6820	20.5415
	32.3701	33.1416	2.3279
Setting 3	27.8788	34.1619	18.3921
	26.9229	33.7100	20.1337
	25.3632	33.1064	23.3888
	24.0073	32.5004	26.1321
	26.3156	31.7522	17.1219
	29.5556	31.2441	5.4044
	34.8065	30.6888	−13.4176
Average gain	13.2575		

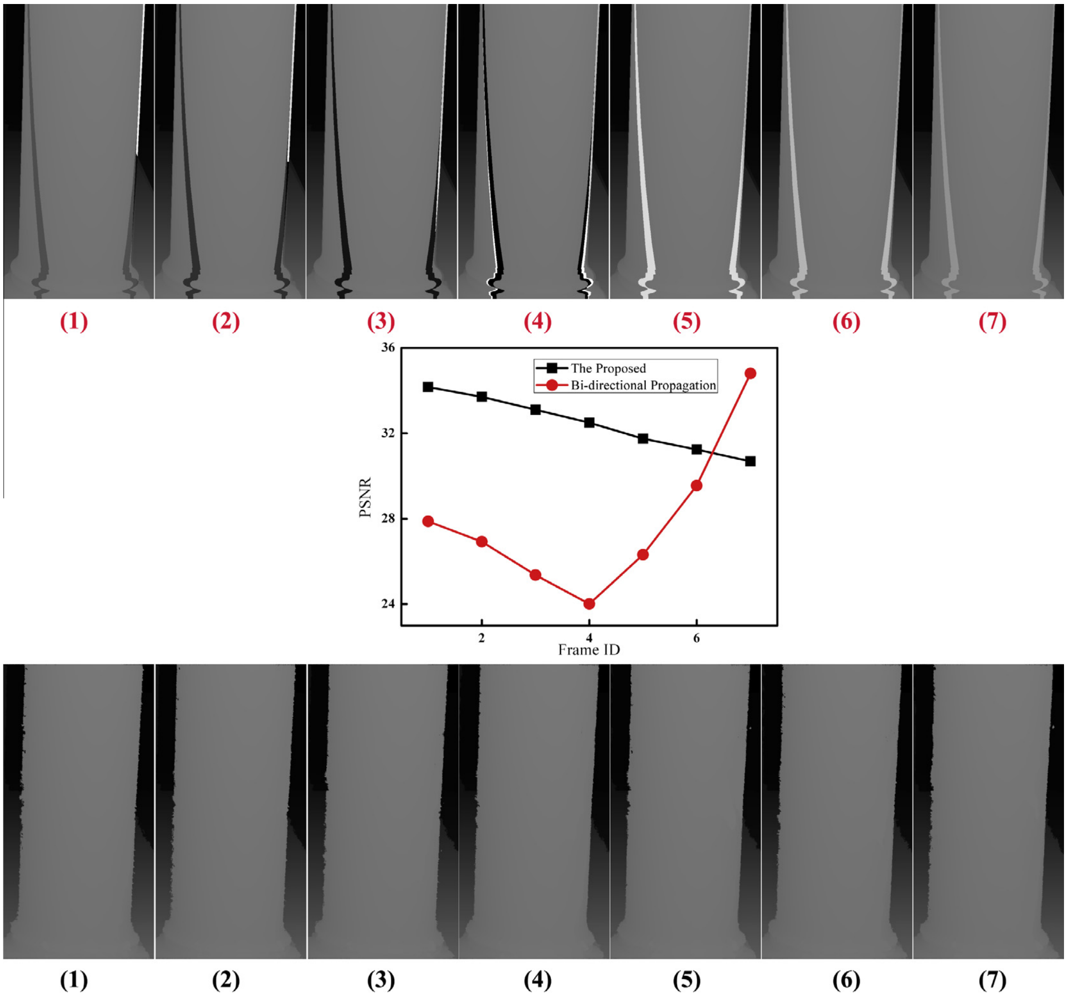

Fig. 7. Objective quality comparisons for test sequence *Undo_Dancer*.

4. Conclusions

Multiview dense depth map generation for dynamic scene is a challenge problem in the field of 3D technologies, where the challenge comes from the physical limitation of current imaging systems. In this paper, we proposed a bundled-optimization scheme to process the complete chain from depth sensing to multiview dense depth map. In this scheme, sensor noises are first detected and removed through a frequent-counting based non-linear filter before depth map up-sampling, and therefore noise amplification is prevented in the procedure of up-sampling. Blurring effect around object boundaries is the by-product of up-sampling, but it is hard to detect in the depth map. We therefore warp the depth map into 3D world coordinate system, and then propose a Blocklet based flying pixel removal method to improve the accuracy of the high resolution depth map. After that, a multi-set graph cut model is proposed to synthesize multiview dense depth map for dynamic scene. The experimental results indicate that our scheme can achieve at least 13.2575% PSNR gains when comparing to the benchmark method, and suggest the effectiveness of the proposed bundled-optimization method.

Acknowledgements

This work was supported by Natural Science Foundation of China (NSFC) (Grant Nos. 61231010, 61202207 and 61472370), and the National Key Technology Research and Development Program of China under Grant No. 2013BAH23F01.

References

[1] P. Aflaki, D. Rusanovskyy, M. Hannuksela, 3dv Sequence for Purposes of 3dv Standardization, ISO/IEC JTC1/SC29/WG11, Doc. M, 2011.

[2] F. Aulí-Llinàs, M.W. Marcellin, J. Serra-Sagrista, J. Bartrina-Rapesta, Lossy-to-lossless 3d image coding through prior coefficient lookup tables, Inf. Sci. 239 (2013) 266–282.

[3] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.

[4] A. Buades, B. Coll, J. Morel, A non-local algorithm for image denoising, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 60–65.

[5] Y. Cao, S. Zhang, Z. Zha, J. Zhang, C.W. Chen, A novel segmentation based video-denoising method with noise level estimation, Inf. Sci. 281 (2014) 507–520.

[6] W. Chen, Y. Chang, S. Lin, et al., Efficient depth image based rendering with edge dependent depth filter and interpolation, in: IEEE International Conference on Multimedia and Expo, 2005, pp. 1314–1317.

[7] J.-H. Cho, S.-Y. Kim, Y.-S. Ho, K. Lee, Dynamic 3d human actor generation method using a time-of-flight depth camera, IEEE Trans. Consum. Electr. 54 (4) (2008) 1514–1521.

[8] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, T.-S. Chua, Camera constraint-free view-based 3-d object retrieval, IEEE Trans. Image Process. 21 (4) (2012) 2269–2281.

[9] Y. Gao, M. Wang, R. Ji, X. Wu, Q. Dai, 3d object retrieval with hausdorff distance learning, IEEE Trans. Ind. Electron. 61 (4) (2014) 2088–2098.

[10] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.

[11] Y. Gao, M. Wang, Z.-J. Zha, Q. Tian, Q. Dai, N. Zhang, Less is more: efficient 3-d object retrieval with query view selection, IEEE Trans. Multimedia 13 (5) (2011) 1007–1018.

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, Rgb-d mapping: using depth cameras for dense 3d modeling of indoor environments, in: International Symposium on Experimental Robotics, 2010.

[13] B. Huhle, T. Schairer, E.A.P. Jenke, Robust non-local denoising of colored depth data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–7.

[14] Y. Jingyu, Y. Xinchen, L. Kun, H. Chunping, W. Yao, Color-guided depth recovery from rgb-d data using an adaptive autoregressive model, IEEE Trans. Image Process. 23 (8) (2014) 3443–3458.

[15] M.I. JTC1/SC29/WG11, View Synthesis Software Manual Release 3.5.

[16] J.-H. Jung, K. Hong, G. Park, I. Chung, J.-H. Park, B. Lee, Reconstruction of three-dimensional occluded object using optical flow and triangular mesh reconstruction in integral imaging, Opt. Express 18 (25) (2010) 26373–26387.

[17] A. Kolb, E. Barth, R. Koch, Tof-sensors: new dimensions for realism and interactivity, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2008.

[18] M. Kurc, O. Stankiewicz, M. Domanski, Depth map inter-view consistency refinement for multiview video, in: Picture Coding Symposium (PCS), 2012, IEEE, 2012.

[19] J.-J. Lee, B.-G. Lee, H. Yoo, Depth extraction of three-dimensional objects using block matching for slice images in synthetic aperture integral imaging, Appl. Opt. 50 (29) (2011) 5624–5629.

[20] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, B. Guo, Kinect identity: technology and experience, Computer 44 (4) (2011) 94–96.

[21] W.-N. Lie, C.-Y. Chen, W.-C. Chen, 2d to 3d video conversion with key-frame depth propagation and trilateral filtering, Electron. Lett. 47 (5) (2011) 319–321.

[22] Q. Liu, Y. Yang, Y. Gao, R. Ji, L. Yu, A bayesian framework for dense depth estimation based on spatial–temporal correlation, Neurocomputing 104 (2013) 1–9.

[23] Q. Liu, Y. Yang, R. Ji, Y. Gao, L. Yu, Cross-view down/up-sampling method for multiview depth video coding, IEEE Signal Proc. Lett. 19 (5) (2012) 295–298.

[24] Q. Liu, Z. Zha, Y. Yang, Gradient-domain-based enhancement of multi-view depth video, Inf. Sci. 281 (2014) 750–761.

[25] S. Liu, P.L. Lai, D. Tian, C. Chen, New depth coding techniques with utilization of corresponding video, IEEE Trans. Broadcast. 57 (2) (2009) 551–561.

[26] K. Lu, Q. Wang, J. Xue, W. Pan, 3d model retrieval and classification by semi-supervised learning with content-based similarity, Inf. Sci. 281 (2014) 703–713.

[27] Y. Mori, N. Fukushima, e.a.T. Yendo, View generation with 3d warping using depth information for ftv, Signal Process.: Image Commun. 24 (1–2) (2009) 65–72.

[28] L. Nie, M. Wang, Y. Gao, Z.-J. Zha, T.-S. Chua, Beyond text qa: multimedia answer generation by harvesting web information, IEEE Trans. Multimedia 15 (2) (2013) 426–441.

[29] K. Oh, S. Yea, A. Vetro, Y. Ho, Depth reconstruction filter and down/up sampling for depth coding in 3-d video, IEEE Signal Process. Lett. 16 (9) (2009) 747–750.

[30] N. Pavillon, J. Kuhn, C. Moratal, P. Jourdain, C. Depeursinge, P.J. Magistretti, P. Marquet, Early cell death detection with digital holographic microscopy, PLoS ONE 7 (1) (2012) e30912.

[31] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Int. J. Comput. Vis. 47 (1/2/3) (2002) 7–42.

[32] Z.J. Smith, K. Chu1, A.R. Espenson, M. Rahimzadeh, A. Gryshuk, M. Molinaro, D.M. Dwyre, S. Lane, D. Matthews, S. Wachsmann-Hogiu, Cell-phone-based platform for biomedical device development and education applications, PLoS ONE 6 (3) (2011) e17150.

[33] C. Varekamp, B. Barenbrug, Improved depth propagation for 2d-to-3d video conversion using key-frames, in: European Conference on Visual Media Production (IETCVMP), 2007.

[34] M. Wang, Y. Gao, K. Lu, Y. Rui, View-based discriminative probabilistic modeling for 3d object retrieval and recognition, IEEE Trans. Image Process. 22 (4) (2013) 1395–1407.

[35] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, T.-S. Chua, Event driven web video summarization by tag localization and key-shot identification, IEEE Trans. Multimedia 14 (4) (2012) 975–985.

[36] M. Wang, R. Hong, X.-T. Yuan, S. Yan, T.-S. Chua, Movie2comics: towards a lively video content presentation, IEEE Trans. Multimedia 14 (3) (2012) 858–870.

[37] M. Wang, B. Ni, X.-S. Hua, T.-S. Chua, Assistive tagging: a survey of multimedia tagging with human-computer joint exploration, ACM Comput. Surv. (CSUR) 44 (4) (2012) 25.

[38] X. Yan, Y. Yang, G. Er, Q. Dai, Depth map generation for 2d-to-3d conversion by limited user inputs and depth propagation, in: 3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-CON), 2011.

[39] Z. Yan, L. Yu, Y. Yang, Q. Liu, Beyond the interference problem: hierarchical patterns for multiple-projector structured light system, Appl. Opt. 53 (17) (2014) 3621–3632.

[40] Y. Yang, H. Deng, J. Wu, L. Yu, Depth map reconstruction and rectification through coding parameters for mobile 3d video system, Neurocomputing (in press). http://dx.doi.org/10.1016/j.neucom.2014.04.088.

[41] Y. Yang, Q. Liu, Y. Gao, B. Xiong, L. Yu, H. Luan, R. Ji, Q. Tian, Stereotime: a wireless 2d and 3d switchable video communication system, in: Proceedings of the 21st ACM International Conference on Multimedia, ACM, 2013.

[42] Y. Yang, Q. Liu, R. Ji, Y. Gao, Dynamic 3d scene depth reconstruction via optical flow field rectification, PloS ONE 7 (11) (2012) e47041.

[43] Y. Yang, Q. Liu, R. Ji, Y. Gao, Remote dynamic three-dimensional scene reconstruction, PloS ONE 8 (5) (2013) e55586.