Signal Processing: *Image Communication* 31 (2015) 34–46

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

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

Interfered depth map recovery with texture guidance for multiple structured light depth cameras

Sen Xiang [a,b], Li Yu [a,b,*], You Yang [a,b], Qiong Liu [a,b], Jialiang Zhou [a,b]

[a] *Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[b] *Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China*

ARTICLE INFO

Article history:
Received 9 June 2014
Received in revised form
12 October 2014
Accepted 12 November 2014
Available online 5 December 2014

Keywords:
Structured light depth camera
Depth recovery
Markov random field
Discrete Poisson equation
3D video

ABSTRACT

In depth acquisition systems with multiple structured light depth cameras (SLDCs), interference between the devices causes interfered regions, which degrade depth quality and impair applications. This paper proposes a novel approach to recover the depth maps. Under the guidance of texture segments, interfered regions are categorized into flat regions and boundary regions. After that, different strategies are applied to the two kinds of regions because of their property differences. For flat regions, a Markov random field (MRF) model is utilized to get the optimal gradient solution. With the gradients, discrete Poisson equation (DPE) is applied to calculate the final depth solution. In boundary regions, another texture-guided MRF is utilized to peruse depth directly. Experiment results demonstrate that the proposed method works well on both synthetic data and captured data. In flat regions, we get smooth-varied depth, and boundaries between interfered and non-interfered regions are seamless. In boundary regions, sharp depth edges between objects are well preserved. Moreover, our method improves the PSNR of synthetic data by 4–14 dB.

1. Introduction

Recently, depth-based applications such as 3DTV, virtual reality, immerse communication are developing rapidly. In these applications, accurate and complete depth information is the foundation to generate high quality viewing content. However, due to limited viewing field, depth from a single viewpoint is far from enough to reconstruct original scenes accurately and completely. To solve this problem, multiple depth maps from different positions and viewing angles are required.

In conventional frameworks, depth maps are estimated passively. In these frameworks, multiview texture maps are first captured with camera arrays. With the texture images, depth maps are calculated using local or global depth estimation algorithms [1]. Several practical camera array systems have been made by Stanford [2], MSRA [3], HHI [4], Carnegie Mellon University [5], and 3DTV program in Europe [6]. This framework can be widely applied, but the estimation process needs heavy computation, and depth cannot be generated in real-time. Another limitation of classic stereo is the reliance of good texture. The basic principle of stereo matching is to build a reliable correspondence for each pixel across stereo views. This demands that each pixel should be 'unique' to be recognized, and good texture is needed, which, unfortunately, is often not true. In scenarios of missing picture information, such as uniformly distributed texture and near darkness objects, these algorithms may fail.

While conventional stereo faces challenges, structure light depth cameras (SLDCs), such as Kinect [7], have great advantages. SLDCs apply special hardware and algorithms to acquire depth sequences in real-time. What's more, they

* Corresponding author.
E-mail address: hustlyu@mail.hust.edu.cn (L. Yu).

http://dx.doi.org/10.1016/j.image.2014.11.004
0923-5965/

project designed patterns to objects actively, and depth maps are calculated with the deformed patterns instead of object textures. The patterns are specially designed, which makes every pixel unique, and robust depth estimation is achieved. By deploying multiple such devices, depth maps from multiple viewing positions and angles are fused together [8]. Moreover, multiple depth cameras are also utilized to develop numerous applications, such as Lightspace [9], immersive 3D teleconference [10], 3D scanning [11], gesture interaction [12] and gas flow capturing [13].

However, a serious problem of SLDCs is inter-device interference. In detail, when multiple depth cameras are deployed simultaneously, the devices interfere with each other, and depth quality is degraded. The reason of interference lies in the working principle of SLDCs, which, in other words, makes it an intrinsic problem.

Currently, several approaches, which can be divided into 2 categories, are proposed to solve this problem. Solutions in the first category try to avoid or reduce the effects of influence. The most common idea is to use frequency and time division. Ho [14] used a frequency multiplexing method in a system with three time-of-flight cameras, but the approach is inapplicable for SLDCs for their working principles are different. Schroder et al. [15] and Faion et al. [16] proposed time division solutions. In their system, multiple SLDCs are strictly synchronized and each of them worked in its own time slot. This scheme avoids interference, but it also reduced the frame-rate of depth sequences. Maimone and Fuchs [17] and Butler et al. [18] proposed shaking/movement-based methods. In their schemes, depth cameras moved or shook in different manners. The relative motion blurred interference patterns from other SLDCs, thus easing interference effect. Nevertheless, this movement-based scheme is inapplicable to applications like free viewpoint video, which needs accurate camera parameters to generate virtual views. Moreover, all these frameworks were based on hardware changes, which is inconvenient. Wang et al. [19] proposed a method to reconstruct lost depth from patterns using plane-sweeping [20]. In this method, the entire space was divided into a set of hypothesis planes, and plane-sweeping tested each of them to choose the optimal one. The method reconstructed geometry at virtual views, but the original captured depth maps are not restored.

The other kind of solutions is not specified to interference problem, but applicable to it. These approaches consider influenced depth maps as grayscale images and improve their quality. A generalized framework [21] was proposed to solve a range of image-based graphic problems. The framework approximated global regularization problem, which is computation costly, with a fast iterative joint filtering operation. This framework can deal with applications such as disparity estimation, depth up-sampling and saliency detection. Expanding this framework to stereo views, the authors proposed a new scheme to acquire and enhance depth maps [22]. Another recent work focusing on depth denoising and completion is [23]. This scheme trained a Markov random field (MRF) model to determine the depth layer of each pixel, with which image denoising and completion were achieved. Nevertheless, in this work, both offline training and online training were needed.

Although many factors degrade depth quality for SLDCs, interference leads to severe results. We focused on this topic and proposed a gradient-based approach in our previous work [24], which aimed at restoring lost depth in planar objects. In detail, gradients of interfered pixels are first recovered using local statistic results, and the successive step utilized discrete Poisson equation (DPE) to calculate depth results. Nevertheless, this primary work focused on planar structure, and the solutions under more complicated scenarios, such as object boundaries, are still to be studied.

This paper proposes a novel scheme to solve the interference problem. SLDCs use designed patterns to replace texture maps in depth estimation, which brings it robust performances, but also drawbacks. In detail, although designed patterns provide unique features for stereo, object texture, which also provide useful information, is ignored. From this point, the proposed scheme takes texture map into consideration. First, under the guidance of texture segments, interfered depth pixels are classified into flat regions and boundary regions. Flat regions consist of smoothly varied depth values inside, while in boundary regions sharp edges between objects exist. For the two kinds of regions, different strategies are applied to them. In flat regions, gradient values are recovered using a MRF model first. After that, DPE is applied to calculate the depth values. By using DPE, depth values in the flat regions are smooth, and seamless with the surrounding non-interfered region. In boundary regions, sharp changes in gradient domain cannot be recovered, and the method for flat regions is not applicable. The proposed method uses another MRF model to peruse depth values directly. Texture information is applied in the model to distinguish different objects, and sharp edges between depth layers are preserved.

The main contribution of this paper is as follows:

1. This paper analyzes the reason and influence of interference among multiple SLDCs. Interference causes depth loss rather than incorrect depth values, which brings convenience to depth recovery.
2. This paper classifies interfered regions into flat and boundary regions, and applies different strategies to them for they have different properties. We use a MRF model and DPE in flat regions to get smooth depth values, and another MRF model, which takes texture information into consideration, is applied in boundary regions to preserve sharp depth edges.
3. Experimental results prove that the proposed method performances well on with both synthetic and captured data. Our method improves the subjective quality of interfered depth maps obviously. Moreover, the improvements on PSNR values of synthetic depth maps are up to 4–14 dB.

The remainder of this paper is organized as follows. Section 2 analyzes the interference problem. After that, the proposed method is described in detail in Section 3. Section 4 presents the results of the proposed method

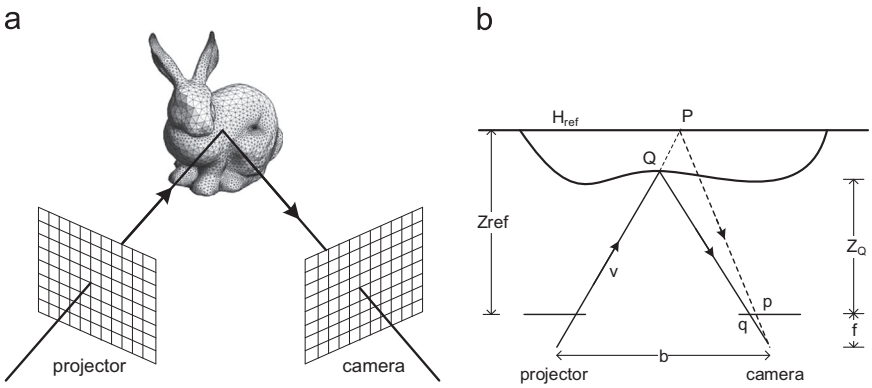

Fig. 1. Sketch of structured light depth cameras (SLDCs). (a) Setup of SLDC and (b) principle of SLDCs.

with both simulated and captured data. Finally, the conclusion is given in Section 5.

2. Problem analysis

2.1. Reason of interference

As shown in Fig. 1(a), the SLDC unit consists of a projector and a camera, which forms a stereo vision system. According to the principle of SLDCs [25], pre-defined patterns are projected from the projector and illuminate objects. From the camera's view, these patterns are deformed, and the relation between object depth and pattern deformation is known. Based on this principle, depth can be derived from pattern deformation.

Fig. 1(b) illustrates the principle of SLDCs in detail. When manufactured, a pre-defined pattern is projected against a reference plane H_{ref} with known depth Z_{ref}, and the captured pattern I_{ref} is recorded in the firmware of the SLDC. When this SLDC works, the projected pattern is exactly the same, but the captured pattern I_{cap} is deformed. In Fig. 1(b), a light ray $\vec{v}$ reaches plane H_{ref} at point P, and reflects to p in I_{ref}. When an object is placed, $\vec{v}$ reaches the object at point Q, and the projection in I_{cap} is point q. Using triangulation, depth of Q is calculated as

$$Z_Q = \frac{b_l f \cdot Z_{ref}}{b_l f + disparity \cdot Z_{ref}} \tag{1}$$

where b_l is the baseline, and f is the focal length. $disparity = |p-q|$ is the vector connecting p and q. To get Z_Q accurately, the most crucial step is to determine disparity. In other words, given q in I_{cap}, the corresponding pixel p in I_{ref} should be detected accurately.

When a single SLDC works, the correspondence is quite clear, and disparity can be detected accurately. However, if multiple SLDCs are deployed simultaneously, each camera captures patterns from different projectors. The reference pattern I_{ref} in the firmware is unchangeable, while the captured I_{cap} is a mixed pattern from different projectors. This contradiction decreases the similarity between I_{cap} and I_{ref}, and leads to failures in disparity matching, which finally causes depth loss. In fact, this problem is intrinsic for SLDCs. As long as I_{ref} is unchangeable while I_{cap} is interfered, depth degradation will always happen.

Fig. 2 illustrates the interference problem with a checkerboard. When a single SLDC works, I_{cap} is shown in Fig. 2(a) and high quality depth map is in Fig. 2(d). After an additional SLDC is deployed, the captured pattern, illustrated in Fig. 2(b), becomes much brighter. Fig. 2(c) shows the absolute difference between the two captured patterns. We find that the interference is quite obvious, and exists everywhere in the pattern. This is because the two SLDCs are toed-in mounted, and their field-of-views (FoVs) are largely overlapped. The interfered depth map is shown in Fig. 2(e), which suffers obvious quality degradation compared with Fig. 2(d). A point should be noticed is that, owing to the designed pattern and robust depth calculation algorithm, many pixels in the overlapped FoV still get their depth values. But for those pixels with severe interference, their depth values are lost such as the black pixels in rectangle 'B'. We should point out that, since the newly deployed SLDC is mounted at the projector's side of the first SLDC, some occluded background pixels are illuminated by neither projectors, and no valid depth values are presented. An example of occluded pixels are 'A' in Fig. 2(d) and (e).

The source of interference lies in the basic fundamental principle of SLDCs, and hardware-based methods also have different drawbacks. This promotes us to analyze the influence of interference, and propose a novel method.

2.2. Influence of interference on depth

As analyzed before, interference causes depth quality degradation. To look into the influence of interference on depth maps, we compare the depth maps in Fig. 2(d) and (e), and the results are exhibited in Fig. 3.

The comparison is made between the histograms of Fig. 2(d) and (e). In Fig. 3(a) and (b), depth values are concentrated at three intervals, corresponding to invalid pixels whose depth values are zero, checkerboard, and background, respectively. The difference histogram, Fig. 3(c), has a positive peak and a negative peak. The positive peak corresponds to invalid pixels, and the negative one corresponds to depth values of the checkerboard. This implies that depth values of the checkerboard are replaced by zero, rather than other incorrect values. This difference also coincides with the black pixels in box 'B' in Fig. 2(e). At the same time, the frequencies of other depth values are almost zero in Fig. 3(c), proving that interference makes no changes to other pixels.

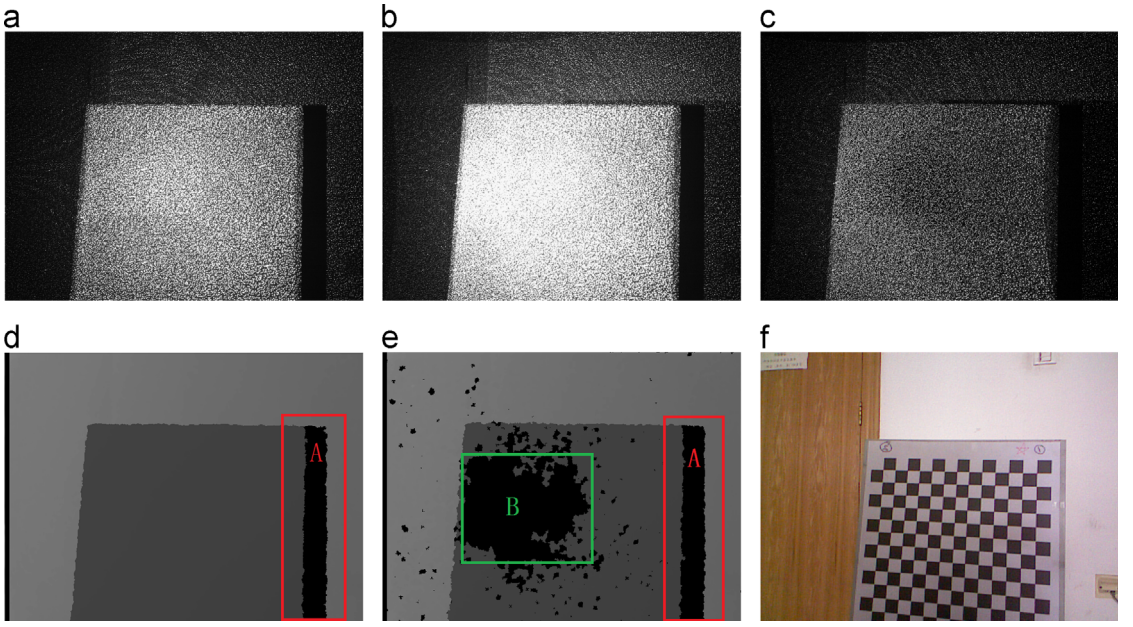

Fig. 2. Captured maps for interference between two SLDCs. (a) Captured pattern without interference, (b) depth map without interference, (c) absolute difference between (a) and (b), (d) non-interfered depth map, (e) interfered depth map, and (f) color map.

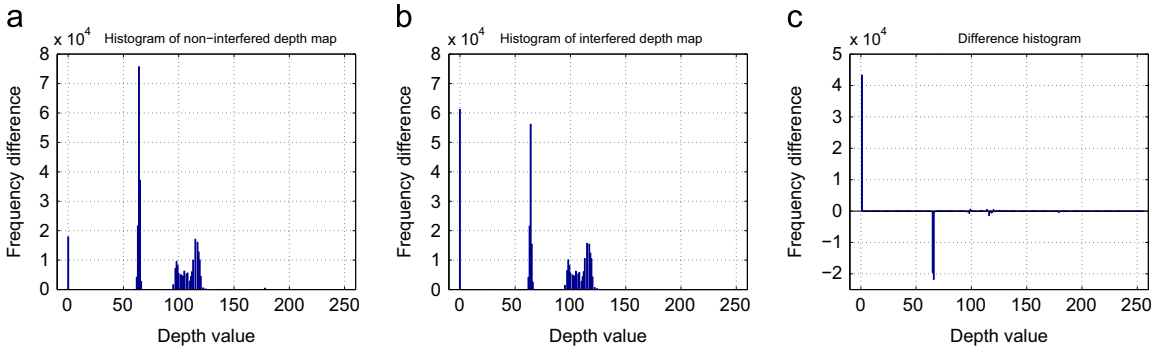

Fig. 3. Histogram of depth value distribution. (a) Histogram of non-interfered depth map. (b) Histogram of interfered depth map. (c) Difference between (b) and (a).

According to the analysis, both occlusion and interference causes invalid pixels, whose depth values are zero, rather than incorrect depth. This property, in fact, brings great convenience to our processing. To eliminate the effect of interference, we only need to process those invalid pixels. At the same time, many other reasons, such as occlusion and specular surfaces, also cause zero-depth pixels. The phenomenon is quite the same with interference, and the proposed scheme can also apply to these problems.

These invalid pixels are grouped into different disconnected regions, and we classify them into 2 categories according to their location and property differences.

1. Flat regions: These regions locate inside objects and are caused by interference. Depth values in these regions change smoothly without discontinuities.
2. Boundary regions: These regions cross different objects and are caused by occlusion or interference. They often cover different depth layers and have depth discontinuities inside them.

After identifying the category of each region, we can apply different methods according to their properties.

3. Proposed interference cancellation method

Fig. 4 illustrates the flowchart of the proposed method. First, we take the texture map as reference to divide interfered regions into flat regions and boundary regions. After that, the two kinds of regions are processed differently due to their property differences. For flat regions, a MRF model is established to recover gradient values, followed by depth value calculation using DPE. In boundary regions, depth values are modeled by another texture-guided MRF model, and depth values are restored with well-preserved object boundaries.

3.1. Region classification

As we analyzed before, interfered regions locate at different parts in depth images, and different strategies

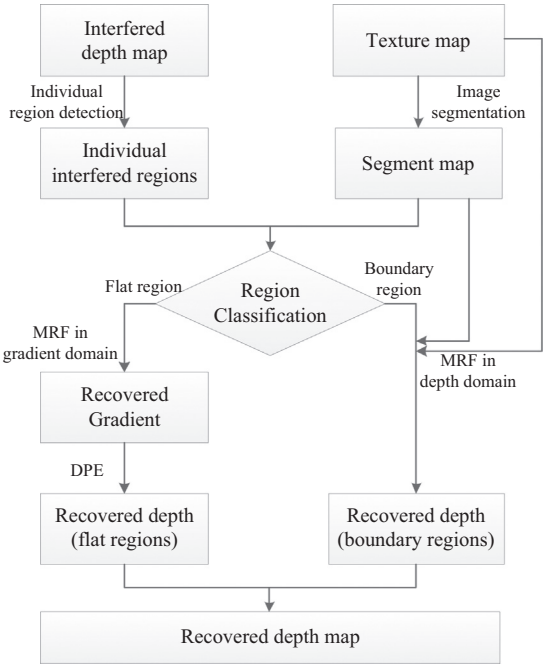

Fig. 4. Flowchart of the proposed method.

should be applied. Invalid depth pixels provide no depth information, and they form many disconnected regions. For each of them, the first step is to judge whether it is a flat region or a boundary region. In the proposed scheme, this is achieved with texture segments.

SLDCs calculate depth with patterns instead of nature textures of objects. This technique makes the performances robust and independent with object appearances, but texture information is ignored in the system. In fact, since depth and texture maps are different descriptions of a same scene, they are highly related. A basic assumption is that pixels sharing a same color also have similar depth values. Although this is not always true, it is still a useful and reasonable approach to distinguish objects in texture maps. For example, in depth estimation [26], the initial disparities are estimated in texture segments, where pixels with similar intensity share a same disparity. The proposed approach is also based on this assumption, and the detailed procedure is shown in Fig. 5.

Texture maps are first segmented with [27], and pixels with similar colors are grouped into a same segment. At the same time, each interfered region is detected in the depth map. In the example in Fig. 5, pixels are grouped into segments 'A', 'B' and 'C' in texture map, while in the depth map, individual interfered regions, R_1 and R_2, are detected. By using the segments, category of each interfered region can be determined. In detail, if an interfered region locates in a single segment, the pixels are believed to locate inside a same depth plane, and the region is considered to be a flat region. On the contrary, when a region covers two or more segments, it is likely to cross object boundaries and be treated as a boundary region. In Fig. 5, all pixels in R_1 belong to segment 'A', and R_1 is a flat regions. R_2 belongs to both 'B' and 'C', and it is a boundary

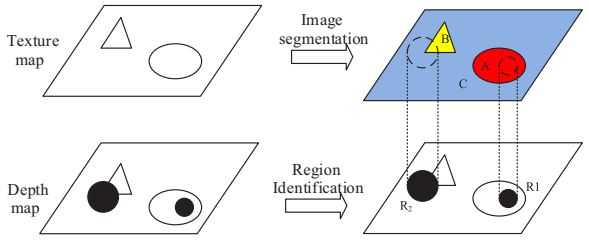

Fig. 5. Sketch for region classification.

region. After each interfered region is classified, different strategies are applied to them.

We should also point out that, image segmentation is a useful tool to distinguish objects with different colors. But in some cases, segmentation cannot be performed accurately. One example is pixels with quite different depth values may still have similar colors, and these pixels are likely to be grouped into a same segment. In these cases, the incorrect segments may lead to errors in recovered depth. In our experimental results, the failure cases will be further discussed.

3.2. Interference cancellation for flat regions

Depth values inside flat regions are smoothly varied, and can be denoted as a linear function [28]:

$$D(x,y) = a + bx + cy, \tag{2}$$

where $D(x,y)$ is the intensity of pixel (x,y), a, b, and c are constraint parameters, in which b and c are gradients along X and Y directions, respectively.

In flat regions, depth values vary smoothly, and their gradients are highly related. In other words, the gradient value for each pixel can be derived from adjacent pixels, which can be modeled with a MRF. According to this assumption, depth values in flat regions are recovered in two steps. First, a MRF model is established in gradient domain to get gradient solution of those pixels. After that, with the recovered gradients, the second step calculates the lost depth values using DPE.

3.2.1. MRF model in gradient domain

We denote the current flat region as Ω_f, and the source region, which consists of valid depth pixels, as Φ. A graph can be constructed for Ω_f. In the graph, the node set consists of all pixels in Ω_f, and the edge set consists of edges connecting each pixel p to its four nearest neighbors $\mathcal{N}_p$. $L_G = \{l_g^1, l_g^2, \ldots, l_g^n\}$ is the set for label candidates. The MRF model demands that each node is only related with its neighbors, and the energy function is defined as

$$E^G = \sum_{p_i \in \Omega_f} E^G_{data}(g_i) + \sum_{p_i \in \Omega_f} \sum_{p_j \in \mathcal{N}(p_i)} \lambda_{ij} E^G_{smoothness}(g_i, g_j) \tag{3}$$

The data term E^G_{data} denotes the cost of labeling gradient value g_i to node p_i, and the smoothness term $E^G_{smoothness}$ is the cost of consistency between pixel p_i and its neighbor p_j. The

candidate values of g_i and g_j are in L_G. E^G_{data} is defined as

$$E^G_{data}(g_i) = (g_i - g_i^{nn})^2,$$

where g_i^{nn} is the gradient value of the nearest pixel in Φ. Since depth values vary smoothly inside flat regions, g_i^{nn} can be used as a prediction of g_i.

Similarly, the smoothness term $E^G_{smoothness}$ is defined as

$$E^G_{smoothness}(g_i, g_j) = (g_i - g_j)^2,$$

where g_i and g_j are gradients of node p_i and p_j, respectively. The smoothness term demands the consistency between neighboring pixels, which implies that adjacent pixels have similar gradient values.

We want to get the optimal label solution of gradient values for the node set, and the optimal solution minimize the energy function:

$$\mathbf{g}^* = \arg\min_{\mathbf{g}} \left\{ \sum_{p_i \in \Omega_f} (g_i - g_i^{nn})^2 + \sum_{p_i \in \Omega_f p_j \in \mathcal{N}(p_i)} \sum \lambda_{ij}(g_i - g_j)^2 \right\} \tag{4}$$

where $\mathbf{g}$ is the gradient solution for all pixels in Ω_f.

This problem is NP-hard, and usually can be solved using iteration algorithms. In iterations, parameter λ_{ij} controls the contribution of $E^G_{smoothness}$. In detail, for current pixel p_i, if the neighbor p_j is invalid, λ_{ij} equals to 0, otherwise, it is assigned to 1. After each round of iteration, statuses of all pixels are updated. An invalid pixel is considered to be a valid one if it has at least one valid neighbor. This guarantees that valid gradient information propagate from source pixels to interfered ones.

The MRF model is applied to both horizontal and vertical gradients. Using these recovered gradients, depth for pixels in Ω_f can be derived.

3.2.2. Optimal depth derivation using DPE

With the obtained gradients in last step, depth values can be calculated using DPE. Assume the inner boundaries of Ω_f and Φ are $\partial\Omega_f$ and $\partial\Phi$, respectively. Poisson equation is described as

$$\nabla^2 d_{\Omega_f}(x,y) = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}, \quad \text{s.t. } d_{\Omega_f}(x,y)\big|_{\partial\Phi} = d_{\Phi}(x,y)\big|_{\partial\Phi}, \tag{5}$$

where ∇ is the Laplacian operator, G_x and G_y are horizontal and vertical gradients.

The discrete form of Poisson equation, known as DPE, is written as

$$\sum_{q_i \in \mathcal{N}_p} d(q_i) - 4d(p) = \frac{\partial G_x(p)}{\partial x} + \frac{\partial G_y(p)}{\partial y} \tag{6}$$

where p is the current pixel being processed, and $\mathcal{N}_p$ denote the four nearest neighbors of p. When q locates inside Ω_f, such as q_1 in Fig. 6, its depth value is an unknown factor in the equation. While if it belongs to $\partial\Phi$, such as q_4 in Fig. 6, it is utilized as a known boundary condition. Specifically, the equation for Fig. 6 is established as

$$\sum_{q_i \in \{q_1, q_2, q_3\}} d(q_i) - 4d(p) = \frac{\partial G_x(p)}{\partial x} + \frac{\partial G_y(p)}{\partial y} - d(q_4) \tag{7}$$

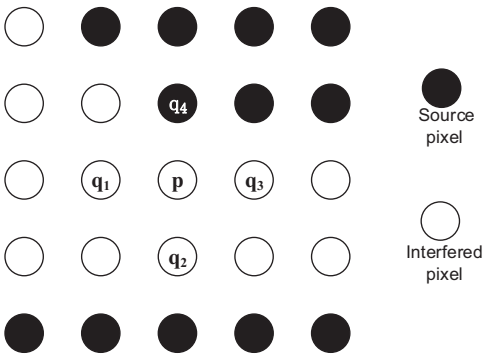

Fig. 6. Examples of the two situations for discrete Poisson equation. Among the neighbors of p, q_4 locates in the source region, the others are in the interfered region.

All terms on the left side are unknowns and known factors are on the right. Assume that the interfered region Ω_f has n_{Ω_f} pixels, we establish an equation for each pixel according to Eq. (6), and these equations form a matrix equation:

$$\mathbf{A}\mathbf{d}_{\Omega_f} = \mathbf{b}, \tag{8}$$

where $\mathbf{d}_{\Omega_f}$ is the vector of unknowns. $\mathbf{A}$ is the coefficient matrix, and $\mathbf{b}$ is the vector of known factors. The optimal solution

$$\mathbf{d}^*_{\Omega_f} = \mathbf{A}^+\mathbf{b}, \tag{9}$$

minimizes the error $\|\mathbf{A}\mathbf{d}_{\Omega_f} - \mathbf{g}\|$, in which $\mathbf{A}^+$ is the pseudo-inverse matrix of $\mathbf{A}$.

DPE benefits our results in two aspects. On one hand, it guarantees that the solution follows the obtained gradients, which keeps the inner structures of objects. On the other hand, due to the boundary condition, the boundaries between flat regions and source regions are seamlessly restored.

3.3. Interference cancellation for boundary regions

Boundary regions cover different planes and sharp depth transitions exist in these regions. Different from the situation in flat regions, the gradient values on edges are isolated from other pixels, and the condition of MRF is not satisfied. Since the sharp gradient changes cannot be recovered, the method in Section 3.2 is inapplicable.

For boundary regions, since they cover different depth layers, the most crucial point is to preserve depth edges. The proposed method uses a segment-based MRF model to label depth values in boundary regions directly, in which texture segments are used to exclude the influences between objects. Depth values are labeled only referring to pixels in the same segment, which preserves sharp depth edges between objects.

Like the matter in Section 3.2, for each boundary region Ω_b like R_2 in Fig. 5, an undirected graph can be constructed. The node set contains all pixels in Ω_b, and the edge set consists of edges connecting each pixel to its four neighbors. $L_D = \{l_d^1, l_d^2, \ldots, l_d^N\}$ is the set for label candidates.

Like Eq. (3), and the energy function is defined as

$$E^D = \sum_{p_i \in \Omega_b} E^D_{data}(g_i) + \sum_{p_i \in \Omega_b p_j \in \mathcal{N}(p_i)} \sum \beta_{ij} E^D_{smoothness}(d_i, d_j) \tag{10}$$

where E^D_{data} is the data term and $E^D_{smoothness}$ is the smoothness term. d_i and d_j are labeled depth values, which belong to L_D, of p_i and p_j, respectively. However, since these pixels do not have observed depth values, the data term E^D_{data} is meaningless, and the energy function is simplified to

$$E^D = \sum_{p_i \in \Omega_b} \sum_{p_j \in \mathcal{N}(p_i)} E^D_{smoothness}(d_i, d_j), \tag{11}$$

where $E^D_{smoothness}$ is defined as

$$E^D_{smoothness}(d_i, d_j) = (d_i - d_j)^2$$

The weighting parameter β_{ij} can be calculated as

$$\beta_{ij} = f(p_j) s(p_i, p_j) \exp(-|c(p_i) - c(p_j)|)$$

$f(p_j)$ equals to 1 only if p_j is a valid pixel, otherwise it is set to 0. Like the case in gradient domain, after every round of iteration, an invalid pixel can update to a valid one if it has at least one valid neighbor. $s(i,j)$ equals to 1 only when p_i and p_j are in a same segment. This prevents mixed depth values from different depth planes, and keeps clear and sharp object edges. We convert the texture map into grayscales, and $c(p_i)$ and $c(p_j)$ are gray levels of p_i and p_j. Since the pixels with similar grayscales should contribute larger to the weighting factor, β_{ij} is reverse related to the absolute difference of $c(p_i)$ and $c(p_j)$.

The optimal solution minimizes the energy function

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \sum_{p_i \in \Omega_b} \sum_{p_j \in \mathcal{N}(p_i)} (d_i - d_j)^2 \tag{12}$$

where $\mathbf{d}$ is the gradient solution for all pixels in Ω_b. The solution can be achieved by optimization methods.

3.4. Optimization

Since it is a NP-hard problem to solve Eqs. (4) and (12), optimization method is needed to solve the problem. We use iterated conditional modes (ICM) [29] to pursuit the solution for its simplicity. After initializing a configuration for all the variables $\mathbf{g}$ in Eq. (4) or $\mathbf{d}$ in Eq. (12), this algorithm iterates over each pixel in the image. For each pixel i, given the current states of its neighborhoods g_j or d_j ($j \in \mathcal{N}(i)$), the value minimizing the energy is calculated and recorded. At the end of each iteration, the optimal values for all pixels get updated with the recorded results, and a new iteration begins. With the iteration goes further, the results converges, and we get desired results.

4. Experimental results

4.1. Experiment setup

To verify the proposed method under both ideal and practical conditions, we carry out the experiments with both synthetic and captured data. In implementation, SLDCs and objects are placed as shown in Fig. 7.

To get synthetic data, two SLDCs are simulated with Pov-Ray, which allows us to arrange scenes arbitrarily. Two SLDCs are parallel settled locating 28 unit lengths from each other, and the distance between projector and camera in each SLDC is 7.2 unit lengths. With the projected and captured patterns, depth maps are calculated using MRCC proposed in [19], and the threshold of MRCC is set to 0.5 to get correct depth.

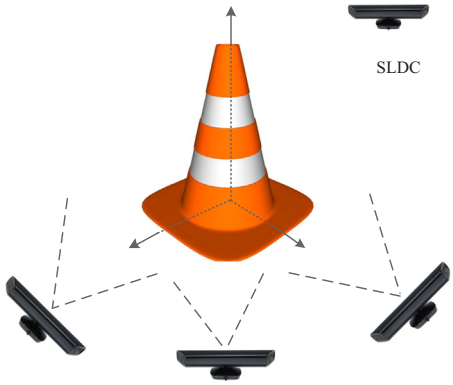

Fig. 7. Sketch of multiview depth acquisition with multiple SLDCs.

We also capture depth and color data with real SLDCs. Two Kinects from Microsoft, providing both color and depth maps at VGA resolution, are fixed to a shelf, locating 30 cm to each other. RGBDemo [30] is utilized to capture both texture and depth data, and is quantized into grayscales linearly.

In stereo vision, it is very likely to have errors at depth discontinuities, which is an intrinsic drawback. To further improve the depth quality, before depth recovery, these unreliable pixels are firstly detected. For each depth edge pixel p, we simply find the nearest texture edge pixel q, and points between them are believed to be unreliable. These pixels are likely to have incorrect depth values, and may even impair depth recovery. We set their depth to zero and calculate reliable values for them in the successive steps. Although this step enlarges the holes in depth maps, final depth maps get more accurate edges.

4.2. Evaluation of interference cancellation

4.2.1. Results on synthetic data

The results on synthetic data are shown in Fig. 8. Five scenes 'Ball', 'Bottle', 'Cones', 'Table' and 'Teapot' are designed. To compare with the state-of-the-art methods, we generate virtual depth maps as [19] did. All experiments are performed on these virtual maps rather than the two original interfered depth images. For each scene, we present the images of every necessary step. Our results are compared with the plane-sweeping approach [19] and our previous work [24].

Texture and interfered virtual depth maps are in Fig. 8(a) and (b). A point should be mentioned is that, in virtual view synthesis, depth from the two original views gets merged, thus making interfered regions in Fig. 8(b) sparse. In fact, the original two depth maps have denser interfered regions.

Results of [19] are illustrated in Fig. 8(c) and (d). Compared with input depth maps, invalid pixels get their depth values and the visual quality also gets improved. However, there are two shortcomings in their results. On one hand, the results have many noisy points, which are incorrect depth values. On the other hand, invalid pixels on

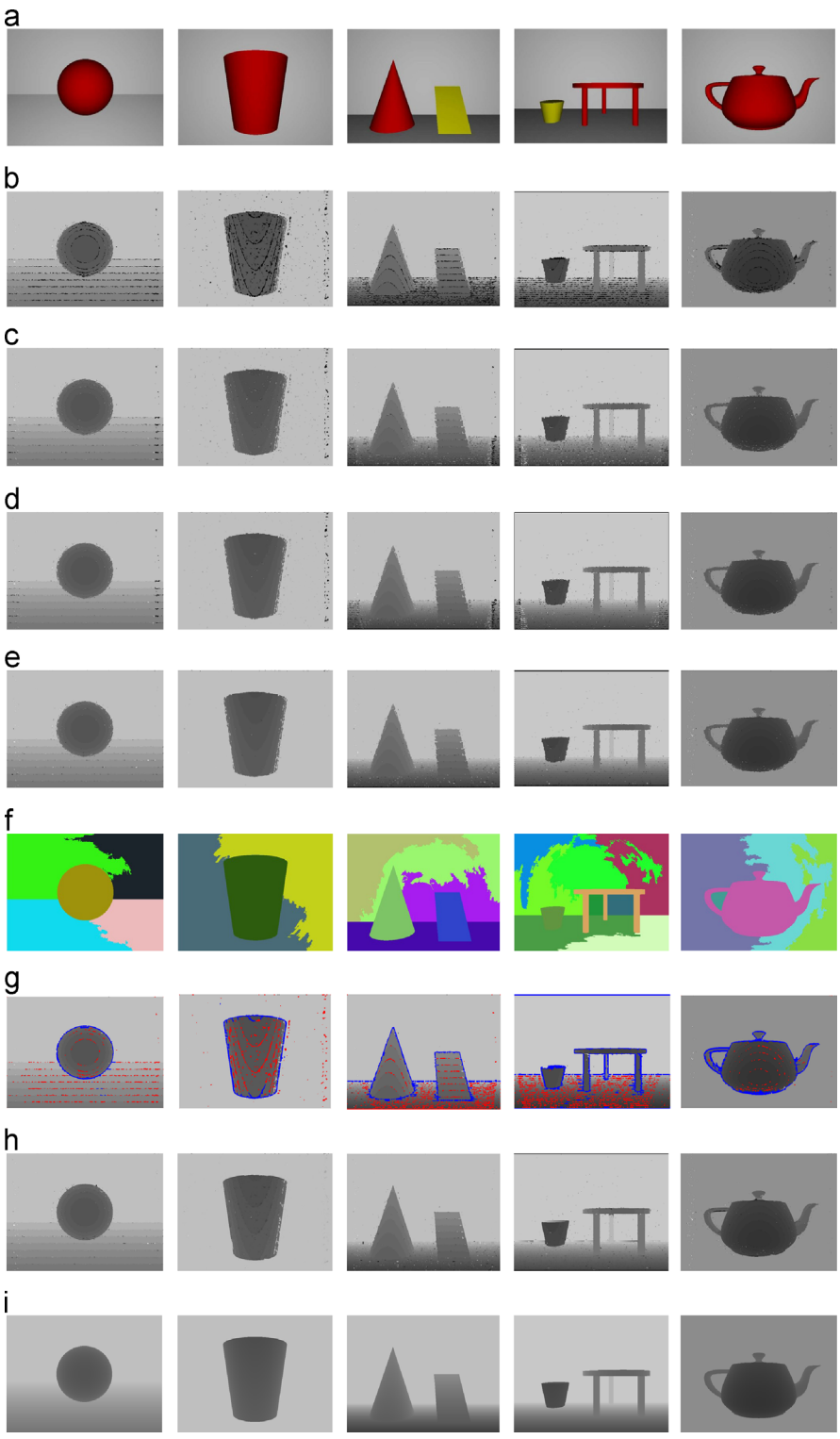

Fig. 8. Results on synthetic data. Columns from left to right are scene 'Ball', 'Bottle', ' Cone', 'Table', and 'Teapot'. Rows from top to bottom are (a) texture maps, (b) interfered depth maps, (c) (d) results of [19] under camera observation constraint and projector–camera constraint, respectively, (e) results of [24], (f) segmentation results, (g) flat and boundary regions, (h) results of the proposed method, and (i) ground-truth depth maps.

image borders are not restored. This is because [19] needs to find corresponding points for every invalid pixel in the views of both SLDCs, but those pixels near image borders do not satisfy such a condition and get no results.

Fig. 8(e) presents the results of our previous work [24]. This work focuses at depth loss of planar structure, but it fails to keep the sharp edges at object boundaries, where gradients change abruptly. The planes in the images are recovered quite well, but at object boundaries, we can find intermediate depth values between the two depth planes.

Results of the propose method are presented in Fig. 8 (f)–(h). Under ideal conditions, different objects have their unique colors in texture maps, which benefits accurate image segmentation. The segments are presented in Fig. 8(f), where each color represents a segment. Fig. 8(g) shows flat and boundary regions, which are in red and blue, respectively. Under the guidance of accurate image segmentation, the final results are generated and shown in Fig. 8(h). Compared with [19], the proposed method improves depth quality, even for pixels near image borders. Moreover, our method generates less noisy pixels for the MRF model use neighboring samples as reference, thus avoiding outliers. As to [24], the proposed scheme, utilizing texture segments as guidance, achieves better performance at object boundaries. The ground-truth depth images shown in Fig. 8(i) provide a benchmark for subjective quality evaluation, and our results are closer to ground-truth.

Fig. 9 presents two zoomed parts of 'Table' and 'Teapot'. Results of the proposed scheme are more accurate and sharper at depth discontinuities. The reason is that the proposed scheme uses segment and texture information as guidance to recover boundary regions. Meanwhile, unreliable pixel detection, the pre-processing, also excludes some incorrect boundary pixels, and benefits edge preserving.

The objective results are shown in Table 1. PSNR values of the interfered depth maps are rather low because of the large amount of invalid pixels. [19,24] improves PSNR, but the results are still not satisfying. The proposed method provides a 4–14 dB gain on the interfered depth maps. For 'Table', 'Bottle', 'Cone' and 'Ball', the proposed method brings PSNR gains on existing methods. For 'Teapot', three methods have almost equivalent qualities.

4.2.2. *Results on captured data*

The proposed method is tested with real SLDCs. Four scenes 'Bag', 'Doll', 'Desk' and 'Boy', are used in the experiments. The former two scenes are quite simple, while the latter two are complex. Compared with synthetic data, the

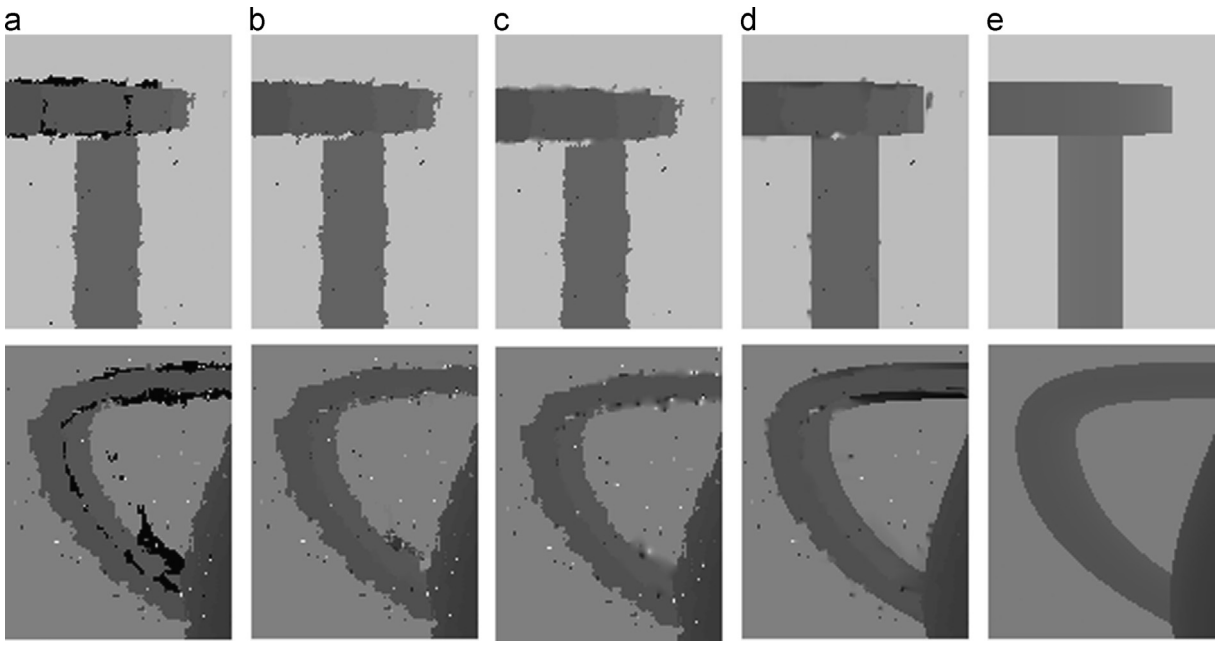

Fig. 9. Details of edges in scene 'Table' and 'Teapot'. (a) Interfered depth maps, (b) results of [19] under projector–camera constraint, (c) results of [24], (d) results of proposed method, and (e) ground-truth depth maps.

Table 1
PSNR of the results of different methods on synthetic data.

Methods	Ball (dB)	Bottle (dB)	Cone (dB)	Table (dB)	Teapot (dB)
Interfered depth map	21.0290	20.7752	20.3300	19.1772	26.7694
Camera observation constraint in [19]	27.4118	24.6312	25.2160	24.4303	29.5431
Projector–camera constraint in [19]	28.0322	25.3113	25.9841	26.1518	30.0962
[24]	31.1544	28.6653	29.4258	28.7401	**30.4399**
Proposed method	**34.2487**	**34.412**	**33.0134**	**30.2637**	30.0548

experiments on captured data are more difficult to perform, for many practical factors, such as lighting conditions and lens distortions, degrade image quality. Moreover, richer texture and denser interfered regions are also challenges. Since the proposed method does not make any changes to hardware, the results are compared with the shake-based solution [18] and our previous work [24], neither of which made changes to the inner structure of SLDCs.

Figs. 10–13 illustrate the experimental results. Texture and interfered depth maps are in the first two columns in each figure. Compared with the synthetic data, the interference problem is much severer, and depth quality is worse.

The third columns show the results of [18]. This approach improves depth quality obviously, but for some severely interfered pixels, this scheme still fails. Results of [24] are shown in fourth columns. Like the cases in the synthetic data, this approach works well for regions inside planar surfaces, but for those regions cover difference depth planes, there exist many blurred pixels. In some scenes, this problem is quite obvious. For example, in Fig. 12(d) and (j), sharp corners of the monitor are disappeared. Instead, the regions are filled with blurred pixels.

The last three columns illustrate our results, which are color-labeled segments, region classification results and

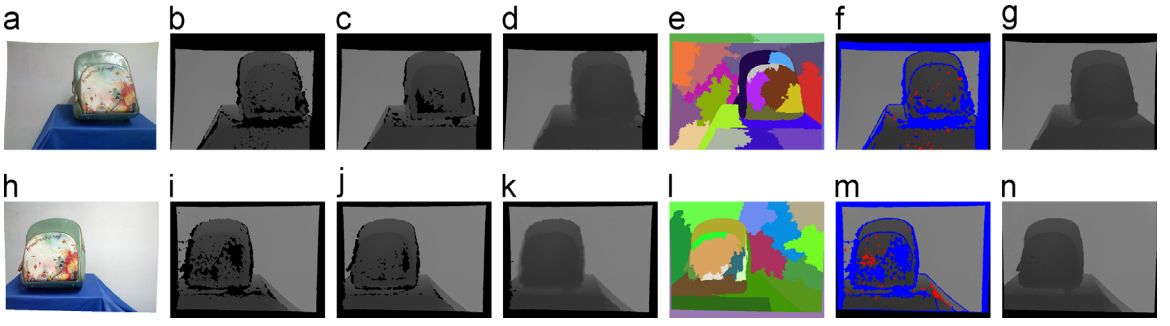

Fig. 10. Result on captured scene 'Bag'. (a), (h) Color maps, (b), (i) interfered depth maps, (c), (j) results of motion-based approach [18], (d), (k) results of [24], (e), (l) segmentation results, (f), (m) flat regions and boundary regions, and (g), (n) results of the proposed method.

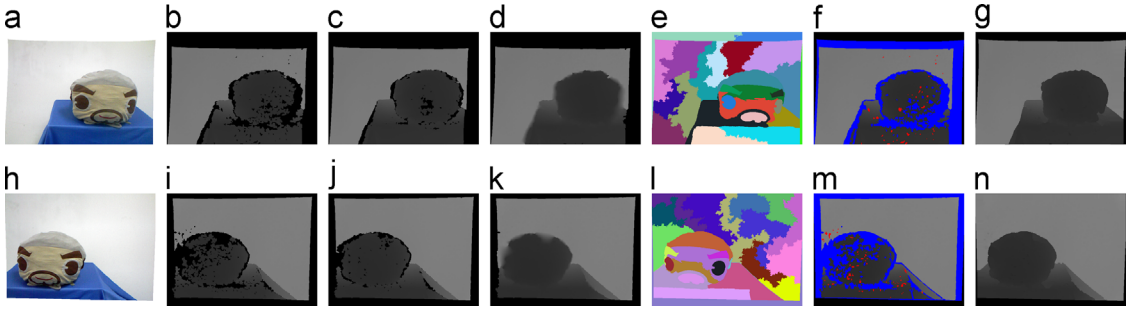

Fig. 11. Result on captured scene 'Doll'. (a), (h) Color maps, (b), (i) interfered depth maps, (c), (j) results of motion-based approach [18], (d), (k) results of [24], (e), (l) segmentation results, (f), (m) flat regions and boundary regions, and (g), (n) results of the proposed method.

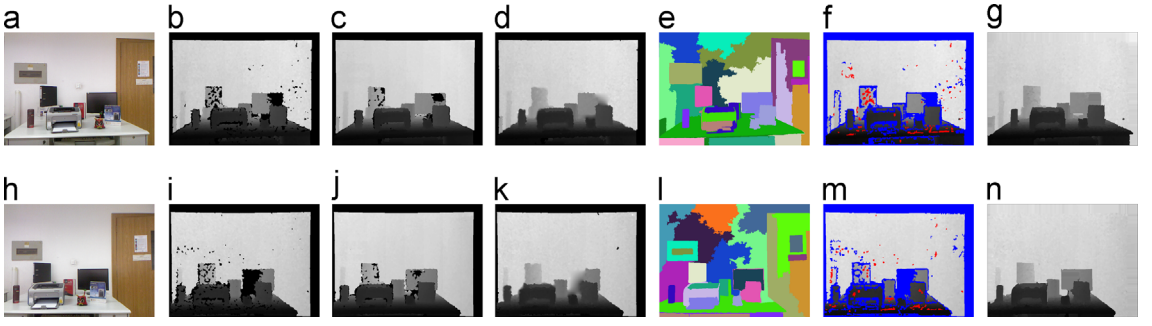

Fig. 12. Result on captured scene 'Desk'. (a), (h) Color maps, (b), (i) interfered depth maps, (c), (j) results of motion-based approach [18], (d), (k) results of [24], (e), (l) segmentation results, (f), (m) flat regions and boundary regions, and (g), (n) results of the proposed method.

recovered depth maps, respectively. In the fifth columns, segments are filled with different colors. Although there are some errors, the segments are generally accurate. With segmentation results, interfered regions are classified into flat ones and boundary ones, which are labeled with red and blue, respectively. Finally, interfered regions are recovered with valid depth and the results are illustrated in the last columns. The depth quality is obviously improved in both flat and boundary regions. In flat regions, depth values are restored smoothly. By using DPE, results near edges between interfered and non-interfered regions are seamless. In boundary regions, depth edges are well preserved because segment maps are used to separate objects from each other and prevent depth mixture of different depth planes.

4.2.3. Failure case analysis

Although the proposed method performs well, there are some failure cases. To demonstrate them clearly, we present an example in Fig. 14.

An important factor influencing our results is texture segmentation. Accurate segments benefit our results. However, image segmentation itself is still to be further studied, and incorrect segments are unavoidable. For example, when foreground and background objects have similar colors or even share a same color, they are very likely to be classified into a same segment, thus leading to incorrect results. In Fig. 14, foreground and background objects in rectangles A, C, D and E are all white, and texture segmentation failed to distinguish them. Finally, the object boundaries are distorted. Besides the original object colors, many factors, such as bad lighting conditions and low textured occlusions, also impair accurate object segmentation, and lead to failure cases.

Moreover, the quality of input depth map is also very important. For instance, it is difficult to get depth of slim objects like the aloe in Fig. 14. Unfortunately, interference even worsens the problem, and the aloe is severely distorted in Fig. 14(b) and (g). Although its color is quite different from the background to pursue accurate segments, the recovered depth is still in poor quality.

4.3. Efficiency analysis

Besides performance, efficiencies of algorithms are also very important. The hardware based approach [18] does not introduce any extra computation. While for the proposed scheme, a detailed complexity analysis is presented.

4.3.1. Complexity of the proposed method

As shown in Fig. 4, the proposed method includes several main steps: (1) image segmentation, (2) region classification, (3) depth recovery for flat regions, and (4) depth retrieval for boundary regions. Assume the depth resolution is $W \times H$,

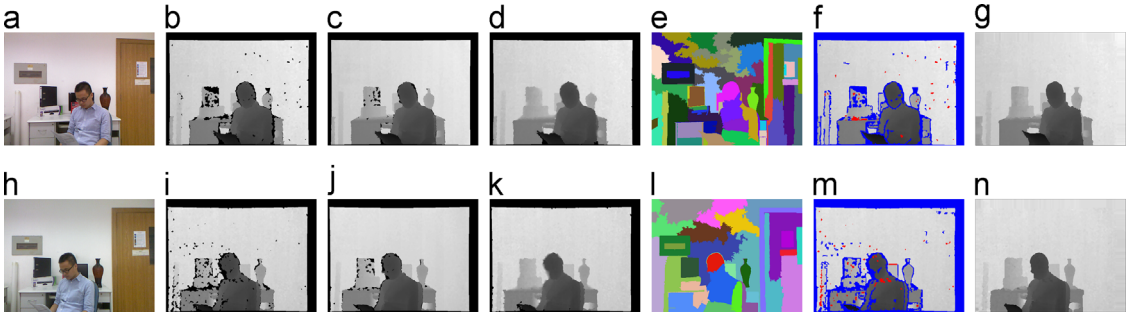

Fig. 13. Result on captured scene 'Boy'. (a), (h) Color maps, (b), (i) interfered depth maps, (c), (j) results of motion-based approach [18], (d), (k) results of [24], (e), (l) segmentation results, (f), (m) flat regions and boundary regions, and (g), (n) results of the proposed method.

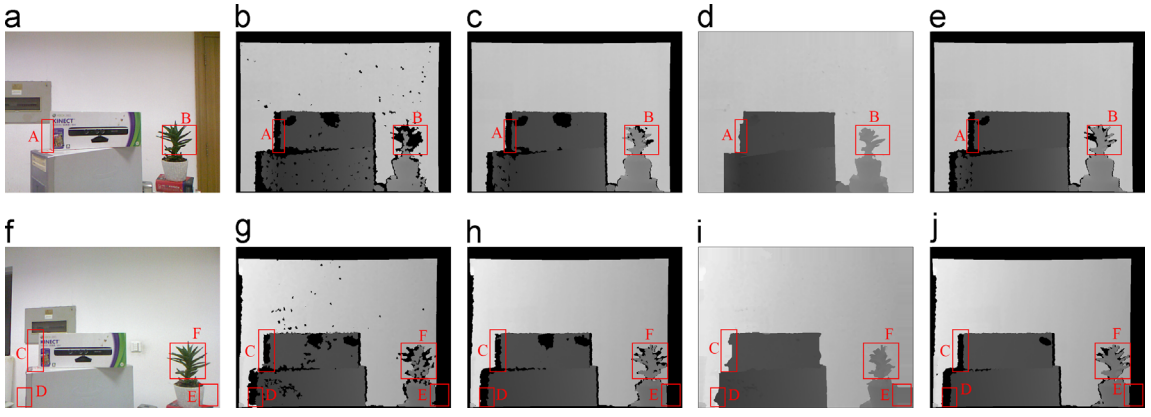

Fig. 14. A failure case. (a), (f) Color maps. (b), (g) depth maps with interference, (c), (h) results of motion-based approach [18], (d), (i) results of the proposed method and (e), (j) depth maps without interference.

and interfered regions have $N=\gamma WH$ pixels. αN pixels locate in flat regions, and the rest $(1-\alpha)N$ pixels are in boundary regions. The computation load should cover all these steps, which will be analyzed in detail.

- According to [27], the computation of image segmentation is $O(WH\log(WH))$, which is independent of the sizes of interfered regions.
- In region classification, we check pixels in every interfered region. In the best case, two pixels are checked before the region is classified to a boundary region. While in the worst case, we need to check all pixels in a region to confirm that it is a flat region. In general, the complexity of this step is $O(N)$.
- In depth recovery for flat regions, there are three substeps. First, the results in Eq. (4) are to be solved. According to ICM, the complexity of optimization is $O(\alpha N)$. Second, DPEs are estimated for all pixels in flat regions. Since each pixel has an equation established according to Eq. (6), the complexity is $O(\alpha N)$. Finally, for each flat region, we need to compute solution based on Eq. (9), which involves matrix inversion of $\mathbf{A}$ and multiplication between $\mathbf{A}^{+}$ and $\mathbf{b}$. Assume $\mathbf{A}$ is a m-by-m matrix, and $\mathbf{b}$ is a m-by-1 vector, the complexities of matrix inversion and multiplication are $O(m^3)$ and $O(m^2)$, respectively. Since the number of flat regions is uncertain, the exact complexity is uncertain, but depends on the distribution of these regions. In the best case, there exists αN flat regions, each with a single pixel, and the complexity is $O(\alpha N)$. In the worst case, all αN pixels gather into one, and the complexity is $O((\alpha N)^3)$.
- In the step to retrieve depth in boundary regions, we only need to get the solution for Eq. (12). Like the case in gradient optimization, the complexity of this step is $O((1-\alpha)N)$ by using ICM.

The overall complexity takes all of the four steps into account. Specifically, the complexities of step (1)(2) and (4) are all in order of $O(N)$, while for the third step, it is related to the distribution of flat interfered regions. When these pixels are more sparsely located, the complexity decreases to lower orders. On the contrary, if they converge together and assemble into several large flat regions, higher orders of complexity are necessary. Although the exact overall complexity is uncertain, we can still get the upper and lower bounds. Considering the extreme cases as analyzed in step (3), the overall complexity is between the lower bound $O(N)$ and upper bound $O(N^3)$.

4.3.2. Complexity comparison with existing approaches

In [19], the plane sweeping algorithm tests hypothesis planes for each pixel and choose the optimal one. Since the number of candidate planes is constant, the overall complexity is $O(N)$. Compared with [19], the proposed scheme generates better results at the cost of higher complexities.

In [24], the whole procedure includes gradient statistics, calculation, and depth retrieval using DPE. In the first two steps, the complexities are all $O(N)$. In the last step, the situation is the same with step (3) in the proposed method, which lies between $O((\alpha N))$ and $O((\alpha N)^3)$. In general, the overall complexity is of the same order of the proposed new scheme. In both methods, high order complexities are caused by DPE. In [24], DPE is applied to all interfered pixels. While in the proposed scheme, it is only applied to flat regions, which, in fact, reduces the total computation load.

5. Conclusions

This paper proposed a scheme for interfered depth recovery in systems with multiple SLDCs. By using texture segments as guidance, the proposed method classified interfered regions into flat and boundary regions, after that, different strategies were applied to them based on their own properties. In flat regions, lost depth was recovered with a MRF model and DPE. While in boundary regions, another MRF model, guided by texture map, was applied to get depth directly. Experiments were performed on both synthetic data and captured images, and the depth quality was remarkably improved. In flat regions, depth values were smooth and coincide with surrounding valid pixels seamlessly. In boundary regions, sharp object edges were well-preserved, which kept the original geometry information. As to objective quality, the proposed method improved the PSNR values of synthetic data by 4–14 dB. What's more, the proposed scheme does not need any hardware changes to SLDCs, which makes it applicable in practical use.

Acknowledgments

This work was supported by National Science Foundation of China (Nos. 61231010 and 61202301) and the Fundamental Research Funds for the Central Universities (HUST no. 2012QN076).

References

[1] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Int. J. Comput. Vis. 47 (1–3) (2002) 7–42.
[2] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, High performance imaging using large camera arrays, ACM Trans. Graph. 24 (2005) 765–776.
[3] J.-G. Lou, H. Cai, J. Li, A real-time interactive multi-view video system, in: Proceedings of ACM International Conference on Multimedia, 2005, pp. 161–170.
[4] A. Smolic, D. McCutchen, 3DAV exploration of video-based rendering technology in MPEG, IEEE Trans. Circuits Syst. Video Technol. 14 (3) (2004) 348–356.
[5] R.T. Collins, O. Amidi, T. Kanade, An active camera system for acquiring multi-view video, in: Proceedings of IEEE Conference on Image Processing, vol. 1, 2002, pp. I–527.
[6] L. Onural, T. Sikora, A. Smolic, An overview of a new european consortium: integrated three-dimensional television—capture, transmission and display (3DTV), in: P. Hobson, E. Izquierdo, I. Kompatsiaris, N. E. O'Connor (Eds.), EWIMT, QMUL, 2004.
[7] ⟨http://www.xbox.com/en-US/kinect⟩, [Online; accessed 2-June-2014], 2014.
[8] R. Nair, K. Ruhl, F. Lenzen, S. Meister, H. Schäfer, C.S. Garbe, M. Eisemann, M. Magnor, D. Kondermann, A survey on time-of-flight stereo fusion, in: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, Springer, 2013, pp. 105–127.
[9] A.D. Wilson, H. Benko, Combining multiple depth cameras and projectors for interactions on, above and between surfaces, in: Proceedings of ACM Symposium on User Interface Software and Technology, 2010, pp. 273–282.

[10] C. Zhang, Q. Cai, P. Chou, Z. Zhang, R. Martin-Brualla, Viewport: a distributed, immersive teleconferencing system with infrared dot pattern, IEEE MultiMed. (2013) 17–27.
[11] J. Tong, J. Zhou, L. Liu, Z. Pan, H. Yan, Scanning 3D full human bodies using kinects, IEEE Trans. Vis. Comput. Graph. 18 (4) (2012) 643–650.
[12] M. Caon, Y. Yue, J. Tscherrig, E. Mugellini, O. Abou Khaled, Context-aware 3d gesture interaction based on multiple kinects, in: Proceedings of International Conference on Ambient Computing, Applications, Services and Technologies, 2011, pp. 7–12.
[13] K. Berger, K. Ruhl, M. Albers, Y. Schroder, A. Scholz, J. Kokemuller, S. Guthe, M. Magnor, The capturing of turbulent gas flows using multiple kinects, in: Proceedings of IEEE International Conference on Computer Vision Workshops, 2011, pp. 1108–1113.
[14] Y.-S. Kang, Y.-S. Ho, High-quality multi-view depth generation using multiple color and depth cameras, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2010, pp. 1405–1410.
[15] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, M. Magnor, Multiple kinect studies, Technical Report, 9-15, ICG, Technical University of Braunschewig, 2011.
[16] F. Faion, S. Friedberger, A. Zea, U. D. Hanebeck, Intelligent sensor-scheduling for multi-kinect-tracking, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 3993–3999.
[17] A. Maimone, H. Fuchs, Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras, in: Proceedings of IEEE International Symposium on Mixed and Augmented Reality, 2011, pp. 137–146.
[18] D.A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, D. Kim, Shake'n'sense: reducing interference for overlapping structured light depth cameras, in: Proceedings of ACM Annual Conference on Human Factors in Computing Systems, 2012, pp. 1933–1936.
[19] J. Wang, C. Zhang, W. Zhu, Z. Zhang, Z. Xiong, P.A. Chou, 3D scene reconstruction by multiple structured-light based commodity depth cameras, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2012, pp. 5429–5432.
[20] R.T. Collins, A space-sweep approach to true multi-image matching, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, pp. 358–363.
[21] M. Lang, O. Wang, T. Aydin, A. Smolic, M.H. Gross, Practical temporal consistency for image-based graphics applications, ACM Trans. Graph. 31 (4) (2012) 34.
[22] N. Stefanoski, C. Bal, M. Lang, O. Wang, A. Smolic, Depth estimation and depth enhancement by diffusion of depth features, in: IEEE International Conference on Image Processing, 2013, pp. 1247–1251.
[23] J. Shen, S.-C. S. Cheung, Layer depth denoising and completion for structured-light RGB-D cameras, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 1187–1194.
[24] S. Xiang, L. Yu, Q. Liu, Z. Xiong, A gradient-based approach for interference cancelation in systems with multiple kinect cameras, in: Proceedings of IEEE International Symposium on Circuits and Systems, IEEE, 2013, pp. 13–16.
[25] B. Freedman, A. Shpunt, M. Machline, Y. Arieli, Depth mapping using projected patterns, US Patent App. 11/899,542 (Oct. 2 2008).
[26] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, ACM Trans. Graph. 23 (3) (2004) 600–608.
[27] M.-Y. Liu, O. Tuzel, S. Ramalingam, R. Chellappa, Entropy rate superpixel segmentation, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 2097–2104.
[28] S.Z. Li, Markov Random Field Modeling in Image Analysis, Springer, 2009.
[29] J. Besag, On the statistical analysis of dirty pictures, J. R. Stat. Soc. Ser. B (Methodol.) (1986) 259–302.
[30] N. Burrus, RGBDemo, ⟨http://labs.manctl.com/rgbdemo/⟩, [Online; accessed 2 June 2014] (2014).