

Texture-Free Large-Area Depth Recovery for Planar Surfaces

Zengqiang Yan ^{#1}, Li Yu ^{#2}, Zixiang Xiong ^{*3}

[#] School of Electron. Inf. & Commun., Huazhong Univ. of Sci. & Tech., Wuhan, China

¹ ZengqiangYan@hust.edu.cn

² hustlyu@hust.edu.cn

^{*} Department of Electrical and Computer Engineering, Texas A&M University, College Station, USA

³ zx@ece.tamu.edu

Abstract—This paper presents a texture-free depth enhancement method for large-area depth recovery. The proposed algorithm identifies a large-area depth missing region, and iteratively segments its contour by setting different initial pixels in each iteration. Coordinate transformation is used to analyze the distribution of each contour segment. By examining distributions of all contour segments, statistical histogram analysis is applied in our approach to select contour pixels. Then, selected pixels are projected into the world coordinate system, and multiple linear regression is utilized for surface function approximation. Missing depth values of a large-area depth missing region can be recovered with guidance of the approximated surface function. Quantitative and qualitative evaluations over state-of-the-art depth enhancement methods demonstrate the effectiveness and superiority of our method. Being texture-free, the proposed method has the flexibility of being merged into traditional depth enhancement methods.

I. INTRODUCTION

Acquiring high-quality depth data has been one of the most important issues in the field of 3D computer vision and many 3D related applications such as image-based rendering [1], visual tracking [2], virtual viewpoint synthesis [3] and gesture recognition [4]. Different from texture image that can be readily captured with high quality, acquiring accurate depth maps is difficult.

Till now, depth acquiring methods can be roughly divided into two categories: passive methods [5], [6] and active methods [7], [8]. Since passive methods struggle to acquire depth information in textureless, depth-discontinuous, and occluded regions, active methods have drawn more research interests. However, this technology has its inherent problems (noisy, low-resolution, and data loss). Fig.1 shows the texture image and depth map captured by a representative active depth camera Kinect #1. In the depth map of Fig.1 (b), there exist plenty of depth missing regions with different sizes. Depth missing regions with small sizes mainly exist in boundaries and depth discontinuous regions, while large-area depth missing region exists on planar surfaces. For large-area depth missing region of planar surface, almost the whole depth values are missing,

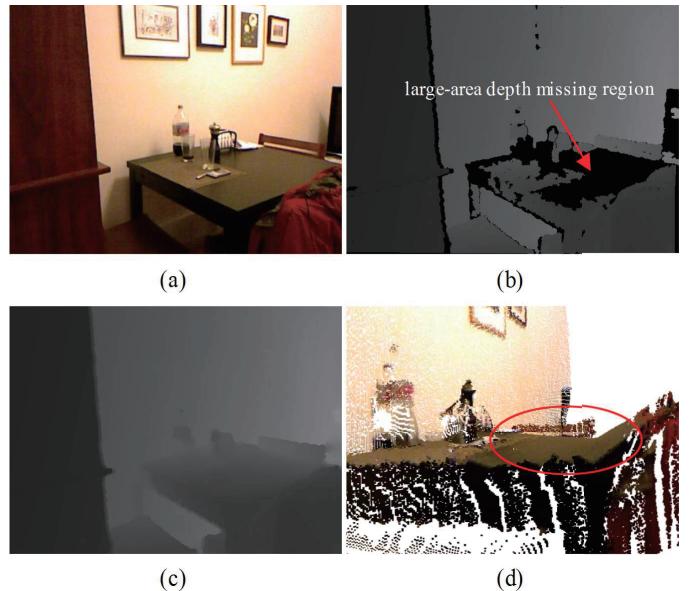


Fig. 1. RGB-D data captured by Microsoft Kinect #1 and depth map enhanced by joint bilateral filter (JBF) [9]: (a) The texture image, (b) Captured depth map, (c) Depth map enhanced by JBF and (d) Details of the point cloud constructed based on (a) and (c).

and only depth values of pixels around the boundaries can be obtained as shown in Fig.1 (b).

To improve the quality of depth maps, many depth enhancement methods have been proposed. Most of the works on depth enhancement can be grouped into two categories as follows:

- **Depth inpainting:** In [9], Kopf *et al.* applied an iterative joint bilateral filter through the help of the texture image. Yu *et al.* [10] proposed refining noisy depth map in the framework of shape-from-shading. Miao *et al.* [11] proposed a texture assisted Kinect depth inpainting framework by analyzing local spatial properties in both depth map and texture image. In [12], Lu *et al.* assembled similar RGBD patches into a matrix and used its low-rank estimation to enhance the depth map. In [13], Liu *et al.* used a heat diffusion framework to conduct inpainting and upsampling for defective depth maps with the guidance of texture image.
- **Depth super-resolution:** Min *et al.* [14] proposed a

weighted mode filtering method for super-resolution and denoising through the help of high-resolution texture image. Kim *et al.* [15] presented a method to enhance depth images spatially and temporally. Yang *et al.* [16] proposed an example-based 2D image super-resolution approach using sparse signal representation. In [17], Yang *et al.* used color image guided autoregression prior to select neighbor pixels for depth recovery and super-resolution.

Current depth enhancement methods can effectively recover missing depth values of small regions, but struggle to address the large-area depth missing problem of planar surfaces. Fig.1 (c) shows the depth map processed by the joint bilateral filter (JBF) in [9]. In Fig.1 (c), depth missing regions with small sizes can be effectively recovered, but the recovered depth values of the large-area depth missing region are inaccurate. We further construct a point cloud based on texture image in Fig.1 (a) and depth map in Fig.1 (c). As marked by the red ellipse in Fig.1 (d), constructed surface is an irregular surface, not a uniform plane.

In this paper, we design depth recovery method to solve the large-area depth missing problem especially for planar surfaces. Different from most depth enhancement methods that used texture information as guidance, we propose a texture-free method for the following reasons: (i) Resolutions of the texture image and depth map might be different, and super-resolution methods can introduce artifacts, (ii) Performance of texture-assisted methods badly decreases especially when different objects share similar texture information.

Considering that neighbor pixels for large-area depth missing region inpainting are quite limited, we propose to recover depth values through the help of its surface function. The main component of our method is to select appropriate pixels for surface function approximation. We first identify the large-area depth missing region in depth map and generate contour map of the region. Secondly, contour pixels are iteratively segmented with different initial pixel in each iteration. For each segment, we project its contour pixels into the world coordinate system, and use the multiple linear regression method to approximate the distribution of these contour pixels. Thus, we can obtain an approximated surface function for each segment. Thirdly, statistical histogram analysis is applied to approximated surface functions of all the contour segments, and contour pixels corresponding to the peaks in histograms are selected as candidates. Finally, we select appropriate contour pixels by refining contour candidates, and approximate the surface function of large-area depth missing region based on these pixels. Missing depth values can be recovered with guidance of the approximated surface function.

II. PROPOSED FRAMEWORK

We aim to recover the depth information of large-area depth missing regions in a depth map, without using corresponding texture image as guidance. Our framework consists of four main steps: (i) Identification of the large-area depth missing

region and extraction of contour map, (ii) Iterative contour segmentation with random initialization, (iii) Candidates selection and refinement, (iv) Depth recovery and further processing. In the sequel, R represents the large-area depth missing region.

A. Large-area Depth Missing Region Identification and Contour Extraction

In this stage, we identify the large-area depth missing region R and extract the contour map of R . To identify the region R , we convert the depth map in Fig.2 (a) into a binary image and select the largest connected component. Since the depth map is noisy, different depth missing regions might be connected as in Fig.2 (b). In Fig.2 (b), the white region is the largest connected component directly selected based on the depth map. Thus, before selecting the largest connected component, we separate connected regions when their connectivity is below a fixed threshold (set as 5). Then the largest connected component after refinement is selected as in Fig.2 (c). In Fig.2 (c), most wrong depth missing regions are removed, but there still exist some wrong regions. Noisy pixels from these wrong regions can be removed based on the method in the following subsection. Fig.2 (d) shows the outer contour of the largest connected component in Fig.2 (c), and the depth values of contour pixels are known and can be used for further processing.

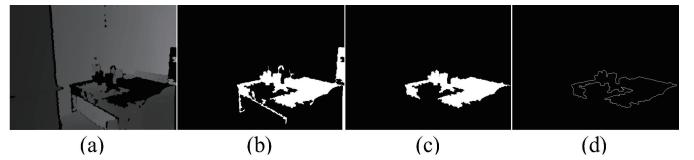


Fig. 2. Large-area depth missing region identification and contour exaction: (a) Depth map, (b) Largest connected component, (c) Identified large-area depth missing region and (d) Contour map of identified region in (c).

B. Iterative Contour Segmentation

In general, there exist many wrong pixels in the contour map, and not all contour pixels can be used for surface function approximation. In our method, we iteratively divide the contour into different segments (usually 200 iterations), and select appropriate contour pixels by applying the statistical histogram analysis method.



Fig. 3. Segmentation results of contour map in three different iterations. In each image, pixels belong to different segments are assigned with different gray values.

Fig.3 shows the segmentation results in different iterations. We set different starting pixels to conduct the segmentation in different iterations. As a result, in Fig.3, segmentation results

are different in different iterations. Then for each segment, we project the contour pixels into the world coordinate system by

$$d(u, v) \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix}, \quad (1)$$

where (u, v) is the coordinates of a contour pixel in the contour map, $d(u, v)$ is the depth value of pixel (u, v) , P is the projection matrix of depth sensor that can be obtained by the calibration algorithm [18], and (x_W, y_W, z_W) represents the coordinates in the world coordinate system. Since P is determined by the camera parameters, more accurate P will generate better depth enhancement results in our method.

Based on the multiple linear regression method in [19], the relationship between z_W and (x_W, y_W) of each contour segment can be represented by

$$Y = X\beta^s + \varepsilon, \quad (2)$$

with

$$Y = \begin{pmatrix} z_W^1 \\ \vdots \\ z_W^n \end{pmatrix}, X = \begin{pmatrix} 1 & x_W^1 & y_W^1 \\ \vdots & \vdots & \vdots \\ 1 & x_W^n & y_W^n \end{pmatrix}, \quad (3)$$

$$\beta^s = (\beta_0^s \quad \beta_1^s \quad \beta_2^s)^T, \varepsilon = (\varepsilon_1 \quad \dots \quad \varepsilon_n)^T,$$

where (x_W^i, y_W^i, z_W^i) is the coordinates of i -th contour pixel, β^s denotes the parameters of the approximated surface function, ε represents the error vector and n denotes the length of the contour segment. Then, we can obtain the coefficient vector β^s of the contour segment by

$$\beta^s = (X^T X)^{-1} X^T Y, \quad (4)$$

and the approximated surface function can be written by

$$z_W = \beta_0^s + \beta_1^s x_W + \beta_2^s y_W. \quad (5)$$

For segments in all iterations, we calculate the approximated surface functions, and select the contour pixels for depth recovery based on the method described below.

C. Candidates Selection and Refinement

In this subsection, we calculate the statistical histograms of approximated surface functions for all contour segments, and generate candidate set by selecting contour pixels corresponding to the peaks of histograms. For effective candidate selection, the approximated surface function of each segment is assigned with a weight to make the distributions more concentrated. Relying on the Gaussian distribution, the weight of the i -th contour segment is given by

$$W(i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(w_i - \mu)^2}{2\sigma^2}\right), w_i \in (0, 1], \quad (6)$$

where $\sigma = \mu = 1$, and w_i represents the accuracy of the approximated surface function of the i -th segment. Since w_i is in $(0, 1]$, $W(i)$ is monotonically increasing with w_i . Thus,

weights of contour segments belong to the large-area depth missing region are higher than the weights of contour segments from wrong regions, and peaks can be effectively formed in the statistical histograms. w_i in (6) is calculated as

$$w_i = \frac{1}{e^L - 1} \left[\exp\left(\frac{L}{1 + \sum_{(x,y) \in S_i} \varepsilon(x, y)}\right) - 1 \right], \quad (7)$$

where S_i is the contour pixel set of the i -th segment, L represents the length of the segment, and $\varepsilon(x, y)$ is the approximation error of contour pixel (x, y) obtained by

$$\varepsilon(x, y) = \|z'_W(x, y) - z_W(x, y)\|. \quad (8)$$

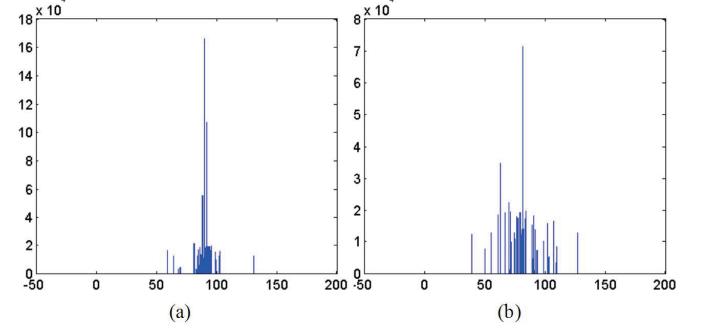


Fig. 4. Statistical histograms of contour segments. For each contour segment, we calculate the angles between its approximated surface function and X_WOZ_W and Y_WOZ_W planes. (a) Statistical histogram of angles with X_WOZ_W plane. (b) Statistical histogram of angles with Y_WOZ_W plane.

For each segment, we calculate the angles between the approximated surface function and the X_WOZ_W and Y_WOZ_W planes. By counting the angles of all contour segments, we can obtain histograms as shown in Fig.4. Based on the histograms, contour pixels correspond to the peaks in histograms are selected as the candidate set Ω_C according to

$$\Omega_C = P_{X_WOZ_W} \cap P_{Y_WOZ_W}, \quad (9)$$

where $P_{X_WOZ_W}$ and $P_{Y_WOZ_W}$ denote the contour pixel sets corresponding to the peaks in histograms. Contour pixels selected based on histogram analysis are shown in Fig.5 (c).

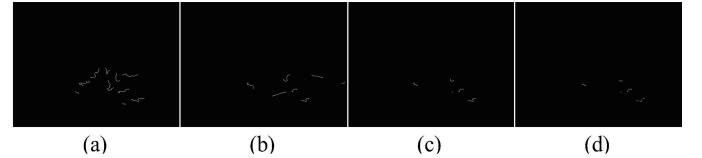


Fig. 5. (a) Contour pixels correspond to the peak of histogram in Fig.4 (a). (b) Contour pixels correspond to the peak of histogram in Fig.4 (b). (c) Candidates selected according to (9). (d) Contour pixels selected for surface function approximation after candidates refinement.

Given the candidates obtained above, we first calculate coordinates (x_W, y_W, z_W) of each candidate based on (1). Then, we can obtain the approximated surface function $z_W = \beta_0^c + \beta_1^c x_W + \beta_2^c y_W$ by applying multiple linear regression to the candidates. To improve the performance of our proposed

method, we further refine the candidates through weight assignments. Taking the i -th candidate (u, v) for example, its weight is calculated as

$$W_i = \alpha \times \text{sign}(T - \varepsilon(u, v)) \times \frac{f_i}{\sum_{j=1,2,\dots,N} f_j}, \quad (10)$$

where α is the coefficient, $\text{sign}()$ is defined as a standard sign function, f_i is the frequency of the i -th candidate, and N is the total number of candidates. Selected contour pixels after refinement are shown in Fig.5 (d).

D. Depth Recovery and Further Enhancement

In this stage, we recover the missing depth values of region R based on the selected contour pixels in Fig.5 (d). For each selected contour pixel in Fig.5 (d), we calculate its coordinate in the world coordinate system by (1). Then we approximate the surface function of region R by applying multiple linear regression to all selected pixels, and rewrite the function as

$$(\beta_1 \quad \beta_2 \quad -1 \quad 1) \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} = 1 - \beta_0. \quad (11)$$

By taking coordinates (x_W, y_W, z_W) of each pixel belonging to R into the approximated surface function, the missing depth value can be recovered as

$$d(u, v) = \frac{1 - \beta_0}{(\beta_1 \quad \beta_2 \quad -1 \quad 1) \cdot P^{-1} \cdot (u \quad v \quad 1)^T}. \quad (12)$$



Fig. 6. Depth recovery results of our proposed method. (a) Depth map enhanced by recovering large-area depth missing region based on our method. (b) Depth map by applying a median filter to depth map in (a). (c) Point cloud constructed based on depth map in (b) and its texture image.

Fig.6 (a) is the depth map enhanced by our proposed method. Comparing the depth map in Fig.6 (a) with original depth map in Fig.1 (b), we can see that the missing depth values of region R are correctly recovered and recovered depth values are smooth. Considering that our method is designed for large-area depth recovery, for small depth missing regions, other depth enhancement methods are applied, and Fig.6 (b) is the depth map by applying a simple median filter. From the point cloud in Fig.6 (c), we see that positional relationships between the recovered region and neighbor regions are correct.

III. EXPERIMENTAL RESULTS

To evaluate the performances of the proposed method, our method is evaluated on the NYU and Middlebury datasets and compared with different depth enhancement methods including

JBF [9], WMF [14] and AD [13]. For subjective evaluations, we construct point clouds based on depth maps processed by different methods, and provide details of the constructed point clouds. For objective evaluations, the quality of output depth images processed by different methods are measured by PSNR and bad pixel rate (BPR) based on the ground truth depth data.

In subsections, the sizes of local windows in median filtering and JBF are set to be 5×5 and 7×7 . In AD, the parameter Σ is set to be 10, while in WMF Σ_w , Σ_c and the size of the local window are set to be 3, 10 and 5×5 , respectively.

A. Experiments on the NYU Dataset

Fig.7 shows the results of different methods on the NYU dataset. Fig.7 (a) and Fig.7 (b) shows the texture and depth images, including details marked by red rectangles. Fig.7 (c) are the depth maps processed by JBF and details of the corresponding point clouds. In JBF, missing depth values are recovered by referring neighbor pixels with the guidance of texture information. However, for large-area depth missing regions, almost the whole depth information of the region is missing, and only a few edge pixels can be used as references. As a result, the recovered depth values are badly influenced by the neighboring pixels, and constructed point clouds of recovered surfaces are irregular surfaces, not uniform planes.

Similar to JBF, both WMF and AD face the problem of available neighbor pixels for large-area depth missing regions being too limited. From the enhanced depth maps and corresponding point clouds in Fig.7 (d) and Fig.7 (e), it is seen that both recovered depth values and constructed surface are wrong. In the second scene of Fig.7 (e), the constructed surface by AD actually is an irregular surface, not a uniform plane. By comparing depth maps and constructed surfaces of JBF, WMF and AD, we see that JBF, WMF and AD fail to recover depth values of large-area depth missing regions.

In contrast, Fig.7 (f) are depth maps and point clouds of constructed surfaces processed by our proposed method. As shown in point clouds of different scenes, the constructed surfaces are uniform planes, and depth values between recovered regions and neighbor regions are continuous.

B. Experiments on the Middlebury Dataset

In this subsection, we manually remove depth values in depth maps in Fig.8 (a), and use different methods to recover depth values. From the experimental results of JBF, WMF and AD, we see that small depth missing regions are effectively recovered. In JBF, WMF and AD, recovered depth values of large-area depth missing regions are calculated by referring neighbor pixels. However, compared with ground true depth maps in Fig.8 (f), the recovered depth values are wrong.

Fig.8 (e) shows the depth maps enhanced by our proposed method. Comparing ground true depth maps in Fig.8 (f), we see that our results are more accurate than those of JBF, WMF and AD. Compared with the depth maps in Fig.8 (f), recovered depth values from our method change more quickly, mainly due to the fact that the projection matrix P in (1) is set as a standard projection matrix of Kinect #1, not the real

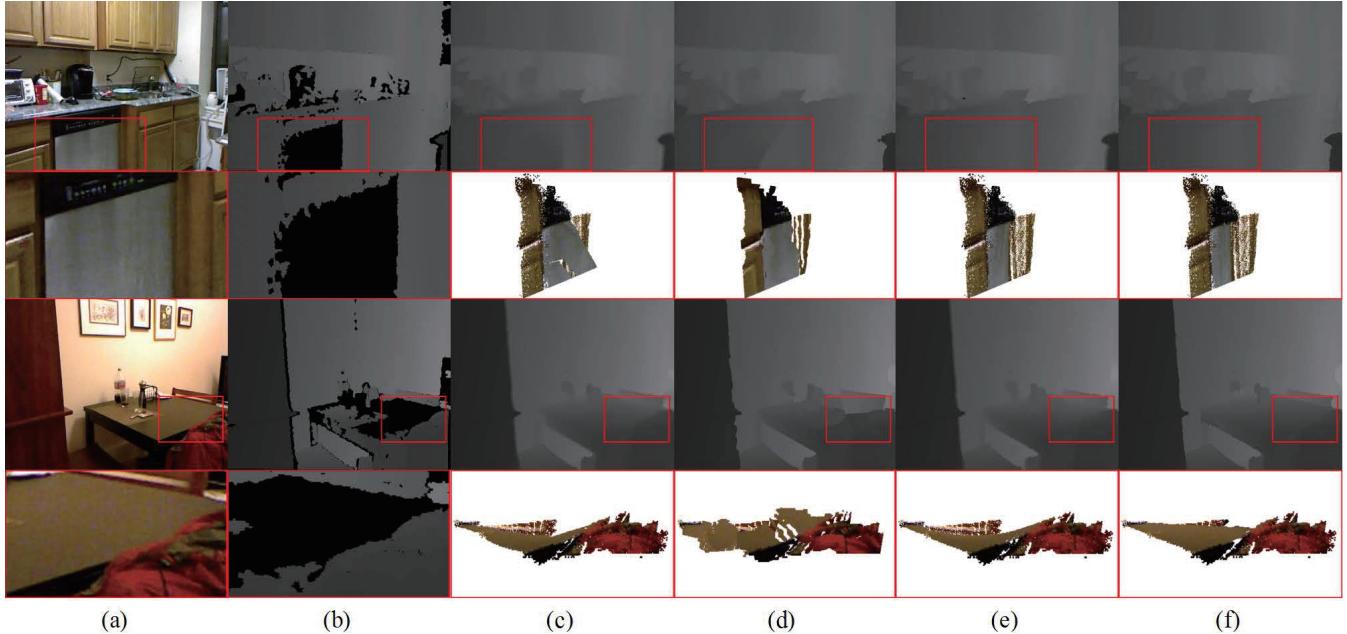


Fig. 7. Visual quality comparison for depth enhancement methods on NYU RGB-D dataset: (a) The texture image, (b) The original depth map and depth maps enhanced by (c) JBF [9], (d) WMF [14], (e) AD [13], and (f) our proposed method. The second, fourth and sixth rows are details of each image. For each depth map enhanced by different methods, we provide the point cloud corresponding to the details labeled by red rectangle.

projection matrix of the depth sensor. Using camera calibration algorithms to obtain a more accurate projection matrix P can further improve the performance.

Quantitative results are provided by calculating peak-signal-to-noise ratio (PSNR) and bad pixel rate (BPR) as shown in Table I and Table II. For PSNR comparison, we first use our method to recover depth values of large-area depth missing regions, and apply other traditional depth enhancement methods for small depth missing regions. Then the obtained results are compared with results processed by traditional depth enhancement methods. In Table I, PSNR results of JBF, WMF, AD and median filtering are effectively improved when our method is used for large-area depth recovery. Comparing

PSNR results in Table I, using our method for large-area depth recovery can achieve an average of 1.1 dB increase.

BPR results are shown in Table II. For threshold T , we mark pixels as bad pixels when the differences between recovered depth values and ground true depth values are beyond the threshold T . In Table II, compared with JBF, WMF and AD, our method can achieve at least 13.42%, 16.71% and 30.45% performance improvement for $T = 0$, $T = 1$ and $T = 2$ respectively. Besides, for each row in Table II, by comparing BPR results of our method under different threshold T , we also see that bad pixel rates significantly decrease with the increase of threshold T . The results indicate that differences between recovered depth values and ground true depth values

TABLE I
PSNR RESULTS OF DIFFERENT DEPTH ENHANCEMENT METHODS.

	JBF	JBF+Ours	WMF	WMF+Ours	AD	AD+Ours	Median	Ours+Median
Art(view 1)	27.1019	28.6162	26.5532	27.7850	27.2102	28.2351	27.4797	28.7095
Art(view 2)	24.7370	25.6984	24.6143	25.4846	24.8039	25.6859	24.9920	25.8121
Midd(view 1)	24.7662	25.9056	24.6124	25.7789	24.7093	25.9081	24.8397	25.9006
Midd(view 2)	24.8552	26.3526	24.6999	26.1330	24.9178	26.2760	24.9476	26.3396

TABLE II
BPR RESULTS OF DIFFERENT DEPTH ENHANCEMENT METHODS.

	T = 0				T = 1				T = 2			
	JBF	WMF	AD	Ours	JBF	WMF	AD	Ours	JBF	WMF	AD	Ours
Art(view 1)	0.0558	0.1613	0.0558	0.0438	0.0547	0.0632	0.0535	0.0320	0.0534	0.0574	0.0516	0.0268
Art(view 2)	0.0718	0.1723	0.0718	0.0641	0.0699	0.0774	0.0691	0.0553	0.0683	0.0723	0.0671	0.0503
Midd(view 1)	0.0753	0.1658	0.0758	0.0679	0.0737	0.0804	0.0745	0.0630	0.0722	0.0751	0.0734	0.0558
Midd(view 2)	0.0796	0.1719	0.0801	0.0634	0.0779	0.0856	0.0788	0.0596	0.0764	0.0798	0.0776	0.0508

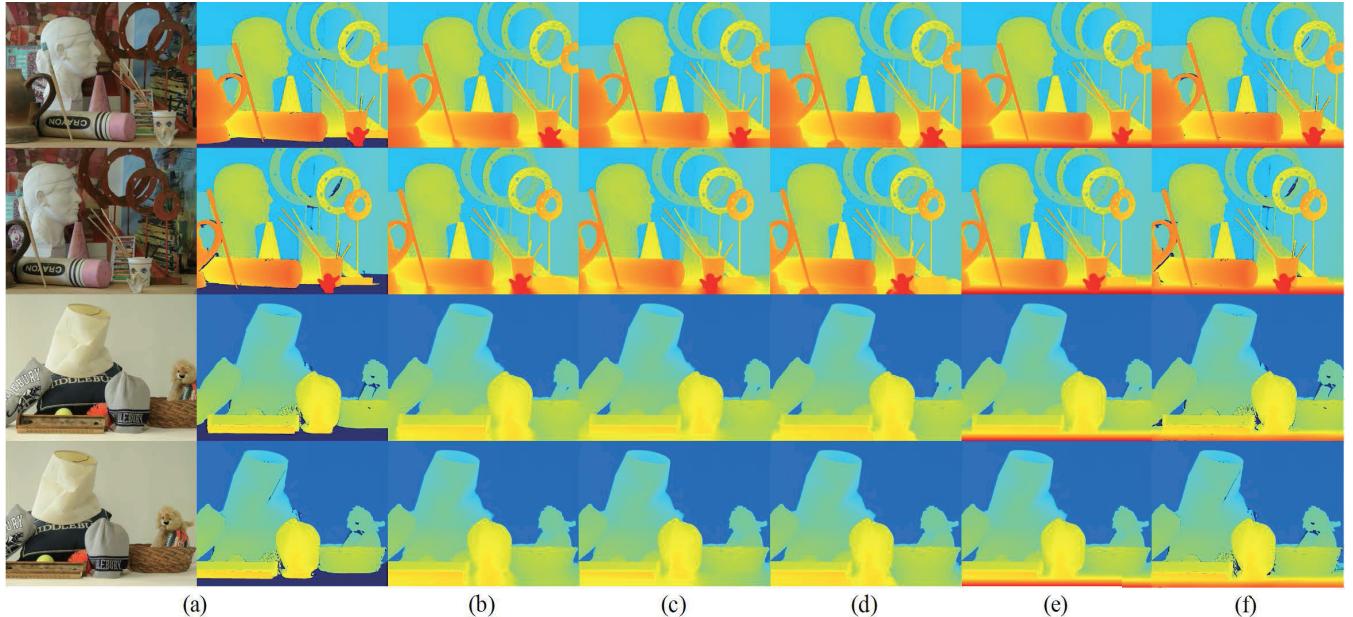


Fig. 8. Visual quality comparison for depth enhancement methods on Middlebury RGB-D dataset: (a) RGB-D pair, depth maps enhanced by (b) JBF [9], (c) WMF [14], (d) AD [13], and (e) our method, (f) Depth ground truth.

are small, and these differences can be effectively narrowed by using a more accurate projection matrix P .

IV. CONCLUSION

In this paper, we have proposed a texture-free depth enhancement method for large-area depth recovery. Given a depth map, we extract an outer contour of large-area depth missing region, and iteratively divided the contour into different contour segments. Each contour segment is projected into the world coordinate system and multiple linear regression method is used for surface function approximation. Then, statistical histograms of approximated surface functions of all contour segments are obtained, and contour pixels correspond to the peaks of histograms are selected as candidates for depth recovery. After that, a candidate refinement method is applied to remove noisy candidates, and the surface function of the region can be obtained by applying the multiple linear regression method to candidates after refinement. Missing depth values are recovered by referring the approximated surface function. In experiments, our method can averagely achieve 1.1 dB increase in PSNR and at least 13.42% performance improvement in BPR.

ACKNOWLEDGMENT

This work was supported in part by NSFC (Grant No. 61231010), NSF (Grant No. 1216001), Research Fund for the Doctoral Program (20120142110015) and 863 High-tech R&D Program (No. 2015AA015903).

REFERENCES

- [1] S. C. Chan, H. Y. Shum, and K. T. Ng, "Image-based rendering and synthesis," *IEEE Signal Proc. Mag.*, 24(6): 22-33, 2007.
- [2] Y. Yuan, J. Fang, and Q. Wang, "Robust superpixel tracking via depth fusion," *IEEE Trans. Circuits Syst. Video Technol.*, 24(1): 15-26, 2014.
- [3] W. Li, J. Zhou, B. Li, and M. I. Sezan, "Virtual view specification and synthesis for free viewpoint television," *IEEE Trans. Circuits Syst. Video Technol.*, 19(4): 533-546, 2009.
- [4] Y. Yao, and Y. Fu, "Contour model based hand-gesture recognition using Kinect sensor," *IEEE Trans. Circuits Syst. Video Technol.*, 24(11): 1935-1944, 2014.
- [5] D. Min, and K. Sohn, "Cost aggregation and occlusion handling with WLS in stereo matching," *IEEE Trans. Image Process.*, 17(8): 1431-1442, 2008.
- [6] M. F. Tappen, and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *Proc. ICCV*, 2003, pp. 900-906.
- [7] S. B. Gokturk, H. Yalcin, and C. Bamji, "Time-of-flight depth sensor-system description, issues and solutions," in *Proc. CVPRW*, 2004.
- [8] Y. Zhang, Z. Xiong, Z. Yang, and F. Wu, "Real-time scalable depth sensing with hybrid structured light illumination," *IEEE Trans. Image Process.*, 23(1): 97-109, 2014.
- [9] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *Proc. ACM SIGGRAPH*, 2007, p. 96.
- [10] L. F. Yu, S. K. Yeung, Y. W. Tai, and S. Lin, "Shading-based shape refinement of RGB-D images," in *Proc. CVPR*, 2013, pp. 1415-1422.
- [11] D. Miao, J. Fu, Y. Lu, S. Li, and C. Chen, "Texture-assisted kinect depth inpainting," in *Proc. ISCAS*, 2012, pp. 604-607.
- [12] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *Proc. CVPR*, 2014, pp. 3390-3397.
- [13] J. Liu, and X. Gong, "Guided depth enhancement via anisotropic diffusion," in *Proc. PCM*, 2013, pp. 408-417.
- [14] D. Min, J. Lu, and Do, M.N., "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Process.*, 21(3): 1176-1190, 2012.
- [15] S. Y. Kim, J. H. Cho, A. Koschan, and M. A. Abidi, "Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor," in *Proc. ICPR*, 2010, pp. 2358-2361.
- [16] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, 19(11): 2861-2873, 2010.
- [17] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model," *IEEE Trans. Image Process.*, 23(8): 3443-3458, 2014.
- [18] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11): 1330-1334, 2000.
- [19] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*, 5th ed. John Wiley & Sons, 2012.