Object Boundary Based Denoising for Depth Images

Mayoore S. Jaiswal[1(✉)], Yu-Ying Wang[2], and Ming-Ting Sun[1]

[1] University of Washington, Seattle, WA 98195, USA
mayoore@uw.edu
[2] National Taiwan University, Taipei, Taiwan

Abstract. Economical RGB-D cameras such as Kinect can produce both RGB and depth (RGB-D) images in real-time. The accuracy of various RGB-D related applications suffers from depth image noise. This paper proposes a solution to the problem by estimating depth edges that correspond to the object boundaries and using them as priors in the hole filling process. This method exhibits quantitative and qualitative improvements over the current state-of-the-art methods.

Keywords: RGB-D camera · Depth image · Hole-filling · Edge detection

1 Introduction

Popular commercial RGB-D cameras can capture both color and depth images at an economical price, and have been used widely in many applications. However, the quality of the depth images is degraded by various holes and noises. A Kinect depth image and its associated color image is shown in Fig. 1. Shadow holes caused by the displacement between the IR projector and IR sensor, generally occurs along object boundaries in depth images as highlighted in red in Fig. 1. Additionally, depth sensors often are unable to accurately estimate the depth values of smooth, shiny, and reflective surfaces because IR rays reflected from these surfaces are weak or scattered. This results in random noises in depth images as highlighted in green in Fig. 1.

Depth hole filling and denoising is challenging because it needs to accurately estimate depth while preserving depth discontinuities around object boundaries. Many approaches for denoising and hole-filling for depth images have been proposed. They fall into one of the following broad categories: (1) filtering based [1, 2, 6–8], (2) plane-fitting based [3, 4], and (3) probabilistic methods [5]. Filtering based approaches use variations of bilateral filters to denoise the depth image. However, due to the false edges from the texture of color images, illumination of the scene, or weak edges due to the similar color of the object and the background, the bilateral filters or the variations of thereof may not produce the best result. Plane fitting based methods fit planes to neighborhood pixels. However, these methods fail when there are object boundaries inside the holes and in complex scenes consisting of multiple objects with complex geometries. For an example of the probabilistic methods, an autoregressive model with color images is proposed in [5]. Though this method produces qualitatively good results in large structures, it blurs smaller object boundaries.

F. Karray et al. (Eds.): ICIAR 2017, LNCS 10317, pp. 125–133, 2017.
DOI: 10.1007/978-3-319-59876-5_15

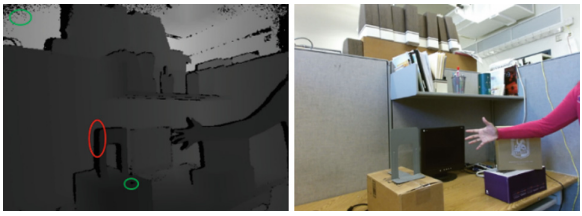

Fig. 1. A depth image and its corresponding color image obtained from a Kinect camera. (Color figure online)

The main contribution of this paper is that we propose a new approach which focuses on constructing edges which correspond to the true object boundaries and using them to guide the hole filling process. We propose a novel depth edge estimation algorithm which estimates depth edges using information from the color edges and noisy depth edges by observing the statistics of the neighborhood depth pixel values. We use adaptive thresholds in detecting color edges to overcome the problem when objects have similar color as background. The refined depth edge image can be used in other applications, such as depth denoising and object segmentation. In the hole-filling process, we use a trilateral filter with an adaptive window size to prevent pixels from the opposite side of an object boundary being used as reference pixels. Experimental results show the effectiveness of our proposed algorithm.

The rest of this paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 shows the quantitative and qualitative effectiveness of the proposed method using various datasets. Finally, Sect. 4 lists the conclusions.

2 Proposed Method

Our proposed approach is shown in Fig. 2 and described in the following subsections.

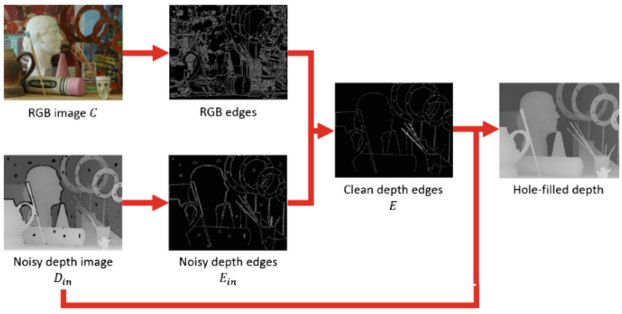

Fig. 2. Illustration of the proposed method.

2.1 Color Edge and Noisy Depth Edge Extraction

The objective of the color edge extraction module is to be able to extract all possible edges even when the foreground and background have similar colors. The edges from the color

image were extracted using Canny edge detector [9]. To reduce the localization error, we use a small standard deviation of 0.1 for the Gaussian filter. Also, in order to detect all possible edges, we use 0.01 and 0.1 for the lower and upper hysteresis thresholds respectively. These choices could yield many edges caused by texture and illumination changes in the scene which will be rejected later. Many previous algorithms have problems when the foreground object has a similar color as the background, since the color edges will be faint and may not be detected by the edge detector. To overcome this problem, we use a threshold value proportional to the variance of the pixel color values inside the hole region. With this adaptive thresholding strategy, the faint color edges can be detected inside the hole regions, without generating many false edges in other regions.

Small random perturbations in the noisy depth image are filtered out using a 5×5 median filter before the Canny edge detector is applied to the depth image.

2.2 Generating Clean Depth Edges

The noisy depth edge image E_{in} obtained from the previous step has false edges caused by random noise and holes, and true edges produced by object boundaries. The clean depth edges are produced in two stages. First, edges coinciding with object boundaries are identified and retained. Other false edges are suppressed. Second, object boundaries in the hole regions of the raw depth image are extracted from the color edges.

To identify edges associated with boundary edges, it is observed that if there are less hole-pixels in the input depth image D_{in} in the neighborhood of an edge pixel location $E_{in}(p)$, this edge pixel is likely caused by the presence of an object boundary. Otherwise it is probably caused by erroneous depth values in the input depth image. Let I be a binary mask and I_q is the value of the mask at pixel q. $I_q = 1$ if q is a hole-pixel in the input depth image D_{in}, otherwise $I_q = 0$. Equation (1) is used to find edges caused by the presence of object boundaries.

$$E_1(p) = \begin{cases} 1, & if \sum_{q \in N_m(p)} G_q \cdot I_q < th, and\, E_{in}(p) = 1; \\ 0, & otherwise. \end{cases} \tag{1}$$

where $E_{in}(p) = 1$ means p is a pixel on an edge in the noisy depth edge image E_{in}, th is a threshold, $N_m(p)$ are pixels in an $m \times m$ local neighborhood centered at p. G_q is the weight at location q with an $m \times m$ Gaussian kernel G centered at p. $E_1(p) = 1$ indicates that p is on an edge caused by an object boundary. Gaussian kernel G with σ_m is used to spatially weigh the depth pixel locations within the $m \times m$ local neighborhood. By using a Gaussian kernel, a distant hole pixel from p would weigh less than a hole pixel close to p within the neighborhood.

Next, false edges produced by random noise and holes in the depth image are removed. The variation of depth pixel values in the locale of false edges caused by random noise in the depth image is small; whereas in the neighborhood of false edges caused by holes in the depth image, the variation of depth pixel values is generally higher when there is an object boundary within the hole. The variation of depth values is evaluated by fitting a plane to an $h \times h$ local neighborhood of a false edge pixel in the raw depth image and computing the distances from each depth pixel in this local

neighborhood to the fitted plane. If the variance of the distances is larger than a threshold, the false edge pixel was likely caused by a hole with an object boundary. These false edges are replaced by edges in the color image. The window size, h is adaptively chosen such that at least 80% of the pixels in the $h \times h$ neighborhood are non-hole depth pixels. The adaptive window size ensures that sufficient number of non-hole depth pixels are taken into account. The initial window size of h is set to 21.

To find object boundaries in the hole regions of the raw depth image D_{in}, we elicit the edges from the color image. The depth hole regions with likely object boundaries are dilated with a 3×3 structuring element with all the elements equal to 1 and the origin at the center. The intersection of the dilated region and the color edge image is the depth edge image corresponding to object boundaries. The clean depth edge image E is the combination of this depth edge image and $E_1(p)$ from Eq. (1).

2.3 Depth Hole Filling

Once the clean depth edges are recovered, we use a trilateral filter as described in Algorithm 1 to construct the recovered depth image D_{rec}. An edge and color aware adaptive trilateral filter in Eq. (2) is used to fill the hole-pixels progressively from the perimeter of the hole to its center. The non-hole pixels outside the perimeter of the hole is used to estimate the hole depth pixels. The size of the local neighborhood $n \times n$ used to fill a depth hole-pixel p varies adaptively to ensure that at least 80% of pixels in the neighborhood are non-hole pixels with initial $n = 7$. With this approach, more non-hole pixels are used in the trilateral filter than using a fixed neighborhood size.

Algorithm 1: Depth hole-filling with edge and color aware adaptive trilateral filter

Input: Recovered depth edge image $\boldsymbol{E}$, input raw depth image $\boldsymbol{D_{in}}$
Output: Recovered depth $\boldsymbol{D_{rec}}$
Initialization: Push every hole-pixel in input depth image $\boldsymbol{D_{in}(p)}$ into a priority queue $\boldsymbol{Q}$ with decreasing ordered by the percentage of non-hole pixels in a $\boldsymbol{n \times n}$ window centered at $\boldsymbol{p}$, initial $\boldsymbol{n = 7}$
While $\boldsymbol{Q}$ is not empty: $\boldsymbol{D_{in}(p)} \leftarrow$ pop $\boldsymbol{Q}$
  Update $\boldsymbol{D_{in}(p)}$**:**
    $\boldsymbol{num} \leftarrow$ percentage of non-hole pixels in $\boldsymbol{n \times n}$ window centered at $\boldsymbol{D_{rec}(p)}$
    If $\boldsymbol{num < 80\%}$: $\boldsymbol{n \leftarrow n + 2}$
      Push $\boldsymbol{D_{in}(p)}$ into $\boldsymbol{Q}$
    Else: Compute $\boldsymbol{D_{rec}(p)}$ using Equation (2)
    End
End
Return $\boldsymbol{D_{rec}}$

The depth hole-pixels $D_{rec}(p)$ can be found as:

$$D_{rec}(p) = \frac{1}{k_p} \sum\nolimits_{q \in N_n(p)} D_{in}(q) \cdot f_s(\|q - p\|) \cdot f_r(E, q, p) \cdot f_i(C, q, p) \tag{2}$$

where $f_s(\cdot)$ is a zero mean spatial Gaussian kernel with a standard deviation σ_d. k_p is a normalizing factor. $\|\cdot\|$ is the Euclidean distance measure. The range kernel $f_r(\cdot)$ is a binary indicator defined as:

$$f_r(E,q,p) = \begin{cases} 1, \textit{if the shortest path connecting pixels } q \textit{ and } p \\ (\textit{excluding pixels } q \textit{ and } p) \\ \textit{does not intersect an edge in } E; \\ 0, \textit{ otherwise}. \end{cases} \tag{3}$$

The range kernel f_r assures that only pixels on the same side of an edge are averaged. The intensity kernel $f_i(\cdot)$ is a binary indicator defined as:

$$f_i(C,q,p) = \begin{cases} 1, & \textit{if } \|C(q) - C(p)\| \\ 0, & \textit{otherwise}. \end{cases} \tag{4}$$

where C is the color image with the R, G and B components normalized to $[0,1]$ and ε is a threshold. The intensity kernel $f_i(\cdot)$ ensures that neighborhood depth pixels with similar color in the corresponding color image are used to fill a hole depth pixel. When p is an edge pixel, it is ambiguous which side of the edge does p belong to. In this case, a surrogate pixel p' which is a pixel in the 8 connected neighborhood of p that is not an edge pixel and has the closest color to p in the corresponding color image is found using Eq. (5). The surrogate pixel p' is used to determine which side of an edge, pixel p belongs to.

$$p' = \min_{u \in N'(p)} \|C(p) - C(u)\| \tag{5}$$

where u is a pixel in the neighbourhood $N'(p)$ such that it is one of the 8 connected pixels of p but is not an edge pixel. $C(p)$ is the RGB color value of pixel p.

3 Experimental Results

In this section, we present results from evaluating our method with respect to other state-of-the-art methods using Mean Absolute Error (MAE) as in Eq. (6).

$$MAE = \frac{1}{N} \sum\nolimits_{p \in \left\{P | D_{syn}=0 \cap D_{gt} \neq 0\right\}} \left| D_{rec}(p) - D_{gt}(p) \right| \tag{6}$$

where D_{rec}, D_{gt} and D_{syn} are the recovered depth, ground truth depth and synthetically degraded depth, respectively. N is the number of hole-pixels in the synthetically degraded depth that are not hole-pixels in the ground truth depth. MAE is calculated only in the synthetically degraded hole-pixels in D_{syn}, which means $D_{syn}(p) = 0$ but the ground truth depth image D_{gt} has valid depth pixel values. This enables us to quantitatively evaluate the algorithms' ability to recover synthetic hole-pixels in object boundary and smooth regions. The parameters settings are:$m = 9$, $\sigma_m = 1$, $th = 0.055$, $t = 0.09$, $\sigma_d = 1$, and ε is 0.1.

Yang et al. introduced a synthetically degraded dataset in [5] which consists of a subset of the Middlebury dataset [10] with random noise and structural holes created on the depth images to imitate the holes in depth images produced by a Kinect. Figure 3 shows a sample of the images in the Yang dataset. To mimic Kinect depth images, structural missing pixels are created along depth discontinuities, and random missing pixels are generated in flat areas.

The proposed method is compared to bicubic interpolation and four state-of-the-art methods JBF [11], Guide [1], CLMF [2] and AAR [5] to produce recovered depth images. The structure similarity index (SSIM) is calculated as described in [12]. Table 1 tabulates the results. The proposed method gets the least *MAE* score on all the images. On average the method reduces the MAE error by 14.02%. Our method also gets better or comparable SSIM values on the dataset.

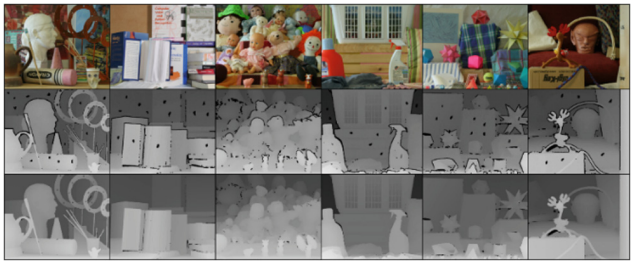

Fig. 3. Thumbnails from the synthetically degraded dataset [5]. The upper row shows the color images. Middle row shows the corresponding depth images degraded by random noise and structural holes. The bottom row shows the ground truth depth images. (Color figure online)

Table 1. Table lists the quantitative results comparing our method to the state-of-the-art methods tested on the synthetically degraded dataset [5].

Algorithm	Art		Book		Dolls		Laundry		Moebius		Reindeer	
	MAE	SSIM	MAE	SSIM	MAE	SSIM	MAE	SSIM	MAE	SSIM	MAE	SSIM
AAR	4.78	0.987	1.99	0.991	1.91	0.988	3.14	0.982	2.40	0.989	2.12	0.990
Bicubic	7.45	0.984	2.30	0.991	2.12	0.986	3.81	0.980	2.65	0.988	3.05	0.987
CLMF	6.40	0.988	2.07	0.992	1.92	0.988	3.51	0.982	2.38	0.990	2.80	0.988
Guide	6.94	0.985	2.10	0.992	1.94	**0.989**	3.42	**0.985**	2.42	0.990	2.81	0.989
JBF	6.97	0.983	2.36	0.989	2.13	0.985	3.70	0.979	2.76	0.986	2.95	0.984
Ours	**4.67**	**0.988**	**1.61**	**0.993**	**1.79**	0.988	**2.50**	0.984	**1.89**	**0.991**	**1.81**	**0.991**

To test the proposed method's ability to reconstruct depth values in hole regions, we repeated the above experiment by calculating the mean absolute error using:

$$MAE_{ob} = \frac{1}{N}\sum\nolimits_{p\in\{P|D_{syn}=0\cap D_{gt}\neq 0\cap J'=1\}} \left|D_{rec}(p) - D_{gt}(p)\right| \tag{7}$$

where J is a mask indicating the depth hole regions which have object boundaries. N is the number of hole-pixels in the synthetically degraded depth that are not hole-pixels in the ground truth but in J. MAE_{ob} is calculated only in the synthetically degraded hole-pixels in D_{syn} which have object boundaries inside the hole region, which means $D_{syn}(p) = 0$ and $M(p) = 1$ but the ground truth depth image D_{gt} has valid depth pixel values. Table 2 tabulates the results. The proposed method gets the best results on all the images. On average the method reduces the MAE error by 36.78%.

Table 2. Table lists the MAE_{ob} results comparing our method to the state-of-the-art methods tested on the synthetically degraded dataset [5].

Algorithm	Art	Book	Dolls	Laundry	Moebius	Reindeer
AAR	10.50	5.93	4.62	8.06	5.55	6.17
Bicubic	17.45	7.71	5.67	10.65	6.55	10.80
CLMF	14.57	6.84	5.08	9.85	5.80	9.93
Guide	16.08	6.96	5.13	9.53	5.93	9.91
JBF	16.35	8.00	5.69	10.15	6.92	10.17
Ours	**10.31**	**3.80**	**3.87**	**5.58**	**3.54**	**4.34**

Figure 4 further illustrates the effectiveness of the proposed method in preserving depth edges. They show cropped and zoomed portions of the 'Book' images in the synthetically degraded dataset. Bicubic interpolation blurs the depth edge. Our method produces the least distorted edges. Guided and CLMF blur the object boundaries similar to bicubic. JBF and AAR produce jagged artefacts near the edges. Our method is able to fill holes while producing clean edges at object boundaries. Quantitative and qualitative experiments prove the strength of the proposed method.

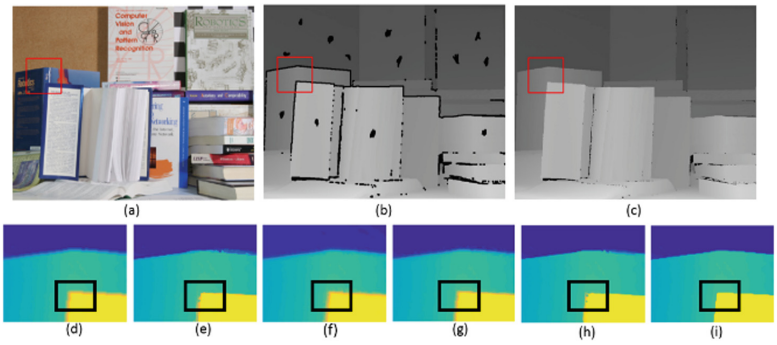

Fig. 4. Visual comparison results on the 'Book' image. (a), (b) and (c) are the color, synthetically degraded depth and the ground-truth depth respectively. (d) – (i) show zoomed in view in the red rectangle region. (d) bicubic (e) JBF (f) Guide (g) CLMF (h) AAR and (i) ours. Note that our method preserves the edge structures the best, there are no "orange" colored pixels in the colored recovered depth image as illustrated in the boxes highlighted in black. (Color figure online)

Next we test the proposed method using color and depth data from a Kinect camera. For this purpose, we use the Kinect data provided by [5]. The depth image was denoised using AAR, Guide, JBF and our method. Due to the lack of ground truth, we present only a qualitative comparison of the results. Figure 5 compares the AAR, Guide, JBF, and our method. The cropped and zoomed images show that the proposed method preserves the depth edges better than other methods. JBF, AAR and Guide blur and distort the object boundaries in the depth image as highlighted in Fig. 5.

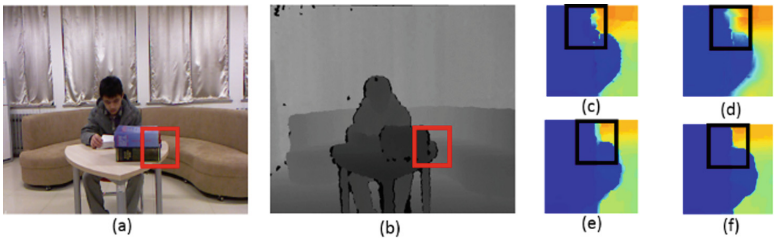

Fig. 5. A visual comparison of results from the state-of-the-art methods on a set of Kinect images. The thumbnails are zoomed in portions of the area highlighted in red the input Kinect images. (a) color image from Kinect, (b) depth from Kinect, (c) AAR, (d) Guide, (e) JBF, and (f) Our method. (Color figure online)

4 Conclusion

In this paper, we propose a new approach to denoise and fill the holes in depth images. The proposed algorithm first extracts an initial depth edge image from the raw depth image which is noisy due to the inaccuracies present in the raw depth image. This initial edge image is refined by using the edges from the color images. The clean depth edge image is used to fill the holes in the depth image by using an edge and color aware trilateral filter. Many parameters are made adaptive to make the algorithm robust. Quantitative and qualitative experimental results demonstrate that the proposed hole filling strategy can generate more accurate depth images than existing methods.

References

1. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35** (6), 1397–1409 (2013)
2. Lu, J., Shi, K., Min, D., Lin, L., Do, M.N.: Cross-based local multipoint filtering. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
3. Xu, L., Au, O.C., Sun, W., Li, Y., Li, J.: Hybrid plane fitting for depth estimation. In: Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) (2012)
4. Matsumoto, K., De Sorbier, F., Saito, H.: Plane fitting and depth variance based upsampling for noisy depth map from 3D-ToF cameras in real-time. In: SciTePress (2015)

5. Yang, J., Ye, X., Li, K., Hou, C.: Depth recovery using an adaptive color-guided auto-regressive model. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 158–171. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33715-4_12
6. Camplani, M., Salgado, L.: Efficient spatio-temporal hole filling strategy for Kinect depth maps. In: International Society for Optics and Photonics, IS&T SPIE Electronic Imaging (2012)
7. Liu, J., Gong, X., Liu, J.: Guided inpainting and filtering for Kinect depth maps. In: 2012 21st International Conference on Pattern Recognition (ICPR) (2012)
8. Wang, Z., Hu, J., Wang, S., Lu, T.: Trilateral constrained sparse representation for Kinect depth hole filling. Pattern Recogn. Lett. **65**, 95–102 (2015)
9. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
10. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Computer Vision and Pattern Recognition (2007)
11. Riemens, O., Gangwal, O., Barenbrug, B., Berretty, R.-P.: Multistep joint bilateral depth upsampling. In: International Society for Optics and Photonics, IS&T SPIE Electronic Imaging (2009)
12. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)