An Iterative, Non-local Approach for Restoring Depth Maps in RGB-D Images

Akash Bapat
Computer Science
UNC Chapel Hill
Email: akash@cs.unc.edu

Adit Ravi
Electrical Engineering
IIT Madras
Email: adit.ravi@gmail.com

Shanmuganathan Raman
Electrical Engineering
IIT Gandhinagar
Email: shanmuga@iitgn.ac.in

***Abstract*—In this paper, we present a novel iterative median filter based strategy to improve the quality of the depth maps provided by sensors like Microsoft Kinect. The quality of the depth map is improved in two aspects, by filling holes present in the maps and by addressing the random noise. The holes are filled by iteratively applying a median based filter which takes into account the RGB components as well. The color similarity is measured by finding the absolute difference of the neighbourhood pixels and the median value. The hole filled depth map is further improved by applying a bilateral filter and processing the detail layer separately using Non-Local Denoising. The denoised detail layer is combined with the base layer to obtain a sharp and accurate depth map. We show that the proposed approach is able to generate high quality depth maps which can be quite useful in improving the performance of various applications of Microsoft Kinect such as pose estimation, gesture recognition, skeletal and facial tracking, etc.**

I. Introduction

Microsoft Kinect was developed to accompany the Xbox 360 video game console as a sensor for motion and gesture recognition [1]. This device employs natural user interface using gestures and audio [2]. Kinect is based on a web-cam style peripheral, which has a color camera and a depth sensor giving a decent resolution of 480×640. The depth sensor consists of an infra-red (IR) laser projector and a monochrome sensor. Although developed for indoor gaming purposes, Kinect is finding applications in various challenging problems of 3D computer vision. With an increasing number of Kinect enthusiasts, new and innovative applications are being released everyday.

Kinect senses the depth of a scene by projecting a structured pattern of infra-red light and sensing it with a monochromatic CMOS sensor. The main problem with Kinect is the erroneous noise-ridden depth data. The Kinect depth data has regions where no depth is measured, which are also known as holes. The holes and noise in the depth data arise due to various reasons. The primary ones are listed below.

1) Spatial separation between IR camera and IR projector which leads to object occlusion,
2) Temporal inconsistencies, and
3) Corruption at object edges.

This paper aims to develop a method to improve the depth map, addressing all the three aspects of inaccuracies associated with the depth data. Firstly, the color data is used as a guide to fill in the holes present in the depth map due to the reasons listed above. This approach is based on a median filter to mitigate the spatial inconsistencies in the depth map. The key idea here is that similar looking pixels have similar depths i.e. neighboring pixels with similar RGB values tend to have similar depths. The proposed approach then focuses on denoising the depth data. We employ a bilateral filter to preserve strong edges in the depth map [3]. The depth map is decomposed into base and detail layers using the bilateral filter. A non-local means denoising algorithm is applied to the detail layer to further enhance the accuracy at the edges [4]. Finally, the base layer and the modified detail layer are combined to obtain the desired more accurate depth map. Image inpainting allows one to fill missing data from the surrounding regions [5]. The proposed method is compared qualitatively with the results obtained by applying inpainting algorithm of Telea [6] instead of the iterative median filter based hole filling algorithm step in the proposed approach.

Here are the major contributions of this work.

1) The primary contribution of this paper is a novel way of using the median filter along with RGB data for depth restoration. Till now, median filters were combined with a guiding image using traditional ways like weighted median filter techniques or by using a truncated neighbourhood. This paper takes a different approach and modifies the median filter suitably to incorporate information in RGB data while not using computationally expensive techniques like weighted median filter.
2) Another contribution of this paper is the approach to conduct hole filling using a single depth image rather than using motion vectors or confidence maps which require a series of depth images. This approach gives a unified method without treating the foreground and background separately while filling the holes.
3) We also propose to use a combined non-local means and edge preserving filter based denoising scheme to achieve better, more accurate depth maps.

The rest of the paper is organized as follows. Section II discusses some relevant works on Kinect depth data restoration techniques. Then we discuss the effect of spatial distance between IR camera and the IR projector in section III-A. In section III-B, we discuss the inconsistencies observed in the depth map. Section IV-A introduces the median filter based hole filling strategy. This is the crucial step of the proposed algorithm. In section IV-B, we discuss some post-processing steps using bilateral and non-local means filters to

generate more accurate depth maps. In section V, we present a comparison between results obtained by the proposed approach and Telea's inpainting algorithm. We conclude the paper with pointers for further improvement of the proposed method.

II. Related Work

The use of an iterative diffusion based method that accounts for both the known depth values and RGB-D segmentation results to recover missing depth information was proposed in [7]. This method used graphical processing units (GPUs) to take advantage of parallelism in the algorithm. In [8], Camplani and Salgado explored the sources of noise and found that the Kinect depth values change drastically even for a static object. This work used an iterative bilateral filter framework and a confidence map to get the depth values. Their approach although novel, is limited by the fact that the scene was assumed to be static. Also computationally expensive joint bilateral filter is iteratively employed in this paper, which results in sub-par performance when one uses this method for real time applications. When motion in the scene is also considered, this method may fail.

In [9], Shen *et al.* assume that the scene can be decomposed into static background and a dynamic foreground comprising of multiple objects. They do an initial training using scenes having only the background. The layers are then labelled using a probabilistic model. Maximum a posteriori (MAP) estimation is used for labelling to preserve edges.

In [10], Berdnikov and Vatolin identified two different causes of holes in depth data and separated these causes automatically in their scheme. They developed suitable hole filling algorithms for each case and came up with an adaptive algorithm. Although this scheme is better than a naive approach to use a simple Gaussian filter, it fails to incorporate edge information provided by the color image. Alternately, a temporal filtering can be employed on Kinect RGB-D video data to achieve stable hole filling [11].

In [12], Milani and Calvagno proposed to split the depth data into different clusters and different segments are restored by correction followed by interpolation. This method is more suitable for layered scenes and is not suitable for any general real world scene. The segmentation into different depth layers is very difficult in the case of most natural scenes.

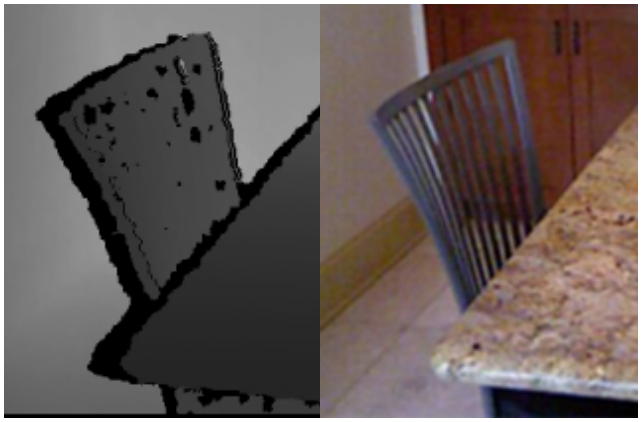
In [13], Schmeing and Jiang addressed the jaggedness of edges in the depth map. They used superpixels to identify correct edge information in the RGB image and this information is used as a guide to improve the edges in the depth map. In [14], Qi *et al.* use an inpainting algorithm which integrates non-local filtering. However, a texture-less depth map may prove to be a problem where termination boundary for the inpainting algorithm is difficult to estimate.

There are works which combine a high resolution RGB image with a low resolution depth map to produce a high resolution depth map using markov random fields (MRF) [15]. Similar MRF approach has been used by defining prior on RGB and depth data using Field-of-Experts framework based on natural image statistics to achieve depth inpainting and upsampling [16].

III. Motivation: Origin of Holes

A. Occlusion

The Kinect device has an IR camera and an IR projector which are spatially separated. Due to this spatial separation, the projector and the sensor have slightly different views of the same object. Hence, there are regions which the IR light cannot reach due to occlusion but the RGB camera can see. This results in contiguous chunks of pixels where no depth measurements are available (see Fig. 1(a)). These holes are the hardest to fill as no depth information is available for large groups of pixels. These regions can be filled by extending the depth of similar looking neighbouring pixels whose depth is known.



(a) (b)

Fig. 1: Holes due to occlusion

B. Spatial and temporal inconsistencies

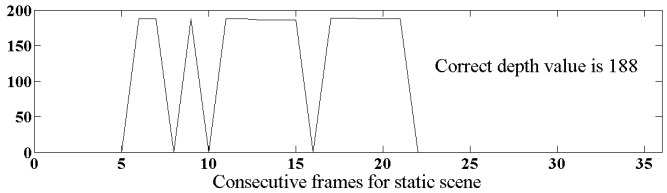


Fig. 2: Variation of depth value for a static scene at one pixel location; Correct depth value is 188.

Kinect gives noisy depth maps where neighbouring pixels are not coherent. Even though two objects are at the same depth, Kinect depth data may not be the same for them. This problem is aggravated by Kinect's temporal inconsistency. As Camplani *et al.* identified in [8], for a static scene at the same point, there is a wide variation of depth detected as time evolves. These fluctuations seriously impede us from taking the time variation as additional information to restore the depth map (see Fig. 2). In the case of a pixel present inside a hole, the depth is not detected for as long as 20 consecutive frames. These problems encourage us to use the median of the neighbourhood of a pixel to restore depth rather than an iterative joint bilateral filter or a filter which uses temporal information or a combination of both. Kinect depth maps have

highly irregular and jagged profiles at object edges. Depths of specular surfaces and black coloured objects in the scene are also difficult to detect.

IV. Proposed Approach

The general outline of the proposed approach is shown in Fig. 3. The iterative median filter based hole filling step is described in section IV-A. After the holes in the depth image are filled by an iterative algorithm, the depth image is split into a base layer and a detail layer using a bilateral filter. The detail layer contains most of the noise and finer depth variations. Hence, it is processed using a non-local means filter to improve the accuracy at the edges. It is then combined with the base layer to get the final improved depth map. This process is discussed in detail in section IV-B.

A. Iterative Median filter based hole filling

The median filter is resistant to shot noise and is edge preserving to some extent. Hence, it is a good choice to fill the holes described in section III. Though it works well in the case of spatial inconsistencies while still preserving edges, it fails to fill holes as no depth value is available for large regions around the holes. Hence it needs to be modified suitably to fill the holes.

In our approach, we consider the RGB image as a guide for detecting object boundaries and to fill the holes. For each pixel location in the RGB image, we find out the region in its neighbourhood which is similar to its median value and then use this information to fill the holes.

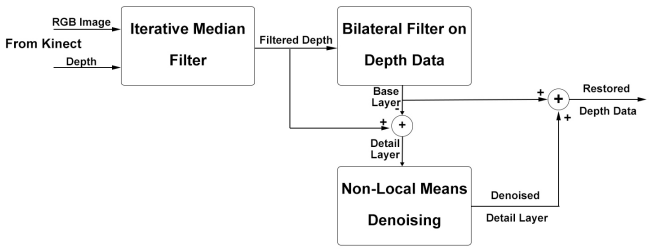


Fig. 3: The Proposed Approach

If there are no holes in the depth map provided by Kinect, the median filter is ideal to mitigate the noise due to temporal inconsistencies. Hence we would like this algorithm to behave as a simple median filter in a case where there are no holes. We shall discuss the proposed algorithm in detail now for a gray-scale image. In the case of an RGB image, the same process can be extended using the data in all three channels.

Consider a single channel of an RGB image $I(x, y)$, where x and y denote pixel co-ordinates in an image grid Ω. Let $N_I(x, y)$ be the neighbourhood for the pixel location (x, y). We use a neighbourhood of 3×3 in this work. Hence $N_I(x, y)$ is a 9-dimensional neighbourhood intensity vector corresponding to each pixel location (x, y). In the following equations, let MED denote the median of an array of intensity or depth values.

$$M_I(x, y) = MED(N_I(x, y)) \tag{1}$$

We subtract the median $M_I(x, y)$ from each pixel in the neighbourhood of a pixel at (x, y) . To get a measure of the similarity, we threshold it with a fixed gray scale threshold C_{th}. This results in another 9-dimensional neighbourhood vector $N_s(x, y)$ which indicates the gray scale similarity.

$$N'_I(x, y) = |N_I(x, y) - M_I(x, y)| \tag{2}$$

$$N_s(x, y) = \begin{cases} 1, & \text{if } N'_I(x, y) < C_{th} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The similarity neighbourhood vector $N_s(x, y)$ is used as a guide for depth data restoration. Let $N_D(x, y)$ denote a neighbourhood in the depth map $D(x, y)$ constructed for a pixel (x, y). For a 3×3 neighbourhood, $N_D(x, y)$ is a 9-dimensional vector. Now, for every pixel (x, y), we pre-calculate $M_D(x, y)$.

$$M_D(x, y) = MED(N_D(x, y)) \tag{4}$$

This removes any shot noise in the depth image and $M_D(x, y)$ serves as a reference for depth values. Once $M_D(x, y)$ has been calculated for every pixel location (x, y), we proceed to the next step. For a given pixel location (x, y), we perform the following operations.

1) We calculate $M_d(x, y)$ as below
$$M_d(x, y) = MED(N_D(x, y)) \tag{5}$$
2) For every pixel location (p, q) in the neighbourhood of (x, y). $M_D(p, q)$ is compared with $M_d(x, y)$ at only those points which were marked similar by the $N_s(x, y)$.
3) If the difference $|M_D(p, q) - M_d(x, y)|$ is smaller than the threshold D_{th}, the pixel $D(p, q)$ is assigned the value $M_D(p, q)$, else $D(p, q)$ is assigned the value $M_d(x, y)$.

Note that $M_d(x, y)$ is different from $M_D(p, q)$ and $M_D(x, y)$. $M_D(p, q)$ and $M_D(x, y)$ are both pre-calculated while $M_d(x, y)$ changes as we update $D(p, q)$. $M_d(x, y)$ is therefore the propogating agent for the depth data into the holes. As the iteration count increases, C_{th} is progressively increased by a small value δ to accommodate the higher variation generally seen in specular surfaces. The hole threshold H_{th} acts as the stopping criterion for the iterative algorithm. This is justified due to the decrease in the percentage of holes with the number of iterations for a depth map. This rate of decrease vary widely for different depth maps depending on the scene. The complete process is explained in Algorithm 1.

B. Bilateral and Non-local means filtering

We propose two types of smoothing to increase the depth map accuracy. First, we smooth planar regions to make the depth variation continuous. Second, we smooth the depth map along the direction of the edge. A bilateral filter blurs less textured regions while edges are kept intact [3]. This property of a bilateral filter is particularly helpful at regions where a hole is inaccurately filled or in the case of inaccurate depth propagation at the edges. We can speed up the bilateral filter

Algorithm 1 Iterative Median Filter Based Hole Filling

Require: $C_{th}, D_{th}, H_{th}, \delta, I, D$
for all $(x, y) \in \Omega$ **do**
  $M_D(x, y) \leftarrow MED(N_D(x, y))$
end for
while $\#holes \geq H_{th}$ **do**
  for all $(x, y) \in \Omega$ **do**
    $M_d(x, y) \leftarrow MED(N_D(x, y)$
    $M_I(x, y) \leftarrow MED(N_I(x, y)),$
    for all $(p, q) \in Neighbourhood(x, y)$ **do**
      if $|M_I(x, y) - I(p, q)| \leq C_{th}$ **then**
        $N_s(p, q) \leftarrow 1$
      else
        $N_s(p, q) \leftarrow 0$
      end if
    end for
    ▷ This forms the similarity vector Ns.
    for all $(p, q) \in Neighbourhood(x, y)$ **do**
      ▷ Now we check similarity flags from N_s.
      if $\big((N_s(p, q) == 1)\&\&(D(p, q) == 0)\big)$ **then**
        if $|M_D(p, q) - M_d(x, y)| \leq D_{th}$ **then**
          $D(p, q) \leftarrow M_D(p, q)$
        else
          $D(p, q) \leftarrow M_d(x, y)$
        end if
      end if
    end for
  end for
  $C_{th} \leftarrow C_{th} + \delta$
end while

computation by using a faster approximate implementation such as the ones proposed by Paris and Durand [17], by Porikli [18] or by Yang *et al.* [19].

In many cases, the depth edges are not jagged. Some part of the edge is detected correctly while the rest may have holes. The approximate nearest neighbour patches are found using a patch match algorithm proposed in [20]. A non-local means filter is then employed to operate on similar looking patches and average over the matched patches [4]. This increases the accuracy of depth value filled in the holes.

After the bilateral filtering step, the depth image is split into a base layer and a detail layer. The base layer, which consists of the bilateral filtered data is denoted by D_B.The detail layer is denoted by D_{Det} and the median filtered depth data obtained using Algorithm 1 is denoted by D_{new}.

$$D_{Det} = D_{new} - D_B \tag{6}$$

The detail layer contains more noise and the texture information than the base layer. Since depth in the real world does not vary drastically in a given neighbourhood, these variations are considered noise. Hence, we want to attenuate the subtle variations present in the detail layer. This layer is denoised using a non-local means filter and then combined with the base layer to obtain a more accurate depth map without holes and noise. The equations below explain the process described above.

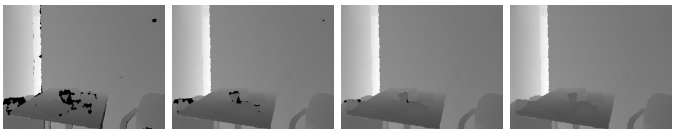
$$D_{NLM} = NLM(D_{Det}) \tag{7}$$

$$D_{final} = D_{NLM} + D_B \tag{8}$$

D_{final} is the final depth map which is obtained after restoration using the proposed approach.

V. Results

To find the effectiveness of this method, the hole filling step as depicted in Fig. 3 is replaced by Telea's inpainting algorithm. For this purpose, we have used the dataset provided by Silberman *et al.* [22]. The dataset is already aligned having a one-to-one correspondence between the depth map and color channels. Grayscale versions of the color channels were used in this study. One can use the entire color channel information by defining a euclidean distance metric in one's preferred color space.



(a) (b) (c) (d)

Fig. 4: (a) Original depth image, (b) Iteration = 3, (c) Iteration = 6, (d) Iteration = 9

We can see the holes progressively getting filled in Fig. 4 as the number of iterations increase. Fig. 4(a) shows the original depth map. Fig. 4(b) shows the result of the iterative median filter algorithm shown in Algorithm 1 after 3 iterations. We can observe that some of the holes are getting filled. Fig. 4(c,d) show the results of Algorithm 1 after 6 and 9 iterations respectively. After 9 iterations, we can observe that almost all of the holes have been filled (see Fig. 4(d)).

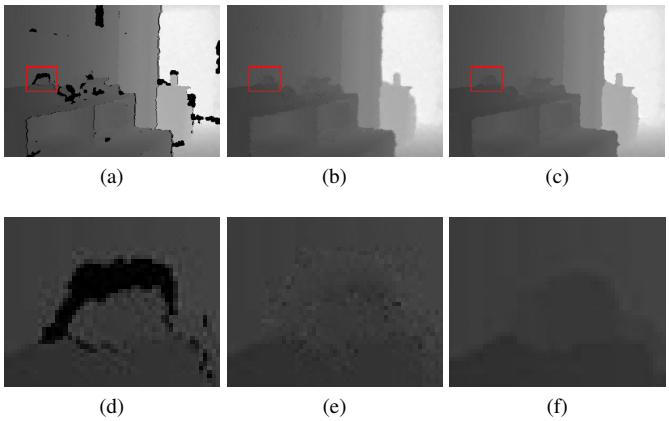
The importance of δ is illustrated by Fig. 5. In the image shown in Fig. 5 the scene contains a highly reflective surface at approximately the same depth. However, the intensity is varying, which is a violation of our assumption that similar looking objects should have similar depth. A non-zero δ ensures that even such areas are filled after a certain number of iterations (see Fig. 5(c)). However when $\delta = 0$, some parts of the hole remains even after the same number of iterations (see Fig. 5(d)). The value of δ should be selected properly in order to make sure that the accuracy is not compromised for non-specular surfaces.

Fig. 6 shows the comparison between the hole-filling capabilities of the proposed approach and the inpainting approach [6]. The difference in the hole-filling capability is evident from Fig. 6 (a-c). Telea's inpainting algorithm depends on the information around the holes. This works well when the holes are in a homogeneous region [6]. As most of the holes are present at the edges in the Kinect depth data, inpainting approach is not quite suitable. The outlined areas in Fig. 6 (a-c) are zoomed into and presented in 6 (d-f). The images in Fig. 6(d-f) illustrate how the proposed method is much better than an inpainting approach. The level of noise is drastically reduced when using the proposed approach. The speckle area introduced by a weighted sum approach is avoided by the median filter based approach (Algorithm 1). Also, the proposed approach sticks to the object edges resulting in clear object boundaries.

Fig. 7 shows the improvement in the depth data at each step of the proposed approach. Fig. 7 (a, b) are the RGB-D images provided by Kinect and Fig. 7(c) is the hole filled image using Algorithm 1. Figs. 7 (d, e) are the binary images to illustrate the regions improved by the bilateral filter and the non-local means filter respectively. Notice that the bilateral filter improves the depth data at homogeneous regions while

(a) (b) (c) (d)

Fig. 5: Effect of δ: (a) Color image, (b) Depth image, (c) $\delta = 2$, (d) $\delta = 0$, With $\delta = 0$, the holes at reflective surfaces remain.

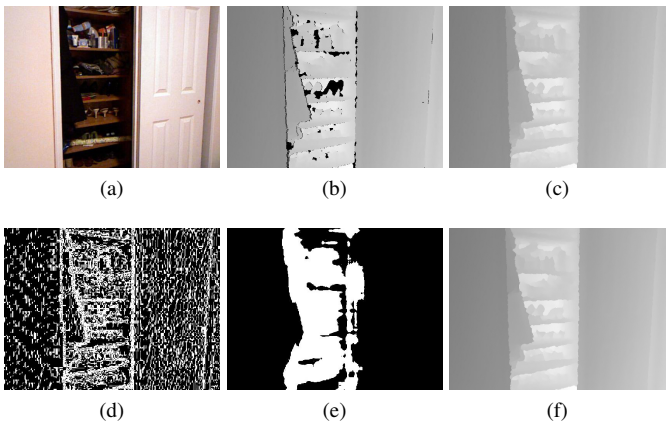


(a) (b) (c)

(d) (e) (f)

Fig. 6: (a) Original depth image, (b) After inpainting step, (c) After median hole filling step, (d)-(f) are the zoomed versions of the outlined area in (a)-(c)

non-local means filter predominantly improves the depth data along the edges where the holes were present.



(a) (b) (c)

(d) (e) (f)

Fig. 7: (a) Original image, (b) Depth image, (c) After median hole filling, (d) Region improved by bilateral filter, (e) Region improved by NLM filter, (f) Final result

In Fig. 8 (a-c), we have considered three different scenes having varying geometry and depth. Comparing the results obtained, we can see that the our result produces a more accurate depth map. Fig. 8 (j-l) show the final result of the proposed approach and Fig. 8 (g-i) show the results obtained using inpainting for hole filling. The median based iterative hole filling procedure followed by non-local denoising step (section IV-B) provides a better depth map. This method considers edges from the beginning and hence crisp edges are maintained. The object boundaries are sharp and consistent with the real object boundaries. Also notice that in Fig. 8(a) there are many black objects for which Kinect could not detect the depth but the proposed approach is successful in restoring them.

Test case	Mean error in gray value
Fronto-parallel, planar	0.39
Non- fronto parallel, planar	1.2
Edges	4.4
Random	1.63

TABLE I: Comparison with ground truth

The proposed approach took approximately 300 seconds to process a RGB-D image of size 480×640. This time was measured on a machine with Intel i7 processor, 8GB RAM and running MATLAB. The number of iterations performed by the iterative median filter component was 15.

Comparison with ground truth:
The holes filled by the median based hole filling step were compared with the ground truth. For this purpose, holes were introduced purposefully in the depth map provided by the Kinect. This was done to ensure availability of ground truth data for a large variety of controlled situations. The quality of data estimated by the median based hole filling step is analysed for different situations like, fronto-parallel setting, planar surfaces, non-planar surfaces and edges.

The size of the holes induced in the depth map was 10×10 pixels. Algorithm 1 was applied to such test cases till all the holes were removed.The depth maps used were of 8 bits per pixel. The errors shown in Table I further reduce as we apply the post processing steps (section IV-B). The error at the edges is sharply reduced by the non-local means denoising step. Some of the limitations of the proposed approach are listed below.

1) Though the proposed approach produces high quality restored depth maps, the approach is slow for processing real-time video. Using faster algorithms for finding median will be a small step in this direction.

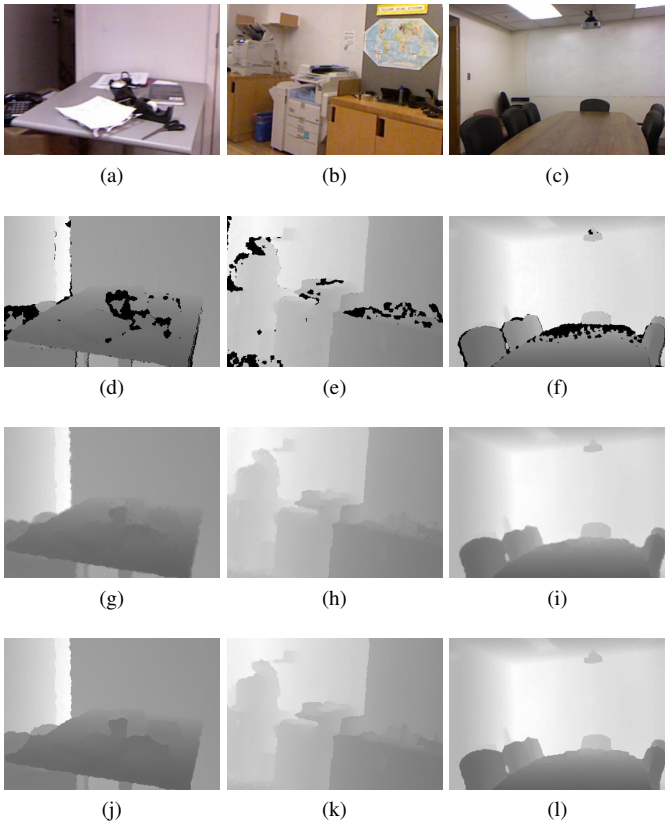


Fig. 8: (a)-(c) Color images for three different scene,(d)-(f) Depth images for three different scene, (g)-(i) Hole filling using Telea's inpainting, (j)-(l) Proposed approach

2) The proposed approach depends on the value of δ for fast convergence. Selection of the most appropriate value for δ corresponding to a given depth map is a challenge.

VI. Conclusion

In this paper, we have introduced a novel concept of using an iterative median filter with a guiding gray scale image to fill holes in the corresponding depth map. We improve the depth map further by applying a non-local means denoising on the detail layer which is obtained by passing the median filtered depth map through a bilateral filter. This method is shown to be successful in filling holes in depth maps obtained from range sensors like the Kinect. When there are no large contiguous holes, this filter acts just like a traditional median filter over the depth map. As the crucial hole filling step proposed in Algorithm 1 consists of only a comparison in contrast to weighted sums, this procedure is simple to implement even on embedded platforms where divisions are costly. This work opens up new possibilities of computer vision applications with RGB-D data. The depth map provided by Kinect sensor has been used in a variety of challenging applications in computer vision such as pose estimation, skeletal tracking, gesture recognition, 3D reconstruction, etc. The proposed approach will act as a vital tool in supplying accurate depth maps to enhance the utility of Kinect sensors in such applications. We believe that incorporating the temporal information in the proposed approach can lead to a more robust algorithm in future.

References

[1] Z. Zhang, "Microsoft kinect sensor and its effect," *Multimedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[2] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, 2013.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceedings of the 1998 IEEE International Conference on Computer Vision*, 1998.

[4] A. Buades and B. Coll, "A non-local algorithm for image denoising," in *In CVPR*, 2005, pp. 60–65.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[6] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, pp. 25 – 36, 2004.

[7] A. Dakkak and A. Husain. (2013) Recovering missing depth information from microsofts kinect.

[8] M. Camplani and L. Salgado, "Efficient spatio-temporal hole filling strategy for kinect depth maps," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2012, pp. 82 900E–82 900E.

[9] J. Shen and S.-C. S. Cheung, "Layer depth denoising and completion for structured-light rgb-d cameras," in *IEEE CVPR*, 2013, pp. 1187–1194.

[10] Y. Berdnikov and D. Vatolin, "Real-time depth map occlusion filling and scene background restoration for projected-pattern based depth cameras," in *Graphic Conf., IETP*, 2011.

[11] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by kinect depth camera," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[12] S. Milani and G. Calvagno, "Joint denoising and interpolation of depth maps for ms kinect sensors," in *IEEE ICASSP*, 2012, pp. 797–800.

[13] M. Schmeing and X. Jiang, "Color segmentation based depth image filtering," in *Proc. Int. Workshop on Depth Image Analysis*, 2012.

[14] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, "Structure guided fusion for depth map inpainting," *Pattern Recognition Letters*, 2012.

[15] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Advances in neural information processing systems*, 2005, pp. 291–298.

[16] C. D. Herrera, J. Kannala, P. Sturm, and J. Heikkila, "A learned joint depth and intensity prior using markov random fields," in *3DTV-Conference, 2013 International Conference on*. IEEE, 2013, pp. 17–24.

[17] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.

[18] F. Porikli, "Constant time o (1) bilateral filtering," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[19] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time o (1) bilateral filtering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 557–564.

[20] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 29–43.

[21] Y. Eshet, S. Korman, E. Ofek, and S. Avidan, "DCSH-Matching Patches in RGBD Images," in *IEEE ICCV*, 2013.

[22] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.