Depth Upsampling Methods for High Resolution Depth Map

Yong-Jun Chang, Sunho Kim, Yo-Sung Ho
Gwangju Institute of Science and Technology
{yjchang, sunhokim, hoyo}@gist.ac.kr

Abstract

A depth camera measures depth information of the object using a structured light or a time-of-flight method. However, those methods have a problem that the resolution of the depth map is small. This problem affects a generation of high-quality three-dimensional (3D) video contents. In this paper, five different upsampling methods for the high-resolution depth map acquisition are introduced and analyzed. Each algorithm has its own characteristics for generation of the high-resolution depth map. Therefore, this paper compares and analyzes those properties. After analyzing advantages and disadvantages of upsampling methods, we also apply depth refinement algorithms to the upsampled depth map for acquiring the high-quality depth map.

Keywords: Depth estimation, Depth camera, Depth map upsampling, Depth error detection, Depth error refinement

1. Introduction

Depth information is an important cue for 3D video contents. To acquire depth information from the object, many depth estimation methods are used such as an active sensor-based method and a passive sensor-based method. The passive sensor-based method usually uses captured scenes for the depth estimation such as stereo matching methods. On the contrary, the active sensor-based method uses depth cameras such as a structured light camera and a time-of-flight (ToF) camera.

The depth map captured by the depth cameras generally has a lower resolution than color images that are captured by general color cameras. To make the resolution of the depth image and the color image the same, many depth upsampling algorithms have been proposed. In order to make the high-resolution depth map, pixel values in the low-resolution depth map are warped to the high-resolution depth image. Since the resolution difference between the low resolution and the high-resolution depth maps, there are holes in the warped depth image.

In this paper, we introduce depth upsampling methods such as a bilinear interpolation, a joint bilateral upsampling (JBU), a noise-aware filter for depth upsampling (NAFDU), a guided image upsampling, and a discontinuity adaptive depth upsampling to fill those holes [1-5]. After that, we detect depth errors and refine those errors based on some algorithms. In the depth error refinement step, we use some filters that are used in the depth map upsampling process.

2. Analysis of Depth Upsampling Methods for High Resolution Depth Map

2.1. Bilinear Depth Upsampling

A bilinear depth upsampling method uses a bilinear interpolation for hole filling process [1]. The upsampled depth map has holes. The number of holes in the high-resolution depth map following the resolution difference between the low-resolution depth map and that of the target depth map. If the resolution of the target depth map is very large, a lot of holes occur in the upsampled depth map. To fill those holes, the bilinear depth upsampling method estimates the value of hole using neighboring pixels in the x and the y directions. This method performs the linear interpolation in the x and the y directions with weighting values depending on the distance from neighboring pixels.

2.2. Joint Bilateral Upsampling

The bilinear interpolation fills holes in the upsampled depth map simply and efficiently. Therefore, it has a low complexity of the upsampling process. However, this method has a weakness near the edge region. Since the bilinear interpolation considers the hole filling using only the weight value according to the distance of the surrounding pixels, the blurring of depth value occurs in the edge region of the object. This problem has a bad effect on the quality of 3D video contents. To relieve this problem, a joint bilateral upsampling (JBU) was proposed [2].

The JBU uses a high-resolution color image and a low-resolution depth image to make a high-resolution depth image. The color image is used as a guided image for depth preservation near the edge region. Eq.

1 represents how to calculate the upsampled depth value using the guided image.

$$\tilde{S}_p = \frac{1}{k_p}\sum_{q_\downarrow \in \Omega} S_{q_\downarrow} f(\|p_\downarrow - q_\downarrow\|) g(\|\tilde{I}_p - \tilde{I}_q\|) \quad (1)$$

In Eq. 1, where $\tilde{S}_p$ is an upsampled depth value at pixel p, $S_{q_\downarrow}$ is a depth value at neighboring pixel q in the low-resolution depth map, $\tilde{I}$ is a pixel value in the high-resolution color image, k_p is a normalization value, f is a spatial kernel, and g is a range kernel. Each kernel calculates the weighting values depending on the pixel and color distances, respectively. Since both spatial and range kernels are used as the weighting values, depth values near the edge region are preserved well. However, this method takes a lot of time because of kernel computation.

2.3. Noise-Aware Filter for Depth Upsampling

The JBU method shows good upsampling results near the edge region. However, it has a texture copy problem because of the range kernel. Since the range kernel uses the guidance color image, it gives low weighting values for textured regions that have large color differences compared to neighboring regions. To improve this problem, a noise-aware filter for depth upsampling (NAFDU) was proposed [3].

The NAFDU uses additional depth information, unlike the JBU. The depth value of the low-resolution depth map is applied to the equation of NAFDU as defined in Eq. 2.

$$\tilde{S}_p = \frac{1}{k_p}\sum_{q_\downarrow \in \Omega} S_{q_\downarrow} f(\|p_\downarrow - q_\downarrow\|)\left[\alpha(\Delta\Omega) g(\|\tilde{I}_p - \tilde{I}_q\|) + (1-\alpha(\Delta\Omega)) h(\|S_{p_\downarrow} - S_{q_\downarrow}\|)\right] \quad (2)$$

In Eq. 2, where $\alpha(\Delta\Omega)$ is a blending function for combining the range kernel g with the depth kernel h. This function value is determined by the depth variance in the kernel. If the depth variance is very large, the kernel may have some textures. In this case, the blending function has a small value. As a result, Eq. 2 gives large weighting value to h to relieve the texture copy problem. The depth kernel h is calculated using the low-resolution depth map.

2.4. Guided Image Filter for Depth Upsampling

The guided image filter preserves the edge region with a small computational complexity [4]. In [4], the time complexity of the guided image filter is $O(1)$. In this method, the guided image is used for improving the quality of the input image. In this paper, the guided image is the input image itself.

Eq. 3 to 5 show all the formulas used in the guided image filter. In Eq. 3, where I is a guided image, p is an input image, and ε is a constant value. By using these values, one of the coefficients a that are used for the equation of the output image is determined. The other coefficient b is defined in Eq. 4. In Eq. 4, where $\mathrm{E}[\cdot]$ means the mean value. The equation of the final output image q is determined by Eq. 3 and Eq. 4 as defined in Eq. 5.

Unlike the joint bilateral filter and NAFDU, which use time-consuming weight functions based on the gaussian distribution, the guided image filter requires mean, variance, and covariance values for each step. The guided image filter for depth upsampling is used for filling the holes in the upsampled depth image.

$$a = \frac{cov(I,p)}{var(I)+\varepsilon} \quad (3)$$

$$b = \mathrm{E}[p] - a \cdot \mathrm{E}[I] \quad (4)$$

$$q = I \cdot \mathrm{E}[a] + \mathrm{E}[b] \quad (5)$$

2.5. Discontinuity Adaptive Depth Upsampling

The NAFDU method removes texture copy problems in the upsampled depth map. However, it does not solve the blurring problem near the edge region clearly. A discontinuity adaptive depth upsampling (DADU) method was proposed to solve this problem efficiently [5]. The basic operation of this method is to perform adaptively depending on the local image characteristic. This method transforms the depth image to the normalized variance image to determine the local image characteristic. Eq. 6 shows the equation of the DADU method.

$$D^u{}_p = \begin{cases} \underset{d\in\Omega}{\operatorname{argmin}} \left|d - D^u{}_{JBU,p}\right|, & (\sigma_{D_{p\downarrow}}{}^2 \geq Th_d) \\ D^u{}_{JBU,p}, & \text{otherwise} \end{cases} \quad (6)$$

In Eq. 6, where $D^u{}_p$ is the final depth value at p for the high-resolution depth map, d indicates valid depth values in the filter kernel, $D^u{}_{JBU,p}$ is the upsampled depth value with the JBU method, $\sigma_{D_{p\downarrow}}{}^2$ is a pixel value in the normalized variance map, and Th_d is a threshold value.

If the value of the corresponding pixel in the normalized variance map is smaller than the certain threshold, the DADU method fills that hole using the depth value that is calculated by the JBU method. If not, the DADU method fills the hole with the depth value which is most similar value to $D^u{}_{JBU,p}$ in the kernel. The large value of $\sigma_{D_{p\downarrow}}{}^2$ means that region may be the edge region. Therefore, the DADU method does not use the JBU value in that region for the edge preserving effect.

The blurring problem near the edge region in the JBU and the NAFDU methods is caused by the filtered depth value. On the contrary, the DADU method uses the unfiltered depth value near the edge region using Eq. 6. Hence, this method preserves the depth value near the edge region well compared to the other methods.

3. High Quality Depth Reconstruction

3.1. Depth Error Detection

Depth upsampling methods in Section 2 reconstruct the high-resolution depth map from the low-resolution depth map. However, there are still some depth errors in the upsampled depth map. Especially, the depth map which is obtained by stereo matching methods usually has disparity errors in the occlusion regions and the edge regions and those errors are propagated to the upsampled depth image.

To improve those depth errors in the upsampled depth map, we apply the depth error detection and refinement algorithms to the upsampled depth map. In this paper, we focus on the enhancement of the disparity map. Therefore, the depth value of this paper means the disparity value.

First, we define the threshold value to detect the depth error in the upsampled depth map. If the depth value is smaller than the threshold, it is regarded as the error pixel.

$$\mu = \frac{1}{MN}\sum_{y\in N}\sum_{x\in M} D_u(x,y) \tag{7}$$

$$\sigma = \sqrt{\frac{1}{MN}\sum_{y\in N}\sum_{x\in M}[D_u(x,y)-\mu]^2} \tag{8}$$

$$threshold = \mu - 1.5\sigma \tag{9}$$

In Eq. 7, where μ is the mean depth value of the upsampled depth image, M is the width, and N is the height of the depth image. Eq. 8 represents the standard deviation of the depth value. From both Eq. 7 and Eq. 8, the threshold value is determined as defined in Eq. 9.

If the disparity value is lower than 15%, it is regarded as the depth error. In other words, we remove outliers in the upsampled depth map using Eq. 9.

$$a\cdot(x-x_1)+b\cdot(y-y_1)+c\cdot(d-d_1)=0 \tag{10}$$

$$d_e = -\frac{a}{c}\cdot(x-x_1)-\frac{b}{c}\cdot(y-y_1)+d_1 \tag{11}$$

The remaining depth errors are detected based on the plane equation that is defined in Eq. 10. To make the plane equation, we need at least three points. Each point includes the pixel position and the depth value at that position. In this paper, we use three points of the upsampled depth map to detect the depth error for each pixel in the depth map.

In Eq. 10, a, b, and c represent the coefficients of the plane equation, respectively. Those coefficients are calculated by the outer product of two pairs of three points. After that, we set Eq. 10 to Eq. 11. In Eq. 11, where d_e means the depth value estimated by the plane equation.

If the difference between the upsampled depth value and the estimated depth value is larger than the threshold, that pixel is regarded as the depth error pixel. Hence, the remaining depth errors are removed by this additional algorithm.

3.2. Depth Error Refinement

We used two methods for the depth error refinement. After that, we compare results of those methods each other. The first method is using both the weight function of the bilateral filter and the plane equation adaptively. The depth error refinement method using the weight function of the bilateral filter uses the weight value that is defined in Eq. 12, 13, and 14 [6]. We call this method as a modified bilateral filter (MBF) method. It finds the maximum weight value to refine the depth value. The maximum weight value means that pixel is the most similar pixel to the current pixel. Therefore, it refines the depth value using the depth information of the neighboring pixel.

In this paper, we combine this method with the plane equation method. We also call the depth refinement method using the plane equation as PLANE method

$$B(i,j) = o(j)\exp\left(-\frac{D(i,j)}{2\sigma_s^2}-\frac{D(I_i,I_j)}{2\sigma_r^2}\right) \tag{12}$$

$$o(i) = \begin{cases} 1, & d_i \neq 0 \\ 0, & otherwise \end{cases} \tag{13}$$

$$d(x,y) = arg\max_{d_j} B(i,j) \tag{14}$$

Eq. 15 represents the equation of first depth error refinement method. In Eq. 15, where d_{PLANE} is same with Eq. 11 and α is the weight value for adaptive use of both MBF and PLANE methods. This value is defined by the normalized variance value of the original color image. If the variance value of the current pixel is large, it may be the textured region. Therefore, we give the large weight value to the MBF method because the MBF method preserves the edge region in the depth image well compared to the PLANE method. Otherwise, we give the large weight value to the PLANE method because this method has better results in the homogeneous region. The second method to refine the depth error is using the guided image filter that is defined in Section 2.4.

$$d(x,y) = \alpha\cdot d_{MBF}(x,y) + (1-\alpha)\cdot d_{PLANE}(x,y) \tag{15}$$

4. Experimental Results

4.1. Performance Evaluation

We first capture the IR image and the depth map of three scenes with different distances between the object and the camera in the same space as depicted in

Table 1: Quantitative results

	Original			Variance-based MBF+PLANE			Guided Image Filtering		
	Matching ratio (%)	Spatial variance	Temporal variance	Matching ratio (%)	Spatial variance	Temporal variance	Matching ratio (%)	Spatial variance	Temporal variance
Box 1	13.01	0.25	1.0	32.14	0.24	1.2	47.19	0.26	1.1
Box 2	4.72	0.5	0.16	4.72	0.49	0.16	58.27	0.51	0.25
Box 3	1.7	0.49	0.0	4.68	0.48	0.0	51.21	0.49	0.0
Avg.	6.48	0.41	0.39	13.85	0.4	0.45	52.22	0.42	0.45

Fig. 1. Each of three scenes was captured at 30 frames, respectively. We applied the proposed methods through these scenes and calculated the quantitative results under three standards: matching ration in the edge region, depth variance in a spatial and temporal axis. To measure the matching ration in the edge region, we detect the edge region of the result depth image. After that, we calculate the matching ratio between the detected edge map and the ground truth edge map. We also measure the variance in spatial and temporal axis for comparing the depth stability.

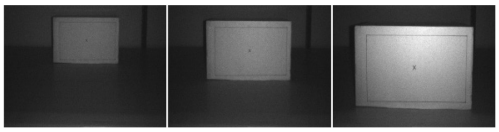

Figure 1. Three scenes capturing a box.

4.2. Comparison of Results

In this experiment, we used the DADU method for the depth map upsampling, because when we compare all the upsampling methods, DADU had better results compared with other algorithms.

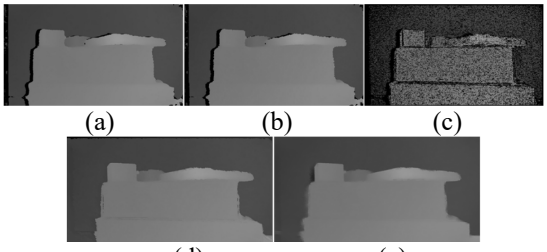

Figure 2. Depth map refinement results – (a) input depth; (b) DADU; (c) error map detection result; (d) refined by the variance-based MBF+PLANE; (e) guided image filtering.

In Fig. 2, we show the result of the depth map upsampling based on DADU, error map detection, and two refinement methods.

Table 1 shows the quantitative evaluations of Fig. 1. In Table 1, the spatial and temporal variance is always similar to the original depth map on average. However, the matching ratio in the edge region is improved. When we use the variance weight-based method, the matching ratio is increased by 7% on average compared to the original result. In additions, when we use the guided image filter, the matching ratio is increased by 46% on average.

5. Conclusion

The depth map that is captured by a depth camera generally has a low-resolution compared to a color image which is captured by the latest color camera. In this paper, five different depth upsampling algorithms are introduced and analyzed. In addition, depth error detection and refinement algorithms are also introduced to obtain the high-quality depth map. As a result, the DADU usually shows better results than the other methods in depth upsampling, and the guided image filter usually shows better results the depth error refinement step than the other methods.

Acknowledgement

This work was supported by 'The Cross-Ministry Giga KOREA Project' grant funded by the Korea government(MSIT) (GK17C0100, Development of Interactive and Realistic Massive Giga- Content Technology)

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Interpolation in Two or More Dimensions,” *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, pp. 123-128, 1992.

[2] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint Bilateral Upsampling,” *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1-5, August 2007.

[3] D. Chan, H. Buisman, C. Theobalt, and S. Thrum, “A Noise-Aware Filter for Real-Time Depth Upsampling,” *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, pp. 1-12, October 2008.

[4] K. He, J. Sun, and X. Tang, “Guided Image Filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397-1409, June 2013.

[5] S. B. Lee, S. Kwon, and Y. S. Ho, “Discontinuity Adaptive Depth Upsampling for 3D Video Acquisition,” *Electronic Letters*, vol. 49, no. 25, pp. 1612-1614, December 2013.

[6] D. Min, S. Yea, and A. Vetro, “Occlusion Handling Based on Support and Decision,” *IEEE International Conference on Image Processing*, pp. 1111-1780, September 2010.