High Quality Depth Map Upsampling for 3D-TOF Cameras

Jaesik Park[†] Hyeongwoo Kim[†*] Yu-Wing Tai[†] Michael S. Brown[§] Inso Kweon[†]
Korea Advanced Institute of Science and Technology (KAIST)[†]
National University of Singapore (NUS)[§]

Abstract

This paper describes an application framework to perform high quality upsampling on depth maps captured from a low-resolution and noisy 3D time-of-flight (3D-ToF) camera that has been coupled with a high-resolution RGB camera. Our framework is inspired by recent work that uses nonlocal means filtering to regularize depth maps in order to maintain fine detail and structure. Our framework extends this regularization with an additional edge weighting scheme based on several image features based on the additional high-resolution RGB input. Quantitative and qualitative results show that our method outperforms existing approaches for 3D-ToF upsampling. We describe the complete process for this system, including device calibration, scene warping for input alignment, and even how the results can be further processed using simple user markup.

1. Introduction

Active 3D time-of-flight (3D-ToF) cameras are becoming a popular alternative to stereo-based range sensors. Such 3D-ToF cameras use active sensing to capture 3D range data at frame-rate as a per-pixel depth. A light source from the camera emits a near-infrared wave which is then reflected by the scene and is captured by a dedicated sensor. Depending on the distance of the objects in the scene, the captured light wave is delayed in phase compared to the original emitted light wave. By measuring the phase delay, the distance at each pixel can be estimated. The resolution of the depth map captured by 3D-ToF cameras is relatively low; typically less then 1/4th the resolution of a standard definition video camera. In addition, the captured depth maps are often corrupted by significant amounts of noise.

The goal of this paper is to estimate a high quality high-resolution depth map from the 3D-ToF through upsampling in the face of sensor noise. To aid this procedure, an auxiliary high-resolution conventional camera is coupled with the 3D-ToF camera to synchronously capture the scene. Related work [19, 4, 7] also using coupled device setups for depth map upsampling have focused primarily on image filtering techniques such as joint bilateral filtering [8, 12] or variations. Such filtering techniques can often over smooth results, especially in areas of fine structure.

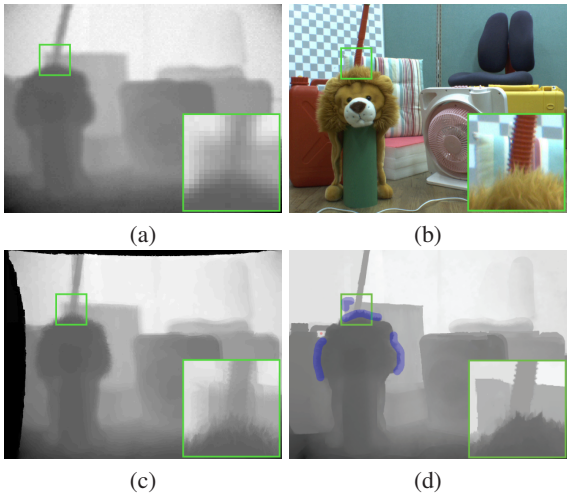


Figure 1. (a) Low-resolution depth map (enlarged using nearest neighbor upsampling), (b) high-resolution RGB image, (c) result from [19], (d) our result. User scribble areas (blue) and the additional depth sample (red) are highlighted. The dark areas in (c) are the areas without depth samples after registration. Full resolution comparisons are provided in the supplemental materials.

We formulate the depth map upsampling problem using constrained optimization. Our approach is inspired by the recent success of nonlocal means regularization for depth map construction from depth-from-defocus [9]. In particular, we describe how to formulate the problem into a least-squares optimization that combines nonlocal means regularization together with an edge weighting scheme that further reenforces fine details. We also employ scene warping to better align the low-resolution imagery to the auxiliary camera input. While this work is more applied in nature, the result is a system that is able to produce high-quality upsampled depth maps superior in quality to prior work. In addition, our approach can be easily extended to incorporate simple user markup to correct errors along disconti-

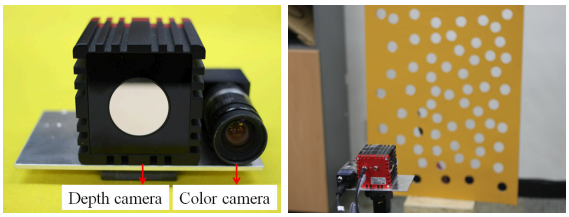
*The first and the second authors provided equal contributions to this work.

nuity boundaries without explicit image segmentation (e.g. Figure 1).

2. Related Work

Previous work on depth map upsampling can be classified as either *image fusion techniques* that combine the low-resolution depth map with the high-resolution image or *super-resolution techniques* that merge multiple misaligned low-resolution depth maps. Our approach falls into the first category of image fusion which is the focus of the related work presented here. Image fusion approaches assume there exists a joint occurrence between depth discontinuities and image edges and that regions of homogenous color have similar 3D geometry [22, 16]. Representative image fusion approaches include [6, 19, 4, 7]. In [6], Diebel and Thrun performed upsampling using an MRF formulation with the data term computed from the depth map and weights of the smoothness terms between estimated high-resolution depth samples derived from the high-resolution image. Yang et al. [19] used joint bilateral filtering [8, 12] to interpolate the high-resolution depth values. Since filtering can often over smooth the interpolated depth values, especially along the depth discontinuity boundaries, they quantized the depth values into several discrete layers. This work was later extended by [21] to use a stereo camera for better discontinuity detection in order to avoid over smoothing of depth boundaries. Chan et al. [4] introduced a noise-aware bilateral filter that decides how to blend between the results of standard upsampling or joint bilateral filtering depending on the depth map's regional statistics. Dolson et al. [7] also used a joint bilateral filter scheme, however, their approach includes additional time stamp information to maintain temporal coherence for the depth map upsampling in video sequences.

The advantage of these bilateral filtering techniques is they can be performed quickly; e.g. Chan et al. [4] reported near real-time speeds using a GPU implementation. However, the downside is they involve can still over smooth fine details. Work by [14] proposed a joint global mode filter based on global image histograms of the low-resolution depth and high-resolution image. Our approach is more related to Diebel and Thrun [6] in that we formulate the problem using an MRF optimization scheme. However, our approach incorporates a nonlocal means (NLM) term in the MRF to help preserve local structure. This additional NLM term was inspired by the recent work by Favaro [9] which demonstrated that NLM filtering is useful maintaining fine details even with noisy input data. Work in [11] has also used the NLM to fuse the 3D point cloud and 2D image to enhance the density of 3D points. We also include an additional weighting scheme based on several image derive features to further reinforce the preservation of fine detail. In addition, we perform a warping step to better align the low-resolution and high-resolution input. Our experimental results on ground truth data shows that our application framework can outperform existing techniques for the majority of scenes with various upsampling factors. Since our goal is high-quality depth maps, the need for manual cleanup for machine vision related input is unavoidable. Another advantage of our approach is that it can easily incorporate user markup to improve the results.



(a) (b)

Figure 2. (a) Our imaging setup uses a 3D-ToF camera which captures images at 176×144 resolution that is synchronized with a 1280×960 resolution RGB camera. (b) Our calibration configuration that uses a planar calibration pattern with holes to allow the 3D-ToF camera to be calibrated.

3. System Setup and Preprocessing

In this section, we describe our system and the preprocessing step to register the 3D-ToF camera and conventional camera and to perform an initial outlier rejection on the 3D-ToF input.

3.1. System Configuration

Figure 2(a) shows our hardware configuration consisting of a 3D-ToF camera and a high-resolution RGB camera. For the depth camera, we use the SwissRanger^{TM} SR4000 [1] which captures a 176×144 depth map. For the RGB camera, we use the Point Grey Research Flea RGB camera with a resolution of 1280×960 pixels. Since the data captured from the two cameras have slightly different viewpoints, we need to register the camera according to the depth values from the low-resolution depth map.

3.2. Depth Map Registration

Let $\mathbf{X}_d = (X, Y, Z, 1)^\top$ be a 3D homogeneous coordinate acquired by the 3D-ToF camera, and $\mathbf{x}_c = (u, v, 1)^\top$ be the 2D homogeneous coordinate of the high-resolution RGB image. We can compute a projection of $\mathbf{X}_d$ onto $\mathbf{x}_c$ by:

$$s\mathbf{x}_c = \mathsf{K} \begin{bmatrix} \mathsf{R} & | & \mathsf{t} \end{bmatrix} \mathbf{X}_d, \tag{1}$$

where s is a scale factor, K is the intrinsic parameters of the RGB camera, and R and t are the rotation and translation matrix which describe the rotation and translation of the the RGB camera and the depth camera with respect to the 3D world coordinate.

To calibrate the two cameras' parameters, we use the calibration method introduced by Zhang [20]. Since the 3D-ToF camera cannot capture textures, we instead use a planar calibration pattern consisting of holes for our purpose (Figure 2(b)). This unique calibration pattern allows us to detect the positions on the planar surface that are observed by the 3D-ToF camera. After camera calibration, for any point, $\mathbf{x}_t$, on the low-resolution depth map with depth value d_t, we can compute its corresponding position in the high-resolution RGB image by the following equation:

$$s\mathbf{x}_c = \mathsf{K_c} \begin{bmatrix} \mathsf{R} & | & \mathsf{t} \end{bmatrix} \mathsf{P_t}^{-1} [\mathbf{x}_t \;\; d_t \;\; 1]^T \qquad (2)$$

where $\mathsf{P_t}$ is the 4×4 projective transformation converting the world coordinate $\mathbf{X}_d$ into the local coordinate of the 3D-ToF camera. We obtain the scaling term s by calculating the relative resolution between the depth camera and the RGB camera. Since the depth map from the depth camera is noisy, we impose a neighborhood smoothness regularization using thin-plate splines to forward map the low-resolution depth map to the high-resolution image.

3.3. Outliers Detection

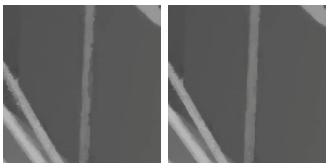
The depth map from the 3D-ToF camera contains depth edges that are blurred by mixing the depth values of two different depth layers along depth boundaries. These blurred depth boundaries are unreliable and should be removed before upsampling. For each pixel in the low-resolution depth map, we compare the depth value of a pixel to the local maximum depth and the local minimum depth within a small local window (e.g. 9×9) in the low-resolution depth map. The contrast between the local maximum and minimum depth determines whether this local window contains two different depth layer. If the depth value of a pixel is at the middle of the two depth layers, we consider this pixel as a boundary pixel. Since the input depth map is noisy, we use an MRF [3] to clean up the noisy estimation as:

$$E(\mathbf{l}) = \sum_p \left(\mathbf{O}_p(\mathbf{l}) + \lambda_{pq} \sum_{q \in N(p)} \mathbf{O}_{pq}(\mathbf{l}) \right), \qquad (3)$$

where $\mathbf{l} \in [0, 1]$ is a map of binary label indicating whether a pixel is an outlier or not, $\mathbf{O}_p(\mathbf{l})$ is the data term defined by the extent of contrast within a small window, $\mathbf{O}_{pq}(\mathbf{l})$ is the smoothness term defined by the Hamming distance between $\mathbf{l}_p$ and its neighbor $\mathbf{l}_q$. This simple outlier rejection step is performed on each input frame captured by the 3D-ToF camera.

4. Optimization Framework

This section describes our optimization framework for unsampling the low-resolution depth map given the aligned sparse depth samples and the high-resolution RGB image.



(a) (b)

Figure 3. Comparison of our result without (a) and with (b) the NLM term. The same weighting scheme proposed in Section 4.2 is used for both (a) and (b). Although the usage of NLM does not significantly affect the RMS error, it is important in generating high quality depth maps especially along thin structure elements.

Similar to the previous image fusion approaches [6, 19, 7], we assume there are co-occurrences of depth boundaries and image boundaries.

4.1. Objective Function

We define the objective function for depth map upsampling as follows:

$$E(\mathbf{D}) = E_{\mathrm{d}}(\mathbf{D}) + \lambda_{\mathbf{s}} E_{\mathrm{s}}(\mathbf{D}) + \lambda_{\mathrm{N}} E_{\mathrm{NLM}}(\mathbf{D}) \qquad (4)$$

where $E_{\mathrm{d}}(\mathbf{D})$ is the data term, $E_{\mathrm{s}}(\mathbf{D})$ is the neighborhood smoothness term, and $E_{\mathrm{NLM}}(\mathbf{D})$ is a NLM regularization. The term $\lambda_{\mathbf{s}}$ and λ_{N} are the relative weights to balance the energy between the three terms. Note that the smoothness term and NLM term could be combined into a single term, however, we keep them separated here for sake of clarity.

Our data term is defined according to the initial sparse depth map:

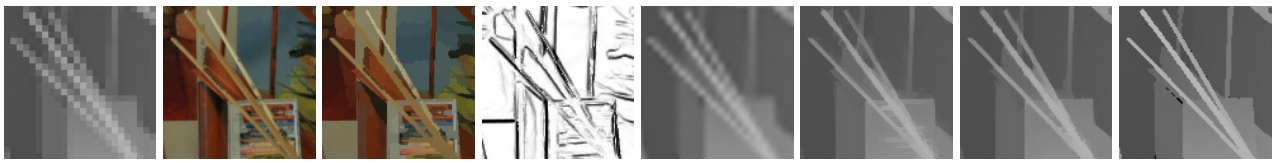
$$E_{\mathrm{d}}(\mathbf{D}) = \sum_{p \in \mathcal{G}} \left(\mathbf{D}(p) - \mathbf{G}(p) \right)^2, \qquad (5)$$

where $\mathcal{G}$ is a set of pixels which has the initial depth value. Our smoothness term is defined as:

$$E_{\mathrm{s}}(\mathbf{D}) = \sum_p \sum_{q \in \mathcal{N}(p)} w_{pq} \left(\mathbf{D}(p) - \mathbf{D}(q) \right)^2, \qquad (6)$$

where $\mathcal{N}(p)$ is the first order neighborhood of p, and w_{pq} is the confidence weighting which will be detailed in the following section. Combining Equation (5) and Equation (6) forms a quadratic objective function which is similar to the objective function in [13]. The work in [13] was designed to propagate sparse color values to a gray high-resolution image, which is similar in nature to our problem of propagating sparse depth values to the high-resolution RGB image.

The difference between our method and that of [13] is the definition of w_{pq}. Work in [13] defined w_{pq} using intensity difference between the first order neighborhood pixels to preserve discontinuities. We further combine segmentation, color information, and edge saliency as well as the bicubic



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4. (a) Low-resolution depth map (enlarged using nearest neighbor upsampling). (b) High-resolution RGB image. (c) Color segmentation by [17]. (d) Edge saliency map. (e) Guided depth map by using bicubic interpolation of (a). (f) Our upsampling result without the guided depth map weighting, depth bleeding occurred in highly textured regions. (g) Our upsampling result with guided depth map weighting. (h) Ground truth. We subsampled the depth value of a dataset from Middlebury to create the synthetic low-resolution depth map. The magnification factor in this example is $5\times$. The sum of squared difference(SSD) between (f) and (g) comparing to the ground truth are 31.66 and 24.62 respectively. Note that the depth bleeding problem in highly textured regions has been improved.

upsampled depth map to define w_{pq}. The reason for this is that we find the first order neighborhood does not properly consider the image structure. As the result, propagated color information in [13] was often prone to bleeding errors near fine detail. In addition, we include a NLM regularization term, which protects the thin structures by allowing the pixels on the same nonlocal structure to reinforce each other within a larger neighborhood. We define the NLM regularization term using an anisotropic structural-aware filter [5]:

$$E_{\text{l}}(\mathbf{D}) = \sum_{p} \sum_{q \in \mathcal{A}(p)} \kappa_{pq} \left(\mathbf{D}(p) - \mathbf{D}(q)\right)^2, \qquad (7)$$

where $\mathcal{A}(p)$ is a local window (e.g. 11×11) in the high-resolution image, κ_{pq} is the weight of the anisotropic structural-aware filter defined as:

$$\begin{aligned} \kappa_{pq} &= \frac{1}{2}\left(\exp(-(p-q)^T \Sigma_p^{-1}(p-q)) + \right. \\ &\quad \left. \exp(-(p-q)^T \Sigma_q^{-1}(p-q))\right), \\ \Sigma_p &= \frac{1}{|\mathcal{A}|} \sum_{p' \in \mathcal{A}(p)} \nabla I(p') \nabla I(p')^T. \end{aligned} \qquad (8)$$

Here, $\nabla I(p) = \{\nabla_x I(p), \nabla_y I(p)\}^T$ is the x- and y- image gradient vector at p, and I is the high-resolution color image. The term Σ_q is defined similarly to Σ_p. This anisotropic structural-aware filter defines how likely p and q are on the same structure in the high-resolution RGB image, i.e. if p and q are on the same structure, κ_{pq} will be large. This NLM filter essential allows similar pixel to reinforce each other even if they are not first-order neighbors. To maintain the sparsity of the linear system, we remove neighborhood entries with $\kappa_{pq} < t$. A comparison of our approach on the effectiveness of the NLM regularization is shown in Figure 3.

4.2. Confidence Weighting

In the section, we describe our confidence weighting scheme for defining the weights w_{pq} in Equation (6). The value of w_{pq} defines the spatial coherence of neighborhood pixels. The larger w_{pq} is, the more likely that the two neighborhood pixels having the same depth value. Our confidence weighting is decomposed into four terms based on color similarities (w_c), segmentation (w_s), edge saliency (w_e), and guided bicubic interpolated depth map (w_d).

The color similarity term is defined in the YUV color space as follows:

$$w_c = \exp -\left(\sum_{I \in YUV} \frac{(\mathbf{I}(p) - \mathbf{I}(q))^2}{2\sigma_{\mathbf{I}}^2}\right), \qquad (9)$$

where $\sigma_{\mathbf{I}}$ controls the relative sensitivity of the different color channels.

Our second term is defined based on color segmentation using the library provided in [17] to segment an image into super pixels as shown in Figure 4(c). For the neighborhood pixels that are not within the same super pixel, we give a penalty term defined as:

$$w_s = \begin{cases} 1 & \text{if } \mathbf{S}_{\text{co}}(p) = \mathbf{S}_{\text{co}}(q) \\ t_{\text{se}} & \text{otherwise} \end{cases} \qquad (10)$$

where $\mathbf{S}_{\text{co}}(\cdot)$ is the segmentation label, t_{se} is the penalty factor with its value between 0 and 1. In our implementation, we empirically set it equal to 0.7.

Inspired by [2], we have also included a weight which depends on the edge saliency response. Different from the color similarity term, the edge saliency responses are detected by a set of Gabor filters with different sizes and orientations. The edge saliency map contains image structures rather than just color differences between neighborhood pixels. We combine the responses of different Gabor filters to form the edge saliency map as shown in Figure 4(d). Our weighting is computed as:

$$w_e = \frac{1}{\sqrt{s_x(p)^2 + s_x(q)^2} + 1}, \qquad (11)$$

where $s_x(\cdot)$ is the value of $x-$axis edge saliency map if p and q are $x-$axis neighborhoods.

Allowing the depth values to propagate freely with only very sparse data constraint can lead to severe depth bleeding. Here, we introduce the guided depth map to resolve

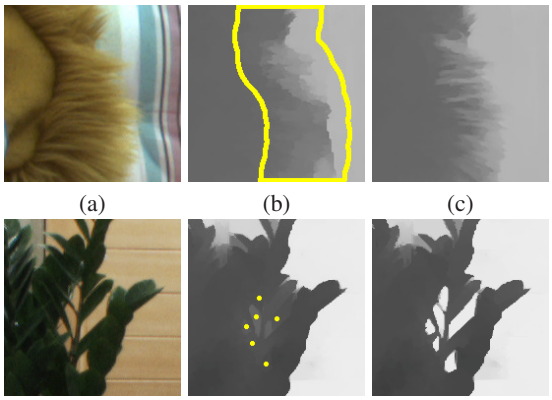


Figure 5. Depth map refinement via user markup. (a)(d) Color image of small scale structure. (b)(e) Upsampled depth map before user correction. The user scribble areas in (b), and the user added depth samples in (e) are indicated by the yellow lines and dots respectively. (c)(f) Refined depth maps.

this problem. The guided depth map weighting is similar to the intensity weighting in a bilateral filter. Since we do not have a depth sample at each high-resolution pixel location, we use bicubic interpolation to obtain the guided depth map, $\mathbf{D}_g$, as shown in Figure 4(e). Similar to the bilateral filter, we define the guided depth map weighting as follow:

$$w_d = \exp -(\frac{(\mathbf{D}_g(p) - \mathbf{D}_g(q))^2}{2\sigma_g^2}), \tag{12}$$

Combining the weight defined from Equation (9) to Equation (13) by multiplication, we obtain the weight $w_{pq} = w_s w_c w_e w_d$. Note that except for the edge saliency term, all the weighting defined in this subsection can be applied to the weighting κ_{pq} via multiplication to the NLM regularization term.

4.3. User Adjustments

Since the goal is high-quality upsampling, it is inevitable that some depth frames are going to require user touch up, especially if the data is intended for media related applications. Our approach allows easy user corrections by direct manipulation of the weighting term w_{pq} or by adding additional sparse depth sampling for error corrections.

For the manipulation of the weighting term, we allow the user to draw scribbles along fuzzy image boundaries, or along the boundaries where the image contrast is low. These fuzzy boundaries or low contrast boundaries represent difficult regions for segmentation and edge saliency detection. As a result, they cause depth bleeding in the reconstructed high-resolution depth map as illustrated in Figure 5(b). Within the scribble areas, we compute an alpha matte based on the work by Wang et al. [18] for the two different depth layers. An additional weighting term will be added according to the estimated alpha values within the

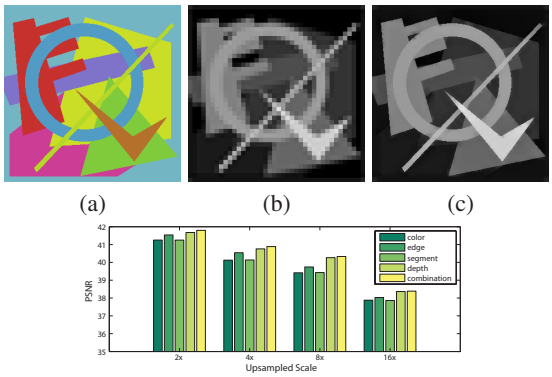


Figure 6. A synthetic example for self-evaluation of our weighting term. (a)(b) A synthetic image pair consists of high-resolution color image and low-resolution depth image. (c) Our $4\times$ upsampled depth map with the combined weighting term. (d) The plot of PSNR accuracy against the results with the combined weighting term and the results with each weighting term individually. The combined weighting term consistently produce the best results under different upsampling scale.

scribble areas. For the two pixels p and q within the scribble areas, if they belong to the same depth layer, they should have the same or similar alpha value. Hence, our additional weighting term for counting the additional depth discontinuity information is defined as:

$$\exp -(\frac{(\alpha(p) - \alpha(q))^2}{2\sigma_\alpha^2}), \tag{13}$$

where $\alpha(\cdot)$ is the estimated alpha values within the scribble areas. Figure 5(c) shows the effects after adding this alpha weighting term. The scribble areas are indicated by the yellow lines in Figure 5(b).

Our second type of user correction allows the user to draw or remove depth samples on the high-resolution depth map directly. When adding a depth sample, the user can simply pick a depth value from the computed depth map and then assign this depth value to locations where depth samples are "missing". After adding the additional depth samples, our algorithm generates the new depth map using the new depth samples as a hard constraint in Equation (4). The second row of Figure 5 shows an example of this user correction. Note that for image filtering techniques, such depth sample correction can be more complicated to incorporate since the effect of new depth samples can be filtered by the original depth sample within a large local neighborhood. Removal of depth samples can also cause a hole in the result of image filtering techniques.

4.4. Evaluation on the Weighting Terms

Our weighting term, w_{pq}, is a combination of several heuristic weighting terms. Here we provide some insight to the relative effectiveness of each individual weighting term

Synthetic	Time(Sec.)	Real-world	Time(Sec.)
Art	21.60	Lion	18.60
Books	26.47	Office with person	16.65
Mobius	24.07	Lounge	18.28
		Classroom	19.00

Table 1. Running time of our algorithm for 8x upsampling. The upsampled depthmap resolution is 1376×1088 for Synthetic and 1280×960 for Real-world examples. The algorithm was implemented using unoptimized matlab code.

and their combined effect as shown in Figure 6. Our experiments found that using only the color similar term can still cause propagation errors. The edge cue is more effective in preserving structure, but cannot entirely remove propagation errors. The effect of the segmentation cue is similar to the color cue as the segmentation is also based on color information, but generally produces sharper boundary with piecewise smoothed depth inside each segment. The depth cue is good in avoiding propagation bleeding, but is not effective along the depth boundaries because it ignores the co-occurrence of image edges and depth edges. After combining the four different cues together, the combined weighting scheme shows the best results. The results produced with the combined weighting term can effectively utilize the structures in the high-resolution RGB image while it can avoid bleeding by including the depth cue which consists with the low-resolution depth map.

5. Results and Comparisons

We tested our approach using both synthetic examples and real world examples as described in the following sections. The value of λ_s, λ_N are chosen as 0.2 and 0.1 respectively, and they are fixed during our experiments. The system configuration for experiments is 3Ghz CPU, 8GB RAM. We implemented our algorithm via Matlab using its built-in standard linear solver. The computation time is summarized in Table 1.

5.1. Evaluations using the Middlebury stereo dataset

We use synthetic examples for quantitative comparisons with the results from previous approaches [6, 19, 10]. The depth map from the Middlebury stereo datasets [15] are used as the ground truth. We down sampled the ground truth depth map by different factors to create the low-resolution depth map. The original color image is used as the high-resolution RGB image. We compare our results with bilinear interpolation, MRF [6], bilateral filter [19], and a recent work on guided image filter [10]. Since the previous approaches do not contain a user correction step, the results generated by our method for these synthetic examples are all based on our *automatic* method in Section 4.1 and Section 4.2 for fair comparisons. Table 2 summaries the RMSE (root-mean-square error) against the ground truth under different magnification factors for different testing examples. Our results consistently achieved the lowest RMSE among all the test cases especially for large scale upsampling. The qualitative comparison with the results from [6] and [19] under $8\times$ magnification factor can be found in Figure 7.

In terms of depth map quality, we found that the MRF method in [6] produces the most blurred result. This is due to its simple use of neighborhood term which considers only the image intensity difference as the neighborhood similarity for depth propagation. The results from bilateral filtering in [19] are comparable to ours with sharp depth discontinuities in some of the test examples. However, since segmentation and edge saliency are not considered, their results can still suffer from depth bleeding highly textured regions. We also found that for the real world example in Figure 1, the results from [19] tended to be blurry.

5.2. Robustness to Depth Noise

The depth map captured by 3D-ToF cameras are always noisy. We compare the robustness of our algorithm and the previous algorithms by adding noise. We also compare against the Noise-Aware bilateral filter approach in [4]. We observe that the noise characteristics in a 3D-ToF camera depends on the distance between the camera and the scene. To simulate this effect, we add a conditional Gaussian noise:

$$p(\mathbf{x}, k, \sigma_d) = k \exp -(\frac{\mathbf{x}}{2(1+\sigma_d)^2}), \tag{14}$$

where σ_d is a value proportional to the depth value, and k is the magnitude of the Gaussian noise. Although the actual noise distribution of 3D-ToF camera is more complicated than the Gaussian noise model, many previous depth map upsampling algorithms do not consider the problem of noise in the low-resolution depth map. This experiment therefore attempts an objective comparison on the robustness of different algorithms with respect to noisy depth maps. The results in term of RMSE are summarized in Table 3.

5.3. Real World Examples

Figure 8 shows the real world examples of our approach. Since the goal of our paper is to obtain high quality depth maps, we include user corrections for the examples in the top and middle row. We show our upsampled depth as well as a novel view rendered by using our depth map. The magnification factors for all these examples are $8\times$. These real world examples are challenging with complicated boundaries and thin structures. Some of the objects contain almost identical colors but with different depth values. Our approach is successful in distinguishing the various depth layers with sharp boundaries. All results without user corrections can be found in the supplemental materials.

(a) (b) (c)

Figure 7. Qualitative comparison on Middlebury dataset. (a) MRFs optimzation [6]. (b) Bilateral filtering with subpixel refinement [19]. (c) Our results. The image resolution are enhanced by $8\times$. Note that we do not include any user correction in these synthetic testing cases. The results are cropped for the visualization, full resolution comparisons are provided in the supplemental materials.

	Art				Books				Mobius			
	$2\times$	$4\times$	$8\times$	$16\times$	$2\times$	$4\times$	$8\times$	$16\times$	$2\times$	$4\times$	$8\times$	$16\times$
Bilinear	0.56	1.09	2.10	4.03	0.19	0.35	0.65	1.24	0.20	0.37	0.70	1.32
MRFs [6]	0.62	1.01	1.97	3.94	0.22	0.33	0.62	1.21	0.25	0.37	0.67	1.29
Bilateral [19]	0.57	0.70	1.50	3.69	0.30	0.45	0.64	1.45	0.39	0.48	0.69	1.14
Guided [10]	0.66	1.06	1.77	3.63	0.22	0.36	0.60	1.16	0.24	0.38	0.61	1.20
Ours	0.43	0.67	1.08	2.21	0.17	0.31	0.57	1.05	0.18	0.30	0.52	0.90

Table 2. Quantitative comparison on Middlebury dataset. The error is measured in RMSE for 4 different magnification factors. The performance of our algorithm is the best among all compared algorithm. Note that no user correction is included in these synthetic testing examples.

6. Discussion and Summary

We have presented a framework to upsample a low-resolution depth map from the 3D-ToF camera using an auxiliary high-resolution RGB image. Our framework is based on a least-square optimization that combines several weighting factors together with nonlocal means filtering to maintain sharp depth boundaries and to prevent depth bleeding during propagation. Although this work is admittedly more engineering in nature, we believe it provides useful insight on various weighting strategies for those working with noisy range senors. Moreover, experimental result show that our results typically out performs previous work in terms of both RMSE and visual quality. In addition to the automatic method, we have also discussed how to extend our approach to incorporate user markup. Our user correction method is simple and intuitive and does not require any addition modifications in order to solve the objective function defined in Section 4.1.

7. Acknowledgements

We are grateful to anonymous reviewers for their constructive comments. This research was partially supported by Samsung Advanced Institute of Technology (RRA0109ZZ-61RF-1) and the National Strategic R&D Program for Industrial Technology, Korea. Yu-Wing Tai was supported by the National Research Foundation (NRF) of Korea (2011-0013349) and the Ministry of Culture, Sports and Tourism (MCST) and Korea Content Agency (KOCCA) in the Culture Technology Research and Development Program 2011. Michael S. Brown was supported by the Singapore Academic Research Fund (AcRF) Tier 1 Grant (R-252-000-423-112).

References

[1] SwissRanger^{TM} SR4000 data sheet, http://www.mesa-imaging.ch/prodview4k.php.

[2] P. Bhat, C. L. Zitnick, M. F. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.*, 29(2), 2010.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 2001.

[4] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A noise aware filter for real-time depth upsampling. In *ECCV Workshop on Multicamera and Multimodal Sensor Fusion Algorithms and Applications*, 2008.

[5] J. Chen, C. Tang, and J. Wang. Noise brush: interactive high quality image-noise separation. *ACM Trans. Graphics*, 28(5), 2009.

[6] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *NIPS*, 2005.

[7] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling range data in dynamic environments. In *CVPR*, 2010.

[8] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graphics*, 23(3):673–678, 2004.

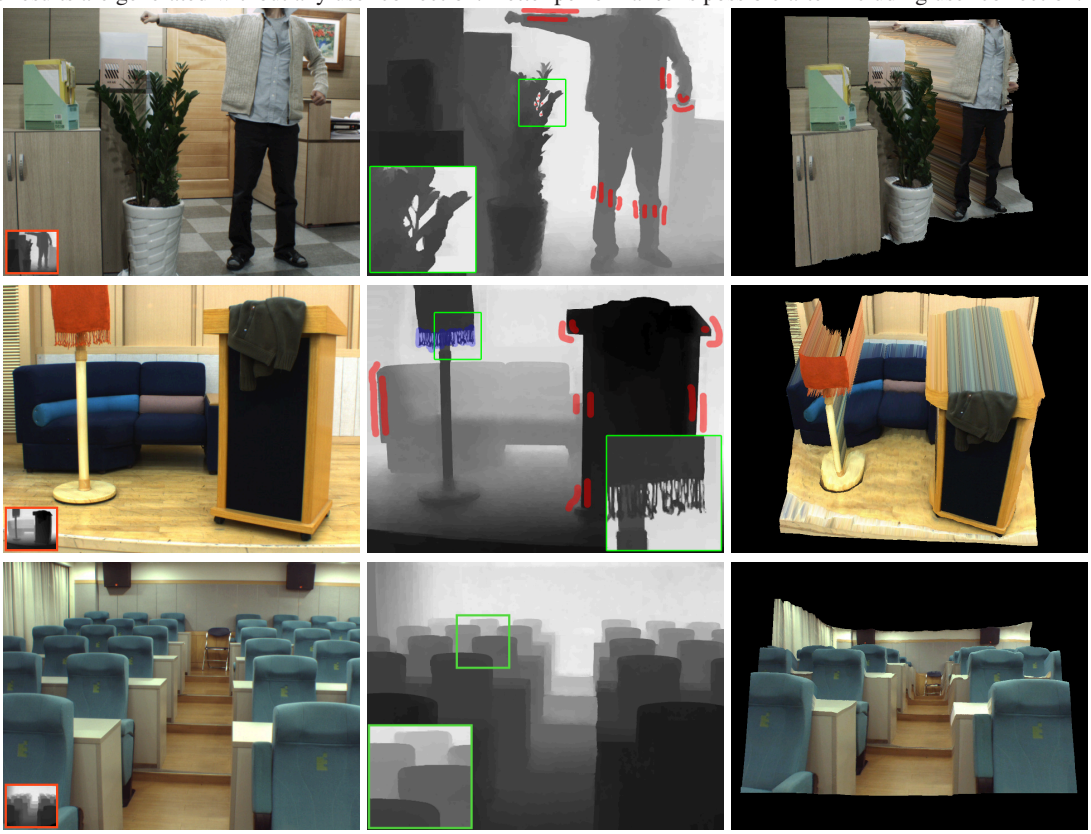
[9] P. Favaro. Recovering thin structures via nonlocal-means regularization with application to depth from defocus. In *CVPR*, 2010.

[10] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.

[11] B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *CVIU*, 114(12):1136–1345, 2010.

	Art				Books				Mobius			
	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
Bilinear	3.09	3.59	4.39	5.91	2.91	3.12	3.34	3.71	3.21	3.45	3.62	4.00
MRFs [6]	1.62	2.54	3.85	5.70	1.34	2.08	2.85	3.54	1.47	2.29	3.09	3.81
Bilateral [19]	1.36	1.93	2.45	4.52	1.12	1.47	1.81	2.92	1.25	1.63	2.06	3.21
Guided [10]	1.92	2.40	3.32	5.08	1.60	1.82	2.31	3.06	1.77	2.03	2.60	3.34
NAFDU [4]	1.83	2.90	4.75	7.70	1.04	1.36	1.94	3.07	1.17	1.55	2.28	3.55
Ours	1.24	1.82	2.78	4.17	0.99	1.43	1.98	3.04	1.03	1.49	2.13	3.09

Table 3. Quantitative comparison on Middlebury dataset with additive noise. Our algorithm achieves the lowest RMSE in most cases. Note that all these results are generated without any user correction. Better performance is possible after including user correction.



(a) (b) (c)

Figure 8. (a) Our input, the low-resolution depth maps are shown on the lower left corner (Ratio between the two images are preserved). (b) Our results. User scribble areas (blue) and the additional depth sample (red) were high-lighted. (c) Novel view rendering of our result. Note that no user markup is required in our results in the third row. More results can be found in supplemental materials

[12] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graphics*, 26(3):96, 2007.

[13] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graphics*, 23(3):689–694, 2004.

[14] D. Min, J. Lu, and M. Do. Depth video enhancement based on weighted mode filtering. *IEEE TIP*, to appear.

[15] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002.

[16] A. Torralba and W. T. Freeman. Properties and applications of shape recipes. In *CVPR*, pages 383–390, 2003.

[17] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[18] J. Wang, M. Agrawala, and M. Cohen. Soft scissors: An interactive tool for realtime high quality matting. *SIGGRAPH'07*.

[19] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *CVPR*, 2007.

[20] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. PAMI*, 22(11):1330–1334, 2000.

[21] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *CVPR*, 2008.

[22] A. Zomet and S. Peleg. Multi-sensor super-resolution. In *WACV*, pages 27–31, 2002.