Plane Fitting and Depth Variance Based Upsampling for Noisy Depth Map from 3D-ToF Cameras in Real-time

Kazuki Matsumoto, Francois de Sorbier and Hideo Saito
Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan

Keywords: Depth map, ToF depth sensor, GPU, Plane Fitting, Upsampling, denoising.

Abstract: Recent advances of ToF depth sensor devices enables us to easily retrieve scene depth data with high frame rates. However, the resolution of the depth map captured from these devices is much lower than that of color images and the depth data suffers from the optical noise effects. In this paper, we propose an efficient algorithm that upsamples depth map captured by ToF depth cameras and reduces noise. The upsampling is carried out by applying plane based interpolation to the groups of points similar to planar structures and depth variance based joint bilateral upsampling to curved or bumpy surface points. For dividing the depth map into piecewise planar areas, we apply superpixel segmentation and graph component labeling. In order to distinguish planar areas and curved areas, we evaluate the reliability of detected plane structures. Compared with other state-of-the-art algorithms, our method is observed to produce an upsampled depth map that is smoothed and closer to the ground truth depth map both visually and numerically. Since the algorithm is parallelizable, it can work in real-time by utilizing highly parallel processing capabilities of modern commodity GPUs.

1 INTRODUCTION

In recent years, depth images have gained popularity among many research fields including 3D reconstruction for dynamic scenes, augmented reality and environment perception in robotics. Depth images are often obtained by stereo vision techniques, which are computationally expensive and not able to calculate the range data in non-texture scenes. This problem was solved by the development of 3D time-of-flight (3D-ToF) depth cameras, such as MESA Swissranger and SoftKinetic DepthSense. A light source from the camera emits a near-infrared wave to 3D objects and the reflected light from scene objects is captured by a dedicated sensor. By calculating the phase shift between the emitted light and the received one, the distance at each pixel can be estimated. Thus, ToF depth cameras can acquire the range data even from textureless scenes in high frame rates.

However, the depth map captured by ToF depth camera is unable to satisfy the requirements for developing rigorous 3D applications. This is due to the fact that the resolution of the depth image is relatively low (e.g. 160×120 pixels for SoftKinetic DepthSense DS311) and the data is heavily contaminated with structural noise. Moreover, the noise increases if the infrared light interferes with other light sources or is reflected irregularly by the objects.

In this paper, we propose joint upsampling and denoising algorithm for depth data from ToF depth cameras, which is based on local distribution of the depth map. The upsampling is performed by simultaneously exploiting the depth variance based joint bilateral upsampling and the plane fitting based on the locally planar structures of the depth map. In order to detect the planar area, we combine normal-adaptive superpixel segmentation and graph component labeling. Our algorithm can discriminate between planar surfaces and curved surfaces based on the reliability of estimated local planar surface structure. Therefore we can apply plane fitting to truly planar distributed areas and utilize depth variance based joint bilateral upsampling to curved or bumpy areas. As a result, we can generate a smooth depth map while preserving curved surfaces. By using massively parallel computing capabilities of modern commodity GPUs, the method is able to maintain high frame rates. The remainder of this paper is structured as follows. In Section 2, we will discuss related works. After describing the overview and the details of our technique in Section 3. Section 4 will show the result of experiments and discuss them. Finally we will conclude the paper in Section 5.

Matsumoto K., de Sorbier F. and Saito H..
Plane Fitting and Depth Variance Based Upsampling for Noisy Depth Map from 3D-ToF Cameras in Real-time.
DOI: 10.5220/0005184801500157
In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2015), pages 150-157
ISBN: 978-989-758-077-2
Copyright © 2015 SCITEPRESS (Science and Technology Publications, Lda.)

2 RELATED WORKS

In order to upsample the depth data captured by a ToF depth camera, several approaches have been proposed which can be divided into two groups. The first one deals with the instability of depth data provided by the RGB-D camera by using several depth images for reducing variations over each pixel depth value (Camplani and Salgado, 2012) (Dolson et al., 2010). However, these methods can not cope with numerous movement of objects in captured scenes or require the camera to be stationary.

The second group applies upsampling methods on only one pair of depth and color images for interpolating depth data while reducing structural noise. Among these methods, Joint Bilateral Upsampling (Kopf et al., 2007) and the interpolation method based on the optimization of a Markov Random Field (Diebel and Thrun, 2005) are the most popular approaches. They exploit information from RGB images to improve the resolution of depth data under the assumption that depth discontinuities are often related to color changes in the corresponding regions in the color image. However the depth data captured around object boundaries is not reliable and heavily contaminated with noise.

(Chan et al., 2008) solved this problem by introducing a noise-aware bilateral filter, which blends the results of standard upsampling and joint bilateral filtering depending on the depth map's regional structure. The drawback of this method is it can sometimes smooth the fine details of depth maps. (Park et al., 2011) proposed a high quality depth map upsampling method. Since it extends nonlocal means filtering with an additional edge weighting scheme, it requires a lot of computational time.

(Matsuo and Aoki, 2013) presented a depth image interpolation method by estimating tangent planes based on superpixel segmentation. In this method, depth interpolation is achieved within each region by using Joint Bilateral Upsampling. (Soh et al., 2012) also use superpixel segmentation for detecting piecewise planar surfaces. In order to upsample the low-resolution depth data, they apply plane based interpolation and Markov Random Field based optimization to locally detected planar areas. These approaches can adapt the processing according to local object shapes based on the information form each segmented region.

Inspired from these approaches, we also use superpixel segmentation for detecting locally planar surfaces and exploit the structure of detected areas. Compared with other superpixel based methods, our method can relatively smooth depth map in real-time.

3 PROPOSED METHOD

Figure 1: Left: SoftKinetic DepthSense DS311. Center: captured color image. Right: captured depth image.

As Figure 1 shows, we use SoftKinetic DepthSense DS311 for our system, which can capture 640 $\times$ 480 color images and 160 $\times$ 120 depth maps at 25-60fps.

Before applying our method, we project each 3D data from depth map onto its corresponding color image by using rigid transformation obtained from camera calibration between color camera and depth sensor. In our experiment, we use the extrinsic parameters given from a DepthSense DS311. After this process, we can obtain RGB-D data in color image coordinate frame.

However, it is still low resolution and includes much noise and occluded depth data around the object boundaries due to slight differences depth camera and color camera positions. Therefore, we first apply depth variance based joint bilateral upsampling to the RGB-D data and generate highly smoothed and interpolated depth map. Next, we calculate the normal map by applying the method proposed by (Holzer et al., 2012). By using this normal map, we apply normal-adaptive superpixel segmentation for dividing the 3D depth map into clusters so that the 3D points in each cluster make up a planar structure. For merging clusters which are located on the same plane, graph component labeling is utilized to segment image by comparing the normals of each cluster. The plane equation of each cluster is computed from the normal and center point associated with the cluster. After that, we evaluate the reliability of each plane and discriminate between planar cluster and curved cluster and apply plane fitting and optimization to the depth map. As a result, our method can generate smooth depth maps which still contain complex shape information.

3.1 Depth Variance Based Joint Bilateral Upsampling

Joint Bilateral Upsampling(JBU) is a modification of the bilateral filter, an edge-preserving smoothing filter for intensity images. The smoothed depth value D_{f_p} at the pixel p is computed from its neighboring pixels

Ω as follows:

$$D_{f_p} = \frac{\sum_{q\in\Omega} g_s(p-q)g_c(C_p-C_q)g_d(D_p-D_q)D_q}{\sum_{q\in\Omega} g_s(p-q)g_c(C_p-C_q)g_d(D_p-D_q)} \quad (1)$$

where g_s, g_c, g_d are Gaussian functions controlled by the standard deviation parameters $\sigma_s C \sigma_c C \sigma_d$ respectively. $p-q$ represents the spatial distance, C_p-C_q is color similarity and D_p-D_q is the depth similarity. As this equation shows, JBU locally shapes the spatial smoothing kernel by multiplying it with a color similarity term and a range term, and thus the edges can be preserved while the non-edge regions are smoothed.

However, the depth map obtained from ToF depth camera includes so much noise around the object boundaries that JBU can suffer from the effects of the noise. In order to remove the noise, we first calculate the mean and standard deviation of specified depth value around each pixel and if the variance is over the threshold, the depth data is removed. After that, the standard deviation is modified according to the depth error's quadratic dependance of distance defined by (Anderson et al., 2005) as follows:

$$\sigma'_l = \frac{\cos(\theta)\sigma_l}{D_m^2} \quad (2)$$

where σ'_l, D_m and θ are the local standard deviation, the local mean and the angle of incidence of infrared light. Then, σ_c is adapted to better reduce the noise and preserve the edges as follows:

$$\sigma_c = \max\{\sigma_{c_0} + \lambda \cdot \sigma'_l, \sigma_{min}\} \quad (3)$$

where σ_{c_0} is a relatively high sigma of g_c, σ_{min} is the minimum value, and λ is a negative factor. This modification is based on (Chen et al., 2012). Figure 2 shows the depth map captured in the scene of Figure 1 and the depth maps upsampled by JBU and depth variance based JBU. Compared with the center image, the noise around the object boundaries is removed and the depth map is properly upsampled in right image. After applying this technique, the smoothed and upsampled depth map is projected into 3D coordinates using the intrinsic parameters of the color camera.

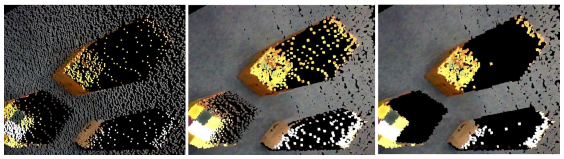

Figure 2: Left: input depth map. Center: JBU. Right: depth variance based JBU.

3.2 Normal Estimation

After utilizing joint bilateral upsampling, the normal estimation technique (Holzer et al., 2012) is applied to the 3D points for computing a normal map in real-time. This technique can generate a smooth normal map by employing an adaptive window size to analyze local surfaces. As this approach also uses integral images for reducing computational cost and can be implemented in GPU, we can calculate normal maps at over 50fps. However, this method can't estimate normals in the pixels around the object boundaries. Therefore, we interpolate the normal map by calculating the outer product of two close points around these invalid pixel vertices. The estimated normal map is visualized in Figure 3.

3.3 Normal Adaptive Superpixel Segmentation

(Weikersdorfer et al., 2012) proposed a novel over-segmentation technique, Depth-adaptive superpixels (DASP), for RGB-D images so that the 3D geometry surface is partitioned into uniformly distributed and equally sized planar patches. This clustering algorithm assigns points to superpixels and improves their centers using iterative k-means algorithms with a distance computed from not only color distance and spatial distance but also the depth value and normal vector. By using the color image, the depth map calculated in Section 3.1 and the normal map generated in Section 3.2, we modify the DASP to use gSLIC method by (Ren and Reid, 2011) in GPU.

The distance $dist_k(p_i)$ between cluster k and a point p_i is calculated as follows:

$$dist_k(p_i) = \frac{\sum_j w_j dist_{k_j}(p_i)}{\sum_j w_j} \quad (4)$$

with the subscript j consecutively representing the spatial(s), color(c), depth(d) and normal(n) terms. w_s, w_c, w_d and w_n are empirically defined weights of spatial, color, depth and normal distances, respectively represented as $dist_{k_s}(p_i)$, $dist_{k_c}(p_i)$, $dist_{k_d}(p_i)$ and $dist_{k_n}(p_i)$. Figure 3 illustrates the result of normal adaptive superpixels, where the scene is segmented as each region is homogeneous in terms of color, depth and normal vector. The normal adaptive superpixel segmentation gives for each cluster its $C_k(X_c, Y_c, Z_c)$ and its representative normal $n_k(a,b,c)$. As a result, each point $V_{k_p}(X_{k_p}, Y_{k_p}, Z_{k_p})$ located on a locally planar surface of a cluster k can be represented as follows

$$aX_{k_p} + bY_{k_p} + cZ_{k_p} = d_k \quad (5)$$

where d_k is the distance between the plane and the origin. Assuming that C_k is located on the planar surface, we can calculate d_k as follows.

$$d_k = aX_c + bY_c + cZ_c \quad (6)$$

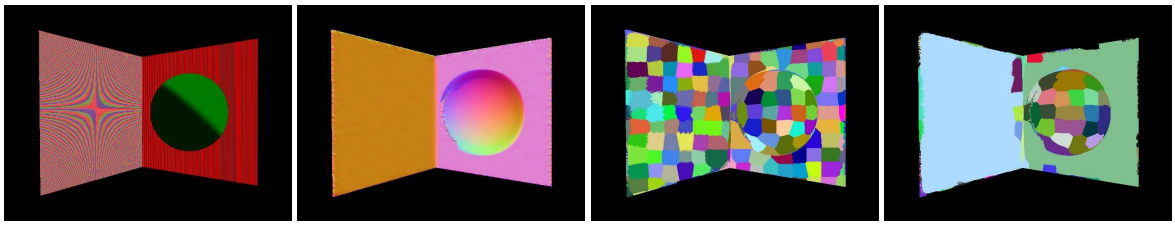

Figure 3: RGB image: normal image: normal adaptive superpixels: merging superpixels.

3.4 Merging Superpixels

Since the superpixel segmentation is over-segmentation proceduce, the post-processing is required to find global planar structures. (Weikersdorfer et al., 2012) also provides the spectral grapth theory, which extracts global shape information from local pixel similarity. However, it requires much computational time because it is not a parallel procedure and can not be implemented in GPU. Therefore, we apply graph component labeling with GPUs and CUDA proposed by (Hawick et al., 2010) to segmented images as illustrated in Algorithm 1. By considering each representative planar equation in given superpixel's clusters, the labeling process is carried out for merging clusters which are distributed on the same planar area.

As Figure 3 shows, we can obtain the global planar area while preserving small planar patches in real-time. Finally, the center and the representative normal vector of each region are computed again by taking the average of normals and center points of the superpixels in each region.

3.5 Plane Fitting and Optimization

By using equation (5), 3D coordinates $V_{k_p}(X_{k_p}, Y_{k_p}, Z_{k_p})$ on planar cluster k are computed from normalized image coordinates $u_n(x_n, y_n)$ as follows:

$$Z_{k_p} = \frac{d_k}{ax_n + by_n + c}, X_{k_p} = x_n Z_{k_p}, Y_{k_p} = y_n Z_{k_p} \quad (7)$$

By judging from the reliability of the plane model calculated during the previous step, we can detect which clusters are planar. The optimized point V_{o_p} is generated by using V_{f_p} computed from the depth variance based JBU in section 3.1 and the variance of normal vectors ψ_k obtained in section 3.4 as follows:

$$V_{o_p} = \begin{cases} V_{f_p} & (|V_{f_p} - V_{k_p}| > \gamma \frac{V_{k_p}^2}{\cos(\theta)} \quad \textbf{or} \quad \psi_k > \delta) \\ V_{k_p}\cos\psi_k + V_{f_p}(1.0 - \cos\psi_k) & (\textbf{otherwise}) \end{cases} \quad (8)$$

Algorithm 1: Superpixel Merging Algorithm.

function LabelEquivalenceHost($D, Size$)
  declare integer $L[Size], R[Size]$
  do in parallel initialize $L[0...Size-1]$ and $R[0...Size-1]$ such that $L[i] \leftarrow NASP[i]$ and $R[i] \leftarrow i$
  declare boolean m
  repeat
    do in parallel in all pixels call Scanning(D, L, R, m) and Labeling(D, L, R)
  until $m = false$
return
function Scanning(D, L, R, m)
  declare integer $id, label_1, label_2, q_{id}[9]$
  $id \leftarrow$pixel ithread ID)
  $label_1, label_2 \leftarrow L[id]$
  $q_{id} \leftarrow$neighbors of id
  for all $id_q \in q_{id}$ **do**
    declare float d_q, θ_q
    $d_{id_q} \leftarrow |d_{NASP[id]} - d_{NASP[id_q]}|$
    $\theta_{id_q} \leftarrow \arccos(n_{NASP[id]} \times n_{NASP[id_q]})$
    if $d_{id_q} < \alpha$ **and** $\theta_{id_q} < \beta$ **then**
      $min(label_2, L[id_q])$
    end if
  end for
  if $label_2 < label_1$ **then**
    $atomicMin(R[label_1], label_2)$
    $m \leftarrow true$
  end if
return
function Labeling(D, L, R)
  declare integer id, ref
  $id \leftarrow$pixel (thread ID)
  if $L[id] = id$ **then**
    $ref \leftarrow R[id]$
    repeat
      $ref \leftarrow R[ref]$
    until $ref = R[ref]$
    $R[ref] \leftarrow ref$
  end if
  $L[id] \leftarrow R[L[id]]$
return

where θ is the incident angle of the infrared light from a depth camera, γ and δ are the adaptively changing thresholds specifically chosen for a given scene for rejecting unreliable plane models. The huge error of plane fitting will be removed by setting the threshold γ. The threshold δ can prevent plane fitting from being applied to curved surfaces. Finally, we apply ordinary bilateral filter to V_{o_p} for smoothing the artifacts around boundaries.

4 EXPERIMENTS

We applied our method on two different scenes captured by SoftKinetic DepthSense DS311(color: 640×480, depth: 160×120) and compared our result(**PROPOSED**) with other related works, Joint Bilateral Filtering based Upsampling(**JBF**), Markov Random Field(**MRF**), **DISSS** proposed by (Matsuo and Aoki, 2013) and **SPSR** presented by (Soh et al., 2012) in terms of runtime and qualitative evaluation. For the quantitative evaluation, we generated the ground truth depth data with a scene rendered via OpenGL. The ground truth depth data was downsampled and added noise according to the noise model of ToF depth camera described in (Anderson et al., 2005). Then, we applied all methods to the noisy depth data and calculated root-mean-square-error(RMSE) and peak signal-to-noise ratio(PSNR) between ground truth and the results in order to compare the accuracy of all the methods. All processes are implemented on a PC with Intel Core i7-4770K, NVIDIA GeForce GTX 780, and 16.0GB of memory. We used OpenCV for trivial visualizations of color and depth images as well as data manipulations, and PointCloudLibrary for 3-dimensional visualization. All GPGPU implementations were done using CUDA version 5.0.

4.1 Qualitative Evaluation

Table 1 shows the parameters for each experiment. We adjust the parameters for the superpixel segmentation and merging superpixels so that we can divide the depth map into truly planar areas. As Figure 6 and 7 demonstrate, our technique can generate smooth and high resolution depth maps form low resolution and noisy data captured by ToF depth camera. **MRF** and **JBF** suffer from noisy data since these methods estimate a pixel depth value from its neighborhood. **DISSS** also applies joint bilateral upsampling in estimated homogeneous surface regions and can't reproduce smooth depth map. The upsampled depth map from **SPSR** is smoothed because it uses both plane fitting and markov random field to upsample the depth data based on local planar surface equation estimated by superpixel segmentation. However, as Figure 10 shows, fissures appear around the boundaries of each region in the upsampled depth map because the superpixel segmentation is processed locally. Figure 10 also shows that our method can obtain denoised depth map particularly in areas of planar surfaces while preserving the curved surfaces and the detail of objects with complex shapes (e.g. the depth map of stanford bunny). The reason is that our method can find global planar areas and adapt the upsampling method based on detected surface structures. Thanks to the preprocessing explained in section 3.1, we can remove the noise around the object boudaries as shown in Figure 9. In order to compare the runtime, all the methods are implemented with GPU and each runtime is shown in Figure 4. Compared with other superpixel based methods, our technique requires far less computational time as shown in Figure 4.

Method	Scene 1	Scene 2	Model 1	Model 2
JBF	3.0	3.0	3.0	3.0
MRF	4.0	4.0	4.0	4.0
SPSR	204.0	202.0	112.0	144.0
DISSS	75.0	71.0	87.0	111.0
PROPOSED	66.0	66.0	40.0	51.0

Figure 4: Runtime (msec).

4.2 Quantitative Evaluation

Based on the characterization of the flash ladar devices (Anderson et al., 2005), we presumed that the depth value variance $\sigma(p, d_{gt})$ at pixel p is discribed as follows:

$$\sigma(p, d_{gt}) = k\frac{d_{gt}^2}{\cos(\theta)} \tag{9}$$

where d_{gt} is the depth value acquired from ground truth depth data, θ is the incident angle of the infrared light from a depth camera and k is the noise coefficient. By using Box-Muller transform and equation 9, we added normally distributed random noise to the downsampled ground truth depth based on the probability distribution described as follows:

$$p(d|d_{gt}, p) \propto exp\left(-\frac{(d-d_{gt})^2}{\sigma(p, d_{gt})^2}\right) \tag{10}$$

In order to evaluate the effectiveness of all methods, we applied them to noisy downsampled depth data ($640 \times 480, 320 \times 240, 160 \times 120$) and calculated RMSE and PSNR. PSNR can be written as follows:

$$PSNR = 20\log_{10}\left(\frac{d_{max}}{RMSE}\right) \tag{11}$$

Table 1: Parameters for experiment.

Method	Parameters	**Scene 1**	**Scene 2**
Depth Variance Based JBU	σ_s, σ_c, σ_d, λ, σ_{min}	30, 50 , 100, -10, 15	70, 50 , 20, -10, 15
Superpixel Segmentation	w_s, w_c, w_d, w_n *iteration*, *clusters*	50, 50 , 50, 150 1, 300	50, 50 , 50, 150 1, 300
Merging Superpixels@	α, β	$220mm$, $\pi/8$	$75mm$, $\pi/12$
Optimization@	γ, δ	0.0001, $\pi/8$	0.0001, $\pi/8$

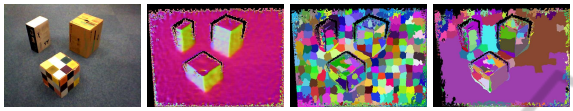

Figure 5: RGB: normals: superpixels: merging superpixels.

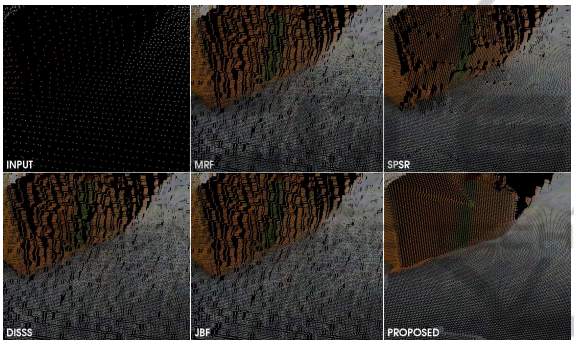

Figure 6: **Scene 1(a)**.

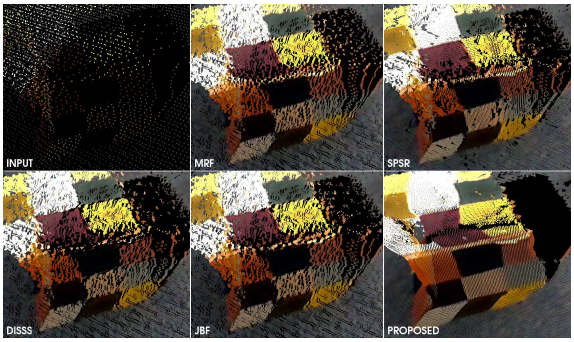

Figure 7: **Scene 1(b)**.

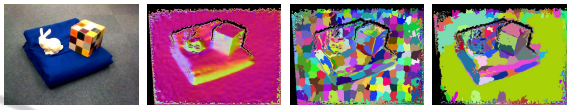

Figure 8: RGB: normals: superpixels: merging superpixels.

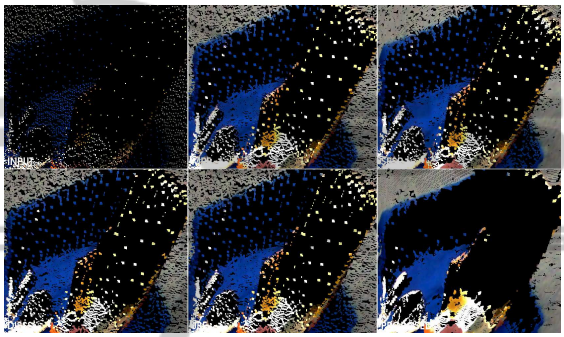

Figure 9: **Scene 2(a)**.

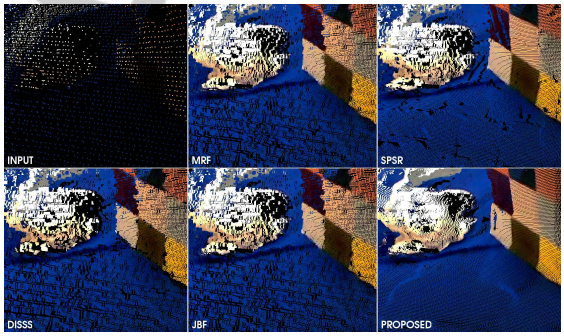

Figure 10: **Scene 2(b)**.

Model 1 consists of three planar surfaces and Figure 11 shows the result of the experiment with **Model 1**. Our technique can generate the closest depth map to the ground truth depth data because the method replaces the noisy depth map entirely with a plane fitted depth map. **Model 2** is composed of planar surfaces and curved surfaces. As Figure 13 illustrates, proposed method is the most accurate method and **SPSR** is the second of all the methods. Since **SPSR** applies the plane fitting and MRF optimization to local planar patches, the noise reduction is performed locally and that sometimes leads to fissure like discontinuities around the edges of each region as we discussed in 4.1. Moreover, the runtime of **SPSR** is the slowest of all methods because of the edge refinement of superpixel boundaries as shown in Table 4. Our method is slower than **JBF** and **MRF** but it can still maintain high frame rates because of parallel processing implemented in GPU. Our technique can reproduce relatively accurate depth map compared with other methods because it can distinguish planar regions and curved regions and apply the appropriate algorithms by combining planar fitting and depth variance based joint bilateral upsampling. To conclude, our technique clearly outperforms other methods, in terms of runtime, visual assessment and accuracy.

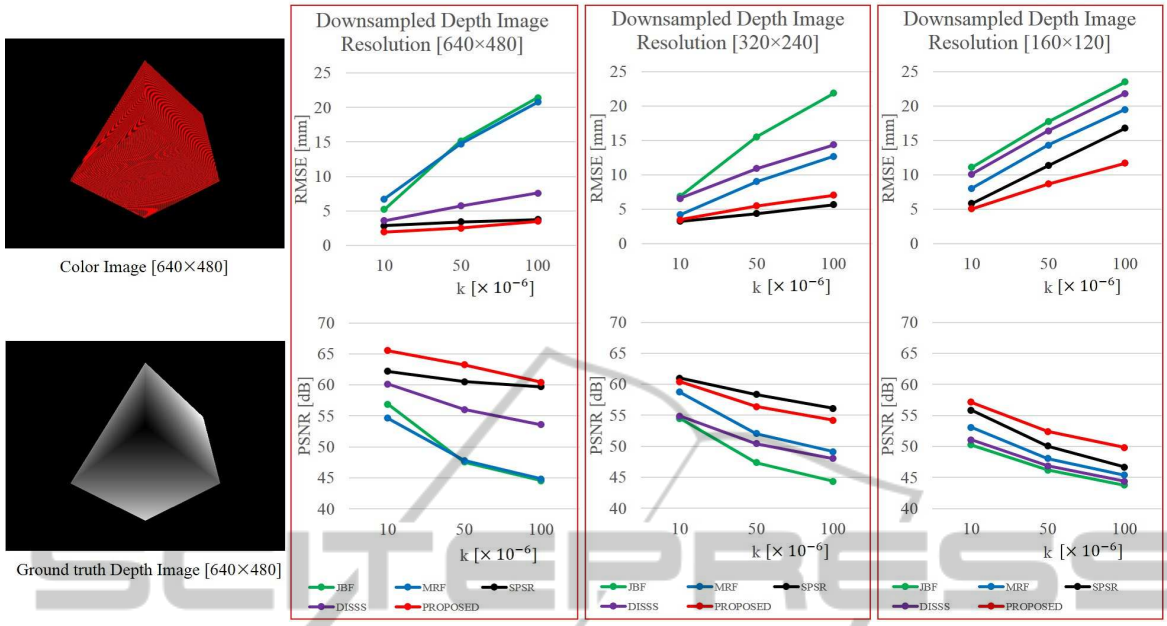

Figure 11: **Model 1** RMSE and PSNR ($d_{max} = 3622.93mm$).

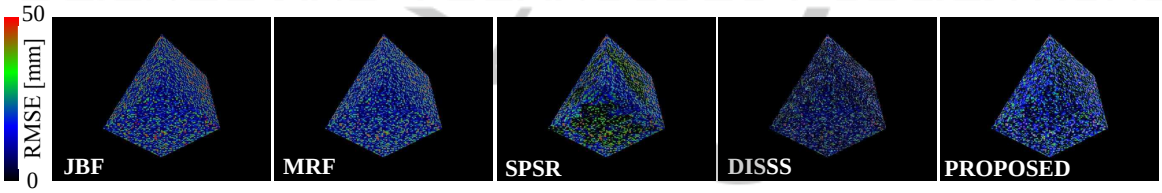

Figure 12: **Model 1** Visualization of RMSE (Input depth isize[160×120], k=50×10^{-6}).

5 CONCLUSIONS

In this work, we proposed a depth image upsampling and denoising algorithm, which has a low resolution depth image from ToF depth camera and a high resolution color image as its inputs. In order to detect planar structures,we combined normal adaptive superpixels and graph component labeling by simultaneously using color image, depth data and normal map. As our method can properly apply plane fitting and depth variance based joint bilateral filter according to the local points structure, it can generate smoothed depth map retaining the shape of curved surfaces.

Our experimental results show that this technique can upsample depth images more accurately than previous methods, particularly when applied to a scene with large planar areas. Since the algorithm is parallelizable, our framework can achieve real-time frame rates thanks to GPGPU acceleration via CUDA architecture, which becomes crucial when such a method is used in computationally expensive applications, such as 3D reconstruction and SLAM.

ACKNOWLEDGEMENTS

This work is partially supported by National Institute of Information and Communications Technology (NICT), Japan.

REFERENCES

Anderson, D., Herman, H., and Kelly, A. (2005). Experimental characterization of commercial flash ladar devices. In *International Conference of Sensing and Technology*, volume 2.

Camplani, M. and Salgado, L. (2012). Adaptive spatio-temporal filter for low-cost camera depth maps. In *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*, pages 33–36. IEEE.

Chan, D., Buisman, H., Theobalt, C., Thrun, S., et al. (2008). A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*.

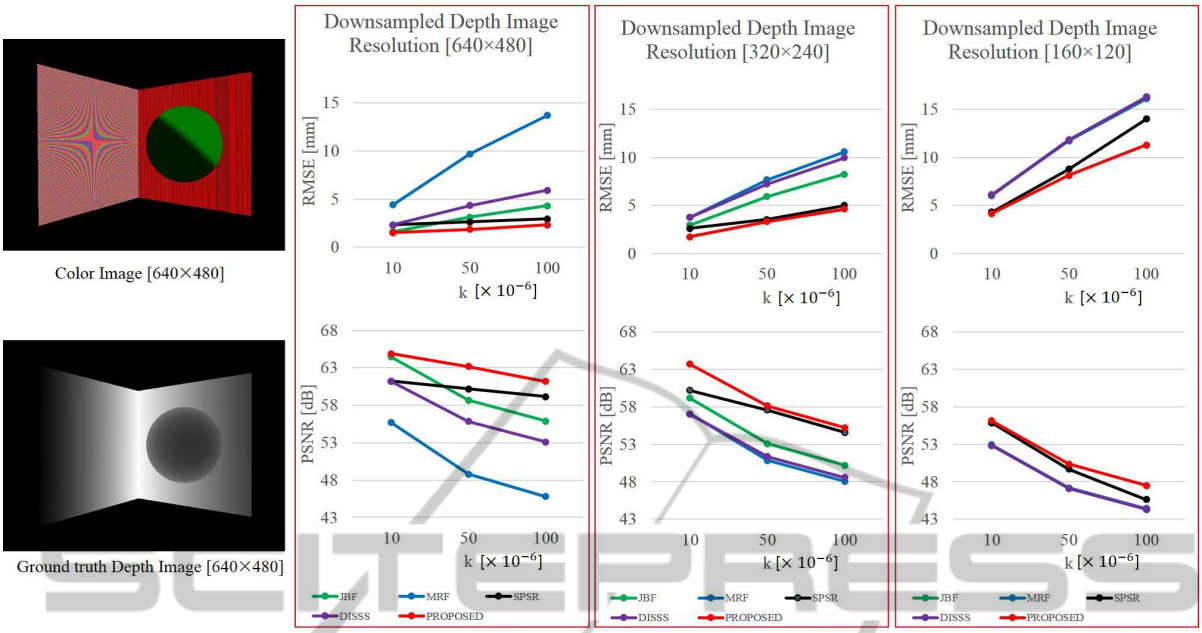

Figure 13: **Model 2** RMSE and PSNR ($d_{max} = 2678.52mm$).

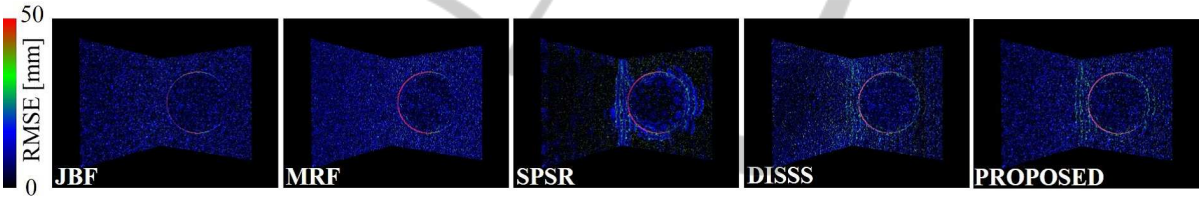

Figure 14: **Model 2** Visualization of RMSE (Input depth size [320 × 240], k=50 $\times 10^{-6}$).

Chen, L., Lin, H., and Li, S. (2012). Depth image enhancement for kinect using region growing and bilateral filter. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3070–3073. IEEE.

Diebel, J. and Thrun, S. (2005). An application of markov random fields to range sensing. In *Advances in neural information processing systems*, pages 291–298.

Dolson, J., Baek, J., Plagemann, C., and Thrun, S. (2010). Upsampling range data in dynamic environments. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1141–1148. IEEE.

Hawick, K. A., Leist, A., and Playne, D. P. (2010). Parallel graph component labelling with gpus and cuda. *Parallel Computing*, 36(12):655–678.

Holzer, S., Rusu, R. B., Dixon, M., Gedikli, S., and Navab, N. (2012). Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689. IEEE.

Kopf, J., Cohen, M. F., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 96. ACM.

Matsuo, K. and Aoki, Y. (2013). Depth interpolation via smooth surface segmentation using tangent planes based on the superpixels of a color image. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 29–36. IEEE.

Park, J., Kim, H., Tai, Y.-W., Brown, M. S., and Kweon, I. (2011). High quality depth map upsampling for 3d-tof cameras. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1623–1630. IEEE.

Ren, C. Y. and Reid, I. (2011). gslic: a real-time implementation of slic superpixel segmentation. *University of Oxford, Department of Engineering, Technical Report*.

Soh, Y., Sim, J.-Y., Kim, C.-S., and Lee, S.-U. (2012). Superpixel-based depth image super-resolution. In *IS&T/SPIE Electronic Imaging*, pages 82900D–82900D. International Society for Optics and Photonics.

Weikersdorfer, D., Gossow, D., and Beetz, M. (2012). Depth-adaptive superpixels. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2087–2090. IEEE.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS