

Depth Interpolation via Smooth Surface Segmentation using Tangent Planes Based on the Superpixels of a Color Image

Kiyoshi Matsuo and Yoshimitsu Aoki

Department of Integrated Design Engineering, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan, 223-8522

kmatsuo@aoki-medialab.org, aoki@elec.keio.ac.jp

Abstract

This paper describes a method for accurately interpolating a low-resolution depth image using a high-resolution color image. In our method, first, tangent planes on each superpixel are estimated from the sparse depth information and dense color information. Then, the neighboring superpixels that have smooth-connectable tangent planes are connected, and the image segmentation to smooth surfaces are achieved. Finally, the low-resolution depth image is interpolated using this smooth surface segmentation.

In experiments with images from the Middlebury stereo datasets, our method interpolates each image at a high rate, and achieves the lowest error when compared to existing techniques.

1. Introduction

There has been much interest in 3D measurement techniques, and recently the field has expanded to include applications where “the purpose is to recognize or model the environment” [15, 5, 13]. This expansion has meant that higher accuracy and higher resolution 3D measurements have been required in a wide range of environments, to analyze or model 3D data. However current 3D measurement devices, such as stereo vision, time of flight cameras, laser range finders, do not have the capability to satisfy these requirements. *Sensor Fusion* approaches are often used to solve this problem. By fusing the data captured by different devices, a greater performance is achieved than if they were used individually.

This paper proposes a fusion technique that takes a low-resolution depth image and a high-resolution color image, and produces accurate and extensive 3D measurements. Our main purpose is the accurate interpolation of a low-resolution depth image using a high-resolution color image. A similar technique was previously proposed [2, 14]. Another challenge in this field is accurate depth interpolation



Figure 1. **Left:** Color image. **Right:** Resulting segmentation. Different colors represent different segments. The smooth surface segmentation information is generated for the depth interpolation. This image is generated using x16 upsampling.

around boundaries between objects. To solve this problem, segmentation information was used in [10, 6]. However, previous work has used prior segmentation information, and performed depth interpolation using global optimization. Alternative methods have used a local filter [7, 4]. These filtering techniques can often produce over smooth results, especially if the depth image is at a low resolution. We propose a new depth interpolation method that simultaneously solves the segmentation problem. In our method, we achieve smooth surface segmentation by combining low-resolution depth information and high-resolution color information. We use superpixel segmentation [3] to detect local regions where shapes are smooth, and we estimate tangent planes of shapes on each superpixel using low-resolution depth information. Then, local tangent planes are connected by checking smooth connectivity, and smooth surface segmentation is performed without using any model of object shapes. Once the smooth surface segmentation is generated, and depth interpolation is achieved by using an upsampling filter on each of the smooth surfaces. Figure 1 and Figure 2 show the resulting image segmentation. We have prevented over-smoothing around object boundaries, because our depth interpolation is performed using smooth surface segmentation information. In addition, because the most part of these processes are processed locally,



Figure 2. **Left:** Color image. **Middle:** Resulting smooth surface segmentation. **Right:** Interpolation result. (magnification factor x16).

our method can be partially implemented as a parallel computing method (similar to other upsampling filters).

We have demonstrated the accuracy of our method by conducting experiments that use images of the Middlebury stereo datasets [12]. We have compared our results with the results of other interpolation methods. In our experiments, our method consistently achieved the lowest RMSE and MAE scores, and the interpolation rates were more than 75%.

1.1. Contribution

- We propose a method for estimating local tangent planes from low-resolution depth information using superpixel segmentation of high-resolution color information;
- We propose a segmentation method that uses the connectivity between local tangent planes without any assumptions on the model of object shapes;
- We introduce a novel depth interpolation procedure that first solves smooth surface segmentation by fusing low-resolution depth and high-resolution color information, and interpolates the depth image using resulting segmentation information;
- We demonstrate that performance of the proposed depth interpolation method and show that it is more accurate than previous methods.

2. Related work

In this section, we discuss previous research on depth interpolation techniques that combined low-resolution depth and high-resolution color images. Because our method is closely related to this type of technique, other interpolation techniques like *super resolution with multi-low-resolution depth images* are not discussed.

Diebel and Thrun proposed an interpolation method that was formulated as an optimization of a Markov random field (MRF) [2]. They inferred an interpolated high-resolution depth image by globally optimizing an MRF that

they defined its data-term using a sparse depth image, and its smoothness-term between neighboring pixels using color differences. In their method, they considered the relationship between smoothness of color and depth. After this, some approaches were proposed that focused on the local difference of color between objects. For example, in [7, 14] the authors proposed a method to interpolate depth data as a weighted sum from a sparse depth image. They used tables of weights that used difference between pixels in terms of color and position. This type of filter is known as a joint bilateral upsampling (JBU) filter.

To further enhance the interpolation accuracy, not only local differences of color and position, but also image segmentation has been used [6, 10]. In these methods, depth data can be accurately estimated by considering a global consistency that is defined by object-regions. Our approach fundamentally supports this idea. Although these previous works use prior segmentation information for global optimization, in our method, depth interpolation is achieved using a local filter, via generating smooth surface segmentation. In [4], a multi-lateral upsampling filter, which used the consistency of objects, was proposed as a parallelizable interpolation method. Garcia *et al.* introduced a new factor to the JBU filter, which was determined using the gradient magnitude of a low-resolution depth image. This technique is called pixel weighted average strategy for depth sensor data fusion (PWAS). This interpolation method can consider the local differences of objects, and prevent the mixing of depth data between objects. However, the technique determines the object boundaries using only low-resolution information. Therefore, a lower resolution depth image results in reduced interpolation accuracy. This is not the case for our method. The accuracy is not reduced, because the approximate high-resolution shape information can be used by tangent planes on superpixels.

3. Proposed method

This section describes our depth interpolation method. In our method, we first estimate the smooth surface segmen-

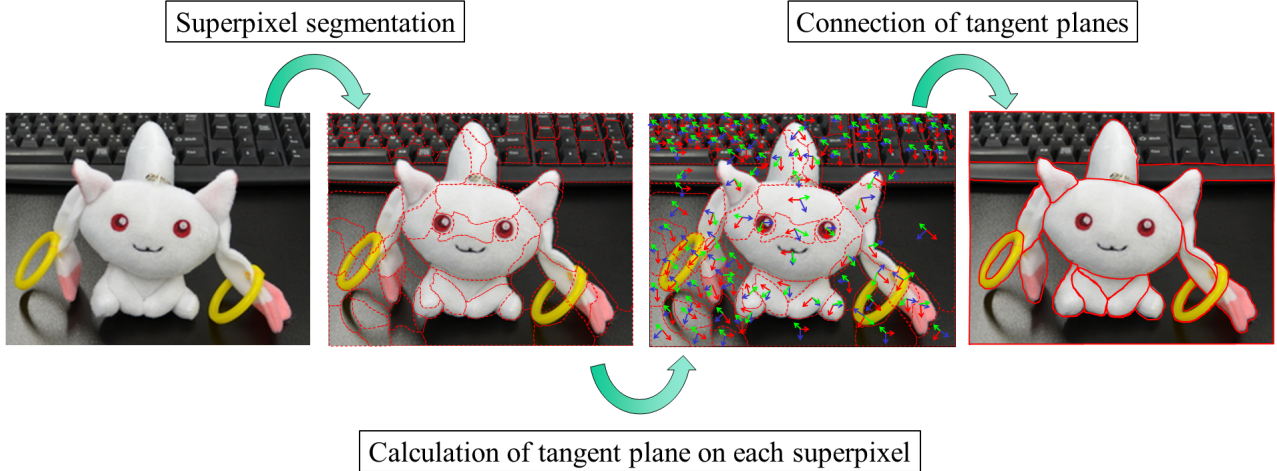


Figure 3. Overview of the processing flow of our smooth surface segmentation.

tation using low-resolution depth and high-resolution color information. Once the smooth surface segmentation is generated, depth interpolation is achieved using an upsampling filter on each of the smooth surfaces. Therefore, we first introduce the procedure of smooth surface segmentation.

3.1. Smooth surface segmentation

Smooth surface segmentation is achieved by combining superpixels which have smooth-connectable tangent planes. In this section, first, the superpixel technique is briefly introduced. Then, our calculation method of tangent plane on a superpixel is shown and a characteristic value of smooth connectivity between two tangent planes is defined. Finally, our smooth surface segmentation method is introduced using this characteristic value of smooth connectivity. Figure 3 shows the overview and we describe the detailed processing flow in section 3.1.4.

3.1.1 Superpixel

“Superpixel segmentation” is a segmentation technique of an image to small connected color-homogeneous regions. “Superpixel” is the one segment. This technique has been found useful pre-processing for many applications. We use superpixels as local regions where shapes are smooth. Some kind of superpixel segmentation methods have been proposed [9, 11, 3, 8, 1]. Our implementation uses the superpixel segmentation technique of [3].

3.1.2 Tangent plane on a superpixel

In Figure 4, outline of the calculation of a tangent plane on a superpixel is shown. First, depth information in a superpixel S is upsampled using the JBU filter [7] that is limited on the superpixel S . Although the JBU filter normally uses

all neighboring depth information, we use only the depth data of the same superpixel in order to prevent the mixing of depth data between different objects. We use the symbol $d(S)$ as the upsampled depth data, *i.e.* $d(S)$ is a point cloud that is defined from upsampled depth information on the superpixel S . A tangent plane with three axes are given by the eigen vectors of the covariance matrix of this 3D point cloud $d(S)$, with the smallest eigenvalue corresponding to the normal of the plane.

The detailed definition of the tangent plane is introduced here. We define an autocorrelation matrix $A(d(S))$ as follows.

$$A(d(S)) = \frac{1}{|d(S)|} \sum_{x \in d(S)} (x - C(S))(x - C(S))^T. \quad (1)$$

In this equation, $|d(S)|$ is the number of the points and $C(S)$ is the center of the point cloud $d(S)$. Because the matrix $A(d(S))$ is a three dimensional symmetric positive semidefinite matrix, it has three non-negative eigen values. We denote the smallest eigen value by $w_n(S)$, the middle eigen value by $w_2(S)$, and the largest eigen value by $w_1(S)$. $N(S)$, $T_2(S)$, and $T_1(S)$ are each corresponding eigen vector to $w_n(S)$, $w_2(S)$, and $w_1(S)$. We normalize these eigen vectors and use them as unit vectors. This calculation is popularly known as the calculation of a normal vector for a point cloud using PCA (Principal Component Analysis). In this context, we name $N(S)$ “normal vector” and $w_n(S)$ “normal width”. For descriptive purposes, $T_1(S)$, $T_2(S)$ are called as “tangent vectors” and $w_1(S)$, $w_2(S)$ are called as “widths of the tangent plane” in this paper. Although these values that are called “widths” are not exactly the widths of the point cloud (they are more near squared value of half-widths), we use this word for ease of explanation. We use the symbol $W(S)$ as the three dimensional vector that have

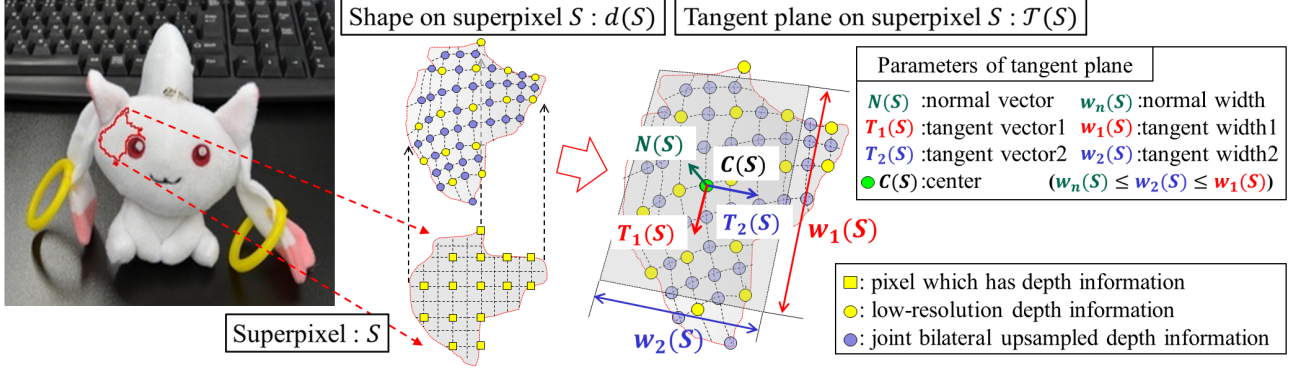


Figure 4. Tangent plane on a superpixel. Tangent plane on a superpixel is calculated using Principal Component Analysis of the shape on the superpixel that is defined by joint bilateral upsampled depth information.

these widths.

$$W(S) = (w_1(S), w_2(S), w_n(S))^T. \quad (2)$$

Finally, we define a tangent plane on a superpixel S using above symbols as follows.

$$\mathcal{T}(S) = (C(S), (N(S), T_2(S), T_1(S)), W(S)). \quad (3)$$

The conceptual image of a tangent plane on a superpixel is shown in Figure 4.

Next, we define the “dimension” of a tangent plane on a superpixel. This is the number of widths that are more than a width-threshold w_{th} .

$$\dim(\mathcal{T}(S), w_{th}) = \begin{cases} 0 & (w_1(S) \leq w_{th}). \\ 1 & (w_2(S) \leq w_{th} < w_1(S)). \\ 2 & (w_n(S) \leq w_{th} < w_2(S)). \\ 3 & (w_{th} < w_n(S)). \end{cases} \quad (4)$$

If the dimension $\dim(\mathcal{T}(S), w_{th})$ is three, the point cloud $d(S)$ is three-dimensional-spatially spreading, if it is two, the point cloud $d(S)$ is planarly spreading, and if it is one, the point cloud $d(S)$ is linearly spreading. The width-threshold w_{th} defines the amplitude of the spread.

The shape of a superpixel in the image-plane may make the dimension of the tangent plane less than two. In this case, in order to enhance the reliability of the tangent plane, we recalculate the tangent plane using the following extended shape on a superpixel. The extended shape on a thin superpixel S is the shape on the extended region \tilde{S} that contains the superpixel S and the pixels neighboring the boundary of S . The conceptual image is shown in Figure 5. For superpixels that have one or zero dimensional tangent planes, we use the tangent planes that are defined from these extended shapes.

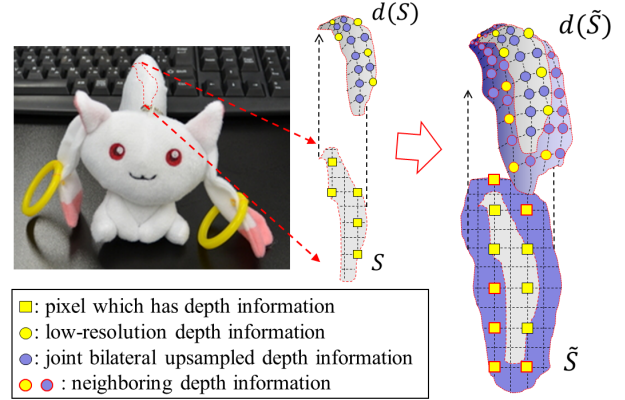


Figure 5. Extension of the shape for a thin superpixel.

3.1.3 Smooth connectivity of tangent planes

We define a characteristic value of smooth connectivity between two tangent planes $\mathcal{T}(S_1), \mathcal{T}(S_2)$ on superpixels S_1 and S_2 . When generating smooth surface segmentation, we use this characteristic value to evaluate smoothness of the shape on the boundary between the two superpixels. The

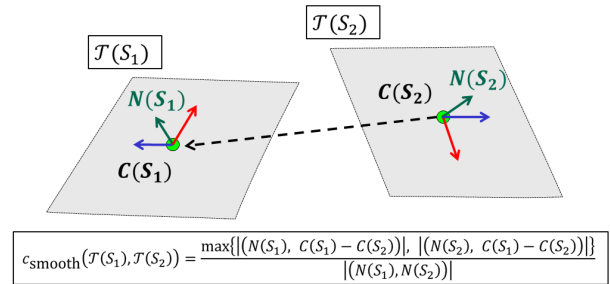


Figure 6. Characteristic value of smooth connectivity between two tangent planes.

characteristic value of smooth connectivity is defined as

$$\frac{\max\{|(N(S_1), C(S_1) - C(S_2))|, |(N(S_2), C(S_1) - C(S_2))|\}}{|(N(S_1), N(S_2))|}. \quad (5)$$

Here the notation (X, Y) describes the inner product of vectors X and Y , and $|(X, Y)|$ describes the absolute value. We use the symbol $c_{\text{smooth}}(\mathcal{T}(S_1), \mathcal{T}(S_2))$ for this characteristic value of smooth connectivity. When the directions of the normal vectors are similar and the difference vector between the centers is nearly orthogonal to the normal vectors, $c_{\text{smooth}}(\mathcal{T}(S_1), \mathcal{T}(S_2))$ becomes small. Therefore, if $c_{\text{smooth}}(\mathcal{T}(S_1), \mathcal{T}(S_2))$ is small, we consider that the shapes on the superpixels S_1, S_2 are smoothly connectable.

3.1.4 Smooth surface segmentation

In this section, we introduce the detailed processing flow of our smooth surface segmentation method. Figure 7 shows the detailed processing flow.

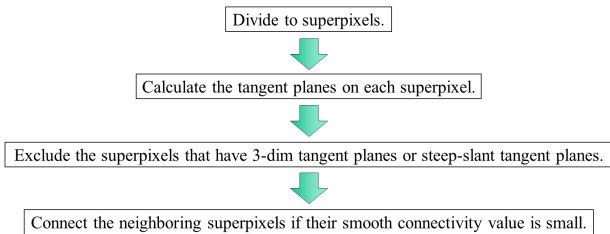


Figure 7. Processing flow of smooth surface segmentation.

We first get over-segmentation of an image into superpixels, and we calculate the tangent planes of each superpixel. Then, we exclude superpixels that have three dimensional tangent planes or steep-slant tangent planes. The dimension of a tangent plane is defined in the equation 4. When the angle between the camera direction vector and the normal vector of the tangent plane is larger than the threshold θ_{th} , we describe it as “the tangent plane is steep-slant”. On the superpixels that have three dimensional tangent planes, the shapes are not smooth surfaces because the point clouds that are defined by depth values on them are spreading three-dimensional-spatially. And on the superpixels that have steep-slant tangent planes, it is difficult to evaluate that they are real smooth surface or not. In order to enhance the accuracy of depth interpolation via smooth surface segmentation, we do not connect such superpixels. In remaining superpixels, if superpixels are neighboring and the characteristic value of smooth connectivity between their tangent planes is smaller than a threshold $c_{\text{smooth}}^{\text{th}}$, the superpixels are united. Therefore, smooth surface segmentation is achieved by combining superpixels that have smooth-connectable tangent planes.

3.2. Depth interpolation with smooth surface segmentation

In our method, depth interpolation is achieved using the previous described smooth surface segmentation. We first count the number of the low-resolution depth data on each segment. Depth interpolation is achieved on the segment using the JBU filter [7] that is limited there if the number of low-resolution depth data is larger than the threshold n_{th} . We interpolate a depth image using the depth data of the same surface segment in order to prevent the mixing of depth data between other surfaces. On each segment where there are few source depth data, we do not interpolate depth data.

4. Experimental results

Images from the Middlebury stereo datasets [12] were used to evaluate our new method. The Middlebury stereo datasets provide color images together with ground truth disparity images of some scenes. The focal length and the length of baseline of the stereo camera that captured these images are given. We have used the image center as the center of the camera because its coordinate is not given. In this setting, we have treated the disparity information and depth information equally. We have used the disparity maps of the datasets as the ground truth, and used them to generate low-resolution images by downsampling with rates of 2, 4, 8, and 16. We used the high-resolution color images directly. We have used scenes *Art*, *Books*, *Dolls*, and *Moebius*. The size of each image is the smallest size 463x370 pixels, because it is difficult to interpolate an image which locally contains varied changes of color and shape. We have used a set of parameters on our experiments. The parameters are shown in Table 1. In the supplemental materials,

| w_{th} | θ_{th} | $c_{\text{smooth}}^{\text{th}}$ | n_{th} |
|----------------------|----------------------|---------------------------------|-----------------|
| 10.0 mm ² | 70.0° | 20.0 mm | 5 |

Table 1. Parameters in our experiments.

experimental results on different parameters are shown.

We have compared our results to PWAS [4] and MRF [2] methods. We have measured the accuracy of the interpolated depth data using the MAE and RMSE scores. We have calculated these errors using the disparity information of the interpolated region. The MAEs and RMSEs of an interpolated depth image d are calculated using the corresponding disparity image disp as follows.

$$\text{MAE} = \sum_{p \text{ has a depth value}} \frac{|\text{disp}(p) - \text{disp}_{\text{true}}(p)|}{N}. \quad (6)$$

$$\text{RMSE} = \sqrt{\sum_{p \text{ has a depth value}} \frac{|\text{disp}(p) - \text{disp}_{\text{true}}(p)|^2}{N}}. \quad (7)$$

In these equations, $\text{disp}_{\text{true}}$ is the ground truth disparity image.

In this paper, we are proposing a partial parallelizable accurate interpolation procedure and not its parallel implementation. Therefore, we have not optimized the implementation of each method. However, we have implemented the proposed method using OpenMP so that we can approximately evaluate its processing time. We ran our experiments to estimate processing time on an Intel Core i7-2600 CPU (3.40GHz) with 16GB RAM.

Figure 8 shows the color images, MAEs, and RMSEs, compared to the ground truth using different magnification factors. The interpolation rates and processing time of our method are also shown. In these graphs, the errors of our method are shown as solid red lines, the errors of the PWAS method are shown as broken green lines, and the errors of the MRF method are shown as broken blue lines. Our interpolation method outperforms PWAS and MRF in terms of accuracy. The differences are large, especially when using large scale upsampling. Additionally, the changes of the errors with respect to the upsampling-rate are smaller than them. These results show that the segmentation information was beneficial for the interpolation of the depth image, and that our method used the information more efficiently than the PWAS technique. The interpolation rates of our method were more than 75%, and its running times were approximately 25 seconds with this implementation.

Figure 1, Figure 2, and Figure 9 show the results of the smooth surface segmentation. There are several regions that are under-segmented. However, the accuracy of the interpolated depth data shows that the resulting segmentation is suitable for upsampling using the JBU filter [7]. The segmentation result is dependent on to the threshold of the smooth connectivity. More segmentation results that are generated with smaller threshold of the smooth connectivity are available in the supplemental materials.

5. Discussion

5.1. Parallelizability

Our interpolation method can be implemented by efficient parallel processing because our method mainly consists of the local processing. The calculations of tangent planes on each superpixel and the calculations of depth values with JBU filter are parallelizable. However, superpixel segmentation and the connection processing of tangent planes are not parallelizable, because they are global processing.

5.2. Limitations

Our interpolation method has regions where depth interpolation is not performed. On superpixels where shapes are not smooth surfaces, our interpolation method does not es-

timate the depth value. Superpixels that have three dimensional tangent planes or steep-slant tangent planes, are that superpixels. In such regions, there are boundaries of objects or complex patterns of color. It is difficult to accurately interpolate depth values in those regions, because the relationship between depth and color is weak. Our method does not interpolate these regions in order to retain a high accuracy in the interpolated depth image. Other techniques can not detect these regions, so this capability is a beneficial characteristic of our interpolation method.

6. Conclusion

This paper describes an accurate depth interpolation method for a low-resolution depth image that uses a high-resolution color image. Our interpolation method is based on a smooth surface segmentation that is generated by low-resolution depth and high-resolution color information. We used a superpixel segmentation of the high-resolution color information to detect local smooth regions. We then estimated tangent planes of shapes on each superpixel using low-resolution depth information. The superpixels that have smooth-connectable tangent planes were connected to achieve the smooth surface image segmentation. Finally, the low-resolution depth image was interpolated using this smooth surface segmentation.

Our experimental results show that this method can interpolate depth images more accurately than previous methods, particularly when using large scale upsampling. This method uses smooth surface segmentation to accurately interpolate an image. Segments in the resulting segmentation do not necessarily relate to objects in an image. However, as the accuracy is better than existing upsampling methods using multi-lateral filters, the resulting segmentation is suitable for upsampling using a JBU filter on each segment.

7. Future work

In section 5.2, we mentioned that our interpolation method does not interpolate depth values on superpixels where it is difficult to estimate the accurate tangent plane. This is major challenge of our interpolation method for the application to the noisy data of real-world. In such superpixels, because there is not enough accurate shape information, we have to use the information of surrounding superpixels in order to estimate the accurate tangent planes. We would like to improve our method by focusing on this point.

References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, nov. 2012.

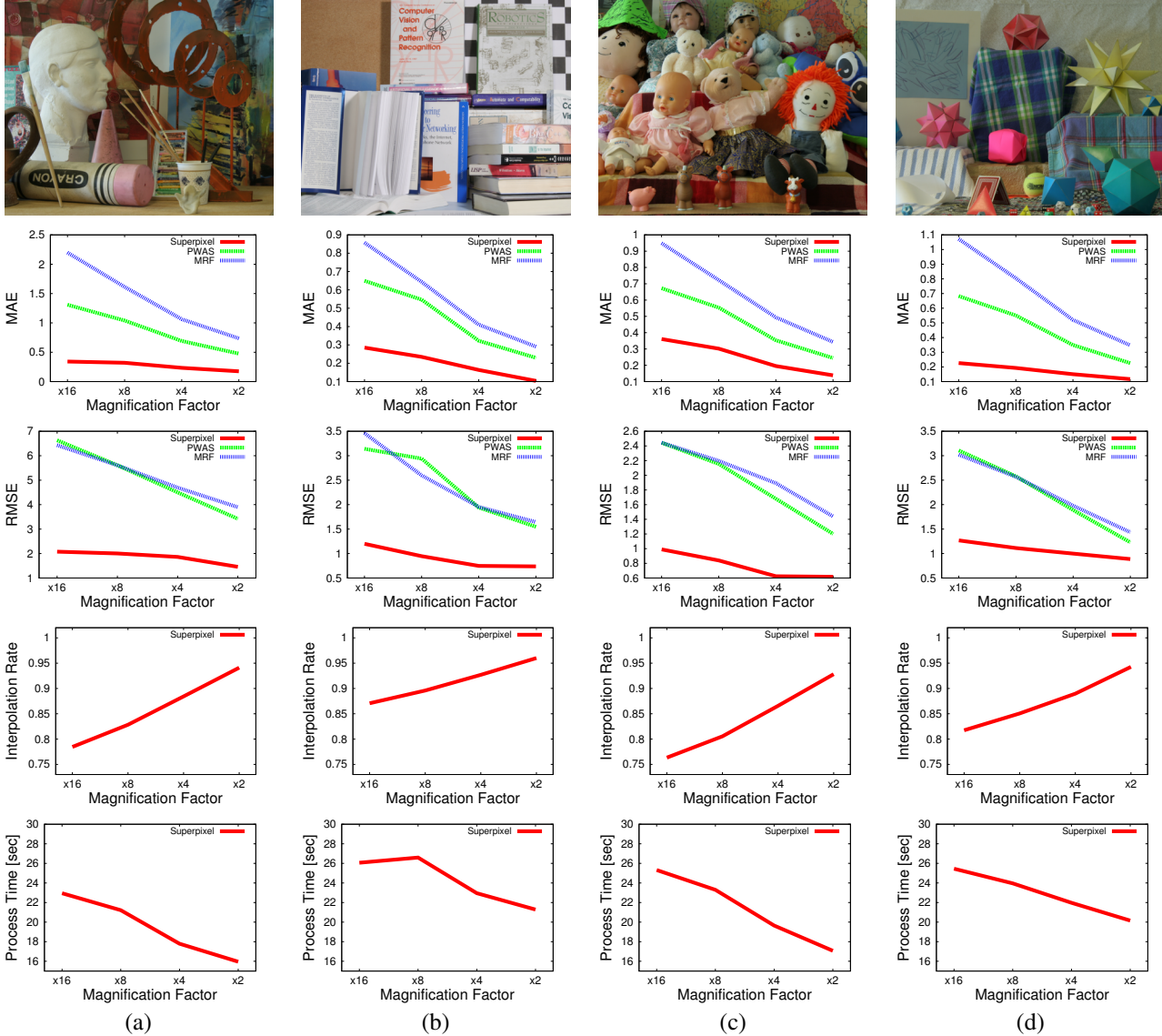


Figure 8. Color images of the scenes, MAEs, RMSEs, interpolation rates, and processing times for (a)Art, (b)Books, (c)Dolls, and (d)Moebius. The errors, interpolation rates, and processing times of our method are shown as solid red lines, PWAS's [4] are shown as broken green lines, and MRF's [2] are shown as broken blue lines.

[2] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 291–298. MIT Press, Cambridge, MA, 2006. 1, 2, 5, 7

[3] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 1, 3

[4] F. Garcia, B. Mirbach, B. Ottersten, F. Grandier, and A. Cuesta. Pixel weighted average strategy for depth sensor data fusion. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2805–2808, sept. 2010. 1, 2, 5, 7

[5] S. Hinterstoisser, S. Holzer, C. Cagniard, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865, nov. 2011. 1

[6] Z. Jia, Y.-J. Chang, T.-H. Lin, and T. Chen. Dense interpolation of 3d points based on surface and color. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 869–872, sept. 2011. 1, 2

[7] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph.*, 26(3), July 2007. 1, 2, 3, 5, 6

[8] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dick-

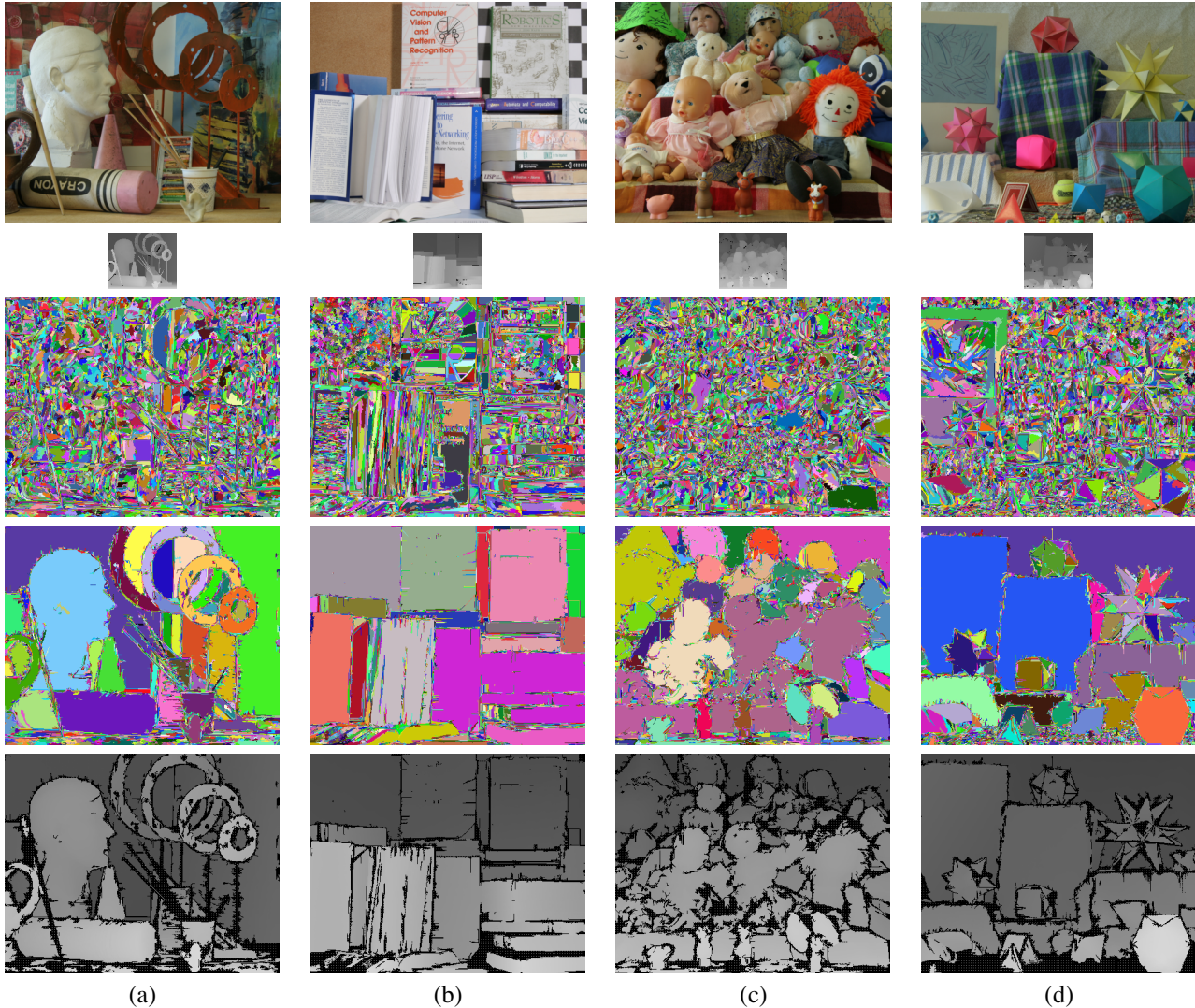


Figure 9. Color images of the scenes, 1/16 disparity images, superpixel segmentations, smooth surface segmentations, and interpolated disparity images for (a)Art, (b)Books, (c)Dolls, and (d)Moebius. These images are generated using x16 upsampling.

- inson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297, 2009. 3
- [9] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, Aug. 2004. 3
- [10] J. Park, H. Kim, Y.-W. Tai, M. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1623–1630, nov. 2011. 1, 2
- [11] X. Ren and J. Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17 vol.1, 2003. 3
- [12] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, june 2007. 2, 5
- [13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgb-d images. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 746–760. Springer Berlin / Heidelberg, 2012. 1
- [14] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, june 2007. 1, 2
- [15] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 708–721. Springer Berlin / Heidelberg, 2010. 1