FILTER BASED ALPHA MATTING FOR DEPTH IMAGE BASED RENDERING

Naoki Kodera, Norishige Fukushima and Yutaka Ishibashi

Graduate School of Engineering, Nagoya Institute of Technology

ABSTRACT

In this paper, we propose a free viewpoint image rendering method combined with filter based alpha matting for improving the image quality of image boundaries. When we synthesize a free viewpoint image, blur around object boundaries in an input image spills foreground/background color in the synthesized image. To generate smooth boundaries, alpha matting is a solution. In our method based on filtering, we make a boundary map from input images and depth maps, and then feather the map by using guided filter. In addition, we extend view synthesis method to deal the alpha channel. Experiment results show that the proposed method synthesizes 0.4 dB higher quality images than the conventional method without the matting. Also the proposed method synthesizes 0.2 dB higher quality images than the conventional method of robust matting. In addition, the computational cost of the proposed method is 100x faster than the conventional matting.

Index Terms— view synthesis, alpha matting, depth image based rendering, depth map refinement.

1. INTRODUCTION

Recently, free viewpoint image rendering has attracted increasing attention. Free viewpoint images are expected as new 3D media. Free viewpoint image rendering can synthesize views on arbitrary viewpoints by users' inputs. One of the rendering methods is depth image based rendering (DIBR) [1, 2]. The DIBR synthesizes free viewpoint images from input images and depth maps. In the DIBR processing, blurred regions on object boundaries spills foreground/background color into the other sides in the synthesized view. Therefore, those boundary regions are degraded.

Mori *et al.* [1] define the spilled regions by hard thresholding, and then erase the regions. Next, the erased regions are interpolated by using the other side view. Zinger *et al.* [2] propose an extended method of [1] that adaptively erodes the spilled regions, and then eroded areas are interpolated by inpainting [3] with depth information. However, the erosion process absorbs the whole information at the pixel in the area, even if the pixel has useful information, such as fractional foreground and background colors. Also, these methods cannot reconstruct blur on the object boundaries.

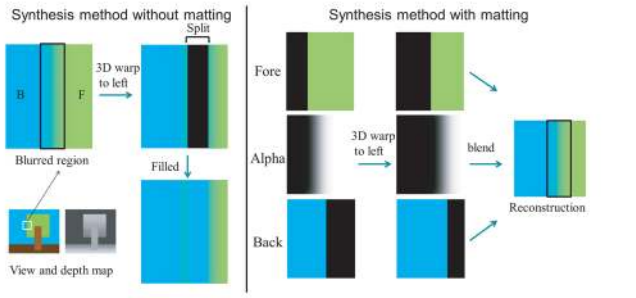


Fig. 1. Problem of blurred region warping, and the solution of alpha matting.

Alpha matting methods [4] for view synthesis [5, 6] can reconstruct the blur. The matting can split the blurred regions into foreground and background regions. Figure 1 (left side) shows the blurred region problem of the synthesis method without matting. After 3D warping, the blurred regions are split, and an occlusion hole is filled with the background color. As a result, the synthesized image has remained a blurred region in the background region. In the synthesis method with matting (Fig. 1 (right side)), we can decompose the input image into a foreground image, a background one and an alpha map at first. Next, each image is warped, and then is blended by alpha blending. The synthesized image does not have the remained blurred region in the background region, and blur on the edge is reconstructed.

In the previous researches of matting based view synthesis methods [5, 6], these methods require valid trimaps, which show foreground/background/unknown regions. Also, the matting optimization does not have real-time performance. Zitnick *et al.* [5] propose a method that computes matting information from all depth discontinuities, and use Bayesian matting [7]. Chan *et al.* [6] propose a method that decomposes object boundaries by segmenting and tracking an object. To obtain the valid trimaps, these methods use a single object moving data, and segmentation from temporal information. These methods are accurate for such regions, but require much computational cost for accurate alpha matting. In addition, there is no objective evaluation of synthesized image signals, but there are only subjective assessments.

Therefore, we propose a fast matting for view synthesis based on filtering. The methods can do the alpha matting for rugged objects. In our method, we detect sparse boundaries

This work was partly supported by SCOPE 122106001.

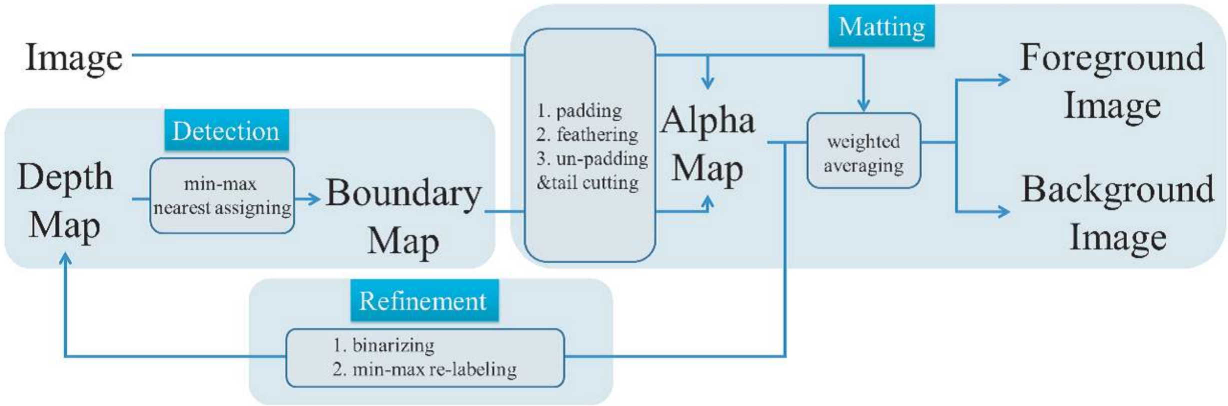


Fig. 2. Flowchart of filter based alpha matting.

from a depth map, and then solve alpha matting by using guided filter [8]. Also, we validate the effectiveness of our method by objective assessments with a lot of datasets.

Organization of this paper is as follows. In Section 2, we propose a view synthesis method with a filter based boundary matting. Section 3 demonstrates the view synthesis performance of the proposed method. Finally, we conclude this paper in Section 4.

2. PROPOSED METHOD

In our rendering, we input left and right (L-R) images I_L, I_R and associate L-R depth maps D_L, D_R. The main difference from the conventional approaches [1, 2] is that we perform matting around boundary regions in the input images. In our method, we split these maps into L-R background images $IB_{L,R}$, L-R foreground images $IF_{L,R}$ and L-R alpha maps $A_{L,R}$ by using a filter based alpha matting. Then, these split maps are warped to a virtual view on a target viewpoint.

The filter based alpha matting contains three processes, which are boundary detection, boundary refinement and boundary matting. Figure 2 shows a flowchart of the method. We detect boundaries, and then refine the boundaries iteratively. Finally we obtain a foreground image, a background image, and an alpha map with our matting. Details of each step of our method are described as follows.

2.1. Boundary detection, refinement, and matting

In our method, we decompose the input image into a boundary image and a non boundary image. The only boundary image is split into a foreground, a background and an alpha map by using alpha matting [4]. The alpha matting can split a natural image I into a foreground image IF, a background image IB and an alpha map A, such that these maps meet the following formula:

$$I \simeq A \cdot IF + (1.0 - A) \cdot IB. \qquad (1)$$

In our method, we do not solve this problem exactly, but solve it simply using the guided filter [8]. The solution of a boundary detection, refinement and matting are as follows.

2.1.1. Boundary detection

For boundary matting, we detect object boundaries in an input image by using an associated depth map. We assume that object boundaries have large differences of depth values in boundary regions. The boundary detection process is as follows. First, we apply an input depth map to the max/min filter. Then, we compute the differences between the input depth map and the resulting maps of the max/min filter.

$$dif(p) = D^{max}(p) - D^{min}(p), \qquad (2)$$

$$dif^{max}(p) = D^{max}(p) - D(p), \qquad (3)$$

$$dif^{min}(p) = D(p) - D^{min}(p), \qquad (4)$$

where p is a current pixel. D is an input depth map. $D^{min/max}$ is a man/min filtered depth map, respectively.

Finally, we divide an input image into a boundary region and a non boundary one with a threshold value. In addition, we divide the boundary region into a foreground region and a background region by dif^{max} and dif^{min}.

$$B = \begin{cases} \text{non boundary} & dif \leq th \\ \text{foreground} & dif > th, dif^{max} \leq dif^{min} \\ \text{background} & dif > th, dif^{max} > dif^{min}, \end{cases} \qquad (5)$$

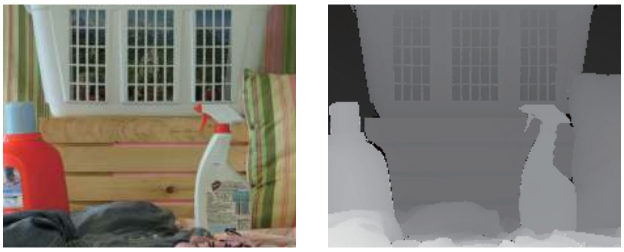
where th is a threshold and B is a boundary map. We set $th = 10$ in our experiment. Then, the pixels in the boundary map are assigned to three labels, non boundary (undefined value), foreground (255) and background (0). Note that, pixel values are in the range $\{0, ..., 255\}$ in this paper.

2.1.2. Boundary refinement

We use the guided filter [8] for boundary refinement. It is because that the object boundary exists in the blurred region where the foreground and background are mixed. The boundary refinement process is as follows.

$$T^{out}(i) = \sum_j w_{ij}(G)T(j), \qquad (6)$$

$$w_{ij}(G) = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} (1 + \frac{(G(i) - \mu(k))(G(j) - \mu(k))}{\sigma^2(k) + \epsilon}), \qquad (7)$$



(a) Input image (b) Input depth map

Fig. 3. Input image and depth map.

where i, j and k are pixel indexes, T^{out} is an output image, T is a filtering target image, G is a guidance image. μ is the mean image of guidance image G and σ^2 is a variance image of guidance on. ϵ is a parameter, which is 1.0 in our experiments, to control the smoothness. ω is the normalized factor of weight in Eq. (9). ω_k is the size of the filter kernel. In our experiment, the kernel size is 7×7.

We feather a boundary region (foreground/background) at boundary map using the guided filter.

$$B' = GuidedFilter(I, B), \tag{8}$$

where B is a boundary map and $GuidedFilter(guide, target)$ function of Eq. (6) is the guided filter. Note that before the guided filtering, we pad the boundary map to avoid convoluting the undefined values in the regions. In the process, the undefined non boundary regions which overlap the guided filter kernel are padded by foreground or the background label to have continuous values. For padding, we dilate or erode foreground/background regions toward undefined regions. After guided filtering, these regions are un-padded to recover undefined regions using the pre-padding mask.

The feathered boundary regions are binarized by a threshold value. In our method, the threshold is the middle value of unsigned byte value (128).

$$B^{ref}(p) = \begin{cases} 255 & B'(p) \geq 128 \\ 0 & \text{else.} \end{cases} \tag{9}$$

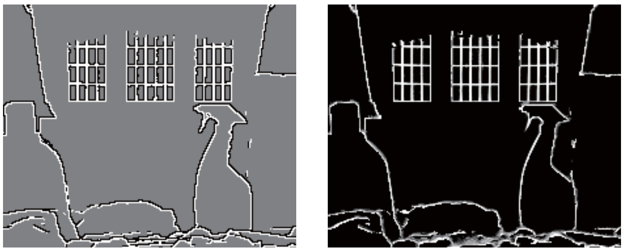
Then, we compare a feathered and binarized boundary map (B^{ref}) with non processed one (B), and then replace a depth value at reversing boundary map pixels with the max or min value of depth map (D^{max}, D^{min})

$$D^{ref}(p) = \begin{cases} D^{max}(p) & B(p) = 0, B^{ref}(p) = 255 \\ D^{min}(p) & B(p) = 255, B^{ref}(p) = 0 \\ D(p) & \text{else.} \end{cases} \tag{10}$$

We iterate the boundary detection in the previous subsection and the refinement process to localize the accurate boundary position. In this paper, we found that 3 times iteration is enough for our experiments.

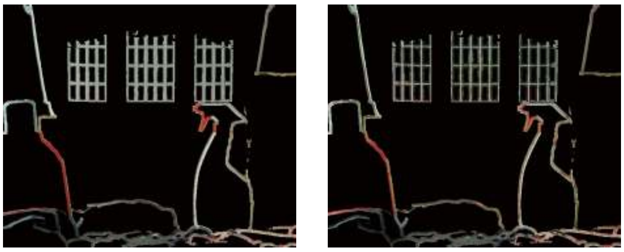
2.1.3. Boundary matting

After refining the boundaries, we generate foreground, background and alpha maps on the boundary regions. We assume that feathered values by the guided filter have almost equal to the values of the alpha map (see [8]).



(a) Boundary map (b) Alpha map

Fig. 4. Generated boundary map and alpha map. Non boundary regions are set to 128 in (a), and are set to 0 in (b).



(a) Foreground strip image (b) Background strip image

Fig. 5. Generated foreground and background strip images.

The boundary matting process is as follows. First, we generate an alpha map. The boundary map B^{ref} generated by the refined depth map (D^{ref}) are feathered by the guided filter.

$$B^{ref'} = GuidedFilter(I, B^{ref}). \tag{11}$$

Then, we cut the tail of the values near foreground (255) / background (0).

$$A(p) = \begin{cases} 255 & B^{ref'}(p) > 255 * c \\ 0 & B^{ref'}(p) < (1-c) * 255 \\ B^{ref'}(p) & \text{else,} \end{cases} \tag{12}$$

where A is an alpha map and c is a percentage value. We set $c = 0.85$ in our experiment.

Next, we generate foreground and background strip images by using a weighted box filter. Foreground and background images are calculated as follows;

$$S_L(p) = \frac{\sum_{s \in N} W_L(s) I(s)}{\sum_{s \in N} W_L(s)}, \tag{13}$$

where p is the current pixel, s is the support pixel, N is an aggregating set of support pixels, S_L is the strip image and L is a label set of foreground/background $L \in \{F, B\}$, I is an input image. W_L is foreground/background weight map. When we generate foreground, if $A(s)$ equal 255, $W_F(s)$ set to 1. Otherwise $W(s)$ set to 0. When we generate background, if $A(s)$ equal 0, $W_B(s)$ set to 1. Otherwise $W_{F,B}(s)$ set to 0.

Figure 3 shows the input image and the depth map, Fig. 4 shows the generated boundary map and alpha map, and Fig. 5 shows the generated foreground and background strip images.

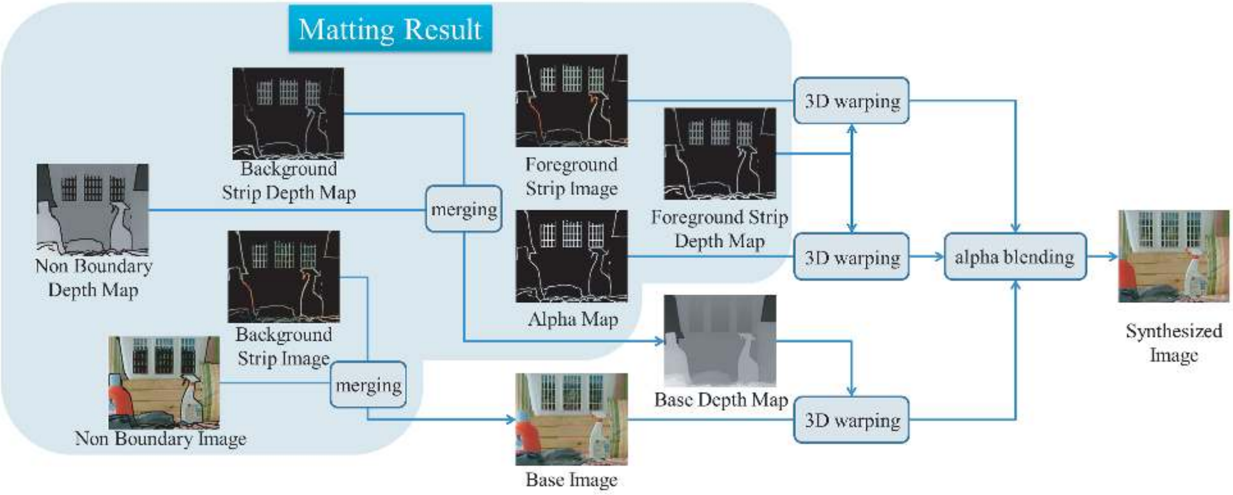


Fig. 6. Flowchart of matting based view synthesis.

2.2. Matting based view synthesis

After the L-R individual boundary matting process, we can obtain left and right foreground strip, background strip and alpha maps of boundary regions. Then we perform matting based view synthesis. Figure 6 shows the flow of the view synthesis process.

At first, we generate a base image, which is addition of the non boundary image and the background strip image. From these maps, we can synthesize three images, which are the foreground image, the base image, the alpha map, on a virtual viewpoint individually by using DIBR. Then, we can render a free viewpoint image by blending the synthesized the foreground and the base images with the alpha map.

After the boundary matting, the boundary regions have two colors, which are the foreground color and background color. However, the pixel in the input depth map in the region has only one depth value. Therefore, we need to generate additional depth values. In our method, we generate the foreground/background strip depth map using the max filter/min filter, and we merge the non boundary depth map and the background strip depth map into the base depth map.

As a next step, we can synthesize a free viewpoint image. We use X^F/X^B to denote a foreground/base of an image X, $X_{L,R}$ to denote a left or right image in a stereo image pair, and X^w to denote a warped image X.

We warp I^F by using the associated foreground depth map, and warp I^B by using the base depth map. The value of α in the alpha map is warped by the foreground depth map.

$$I^{Bw} = warp(D_{L,R}, I^B_{L,R}), \tag{14}$$

$$M^w_{L,R} = warp(D^F_{L,R}, M_{L,R}), M \in \{I^F, A\}, \tag{15}$$

where I = an input image, A = an alpha map, and D = a depth map. In $warp(x,y)$ function, x is a warping depth map, and y is a warping target image. Then, we blend the warped left and right foreground, base and alpha map, individually.

$$M^b = blend(M_{L,R}), M \in \{I^{Fw}, I^{Bw}, A^w\}, \tag{16}$$

where M^b are the blended foreground, base and alpha maps, and the $blend()$ function blends left and right signals. Finally, we can synthesize the free viewpoint image I_{syn} by pasting the synthesized foreground image F^b to the synthesized base one B^b with the synthesized alpha map A^b.

$$I_{syn} = A^b \cdot F^b + (1.0 - A^b) \cdot B^b. \tag{17}$$

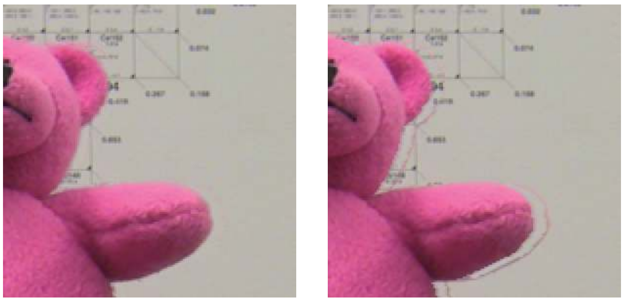
3. EXPERIMENTAL RESULTS

In the experiments, we compare the proposed method with non matting method [1], and then compare our method with matting method [9] later. In our experiments, we use the Middlebury's data sets [10]. We use left and right images as view synthesis anchors, and use the center image as just a reference image in PSNR evaluation. The input depth map is the ground truth provided by the data sets.

Table 1 shows PSNR of the virtual views from various input images. The proposed method has higher PSNR (0.40 dB on the average) than the non matting method.

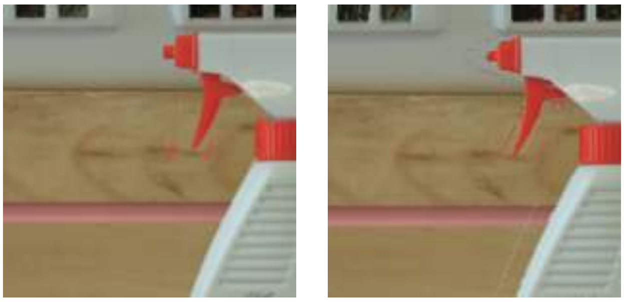
Figure 7 shows zoomed synthesized images at the “teddy” sequence of the proposed and the conventional. The rendering result of the proposed method has more smooth boundaries than the conventional method, and the blurred regions in the image of the conventional method is spilled into the foreground color and the background colors.

Figure 8 shows zoomed synthesized images at the “laundry” sequence of the proposed and conventional. We also can observe contour artifacts in the proposed images as shown in Fig 8. The proposed method can solve object boundaries well near the large object (a bottle), but the method damages boundaries of thinning object (a trigger of the bottle). This is because that the proposed method cannot accurately solve matting in the region which includes a slim foreground object.



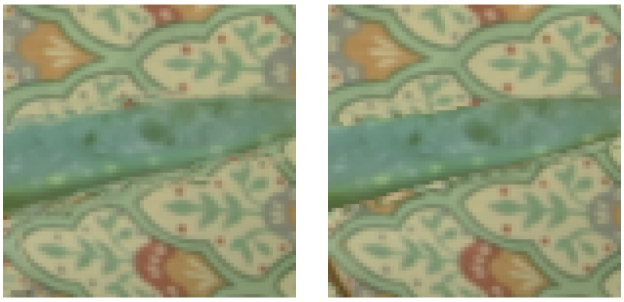
(a) Proposed method (b) Conventional method

Fig. 7. Zoomed synthesized view (teddy).



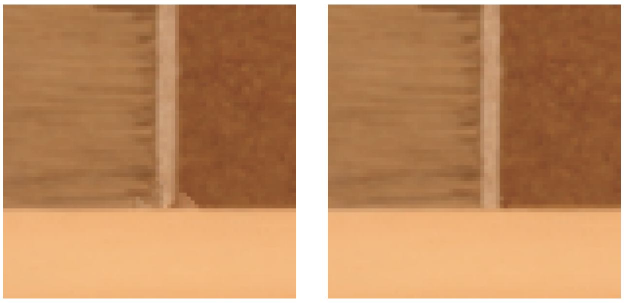
(a) Proposed method (b) Conventional method

Fig. 8. Zoomed synthesized view (laundry).



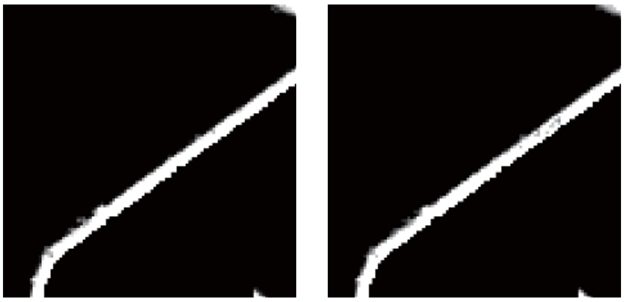
(a) Proposed method (b) Conventional method

Fig. 9. Zoomed synthesized view (aloe).



(a) Proposed method (b) Conventional method

Fig. 10. Zoomed synthesized view (wood2).



(a) Proposed matting (b) Robust matting [9]

Fig. 11. Zoomed alpha map of robust matting (teddy).

Figure 9, 10 shows synthesized images at “aloe” and “wood2” sequences of the proposed and conventional. In these sequences, the proposed method has lower PSNR than the conventional method. The reason for low PSNR is different in “aloe” and “wood2” sequence. In the “aloe” sequence, the reason is that background textures are too complex. The proposed method cannot reconstruct complex textures. Therefore, the proposed method generates low quality background strip images when background textures of input images are complex like “aloe” sequence. In the “wood2” sequence, the reason is that the texture of the background is too simple, but cyclic. In addition, foreground color and background color are similar. In this case, pasting non matting pixels directly has better performance than the images which are processed by the complex alpha matting. Our matting cannot generate complex patterned texture on matting regions. The blended background textures tend to be flat. In the textured regions, this process is not suitable. In contrast, direct copying (non matting) can mix (or has already mixed) foreground and background pixels, if the background texture of warped region and of pre-warped region is almost same.

Next, we compare our method based on filtering and the other matting method. We use robust matting [9] instead of Bayesian matting, which is used in the previous researches of view synthesis with matting [5, 6]. The robust matting is more accurate, and also use trimaps. The trimap used in our experiment is generated from the boundary map. We set the trimap unknown, if the boundary map is 0 or 255. Also, we set the trimap foreground, if the boundary map is non boundary and the padded value is 255. Background is labeled in the same manner. Note that robust matting and the proposed matting have same size of the matting region in this condition. Then, alpha map is obtained by the robust matting. Following steps, such as foreground/background generation and view synthesis, are same as the proposed method.

In Table 1, the proposed method has higher PSNR (0.20 dB on the average) than the robust matting. Figure 11 shows zoomed alpha maps of the proposed and the robust matting on “teddy” sequence. We can observe outliers in the alpha map generated by the robust matting. The robust matting is more accurate in almost regions, but more sensitive than filter based method. Thus there are outliers in the alpha map. For this reason, the proposed method is suitable for view synthesis.

Table 2 shows computing time of two alpha matting methods; the proposed alpha matting and the robust matting. As a result, the proposed alpha matting is the fastest method, and have real-time capability.

Table 1. PSNR (dB) of various images. Bold datasets are shown in figures 7-10.

Input image	Matting (Prop.)	Non matting ([1])	robust matting ([9])
average	37.22	36.82	37.02
teddy	33.73	33.37	33.57
cones	30.57	30.56	30.49
art	34.03	32.65	33.14
dolls	35.25	34.89	35.12
laundry	34.03	32.49	33.75
reindeer	36.01	34.89	33.75
baby1	40.57	39.95	40.57
baby2	37.83	37.65	36.91
baby3	35.38	35.14	35.33
bowling1	38.61	37.80	38.60
bowling2	34.62	34.22	34.50
cloth2	39.42	38.58	39.41
flowerpots	31.78	31.70	31.75
lampshade2	42.40	41.04	42.36
midd1	34.60	33.24	34.68
midd2	35.75	34.90	35.68
monopoly	36.92	36.91	36.89
plastic	43.19	39.59	43.64
rocks2	39.06	39.05	38.79
wood1	44.50	44.30	44.43
books	35.13	36.41	35.15
moebis	37.20	37.63	36.53
aloe	34.31	34.46	34.00
cloth1	40.99	41.72	40.88
cloth3	38.59	39.16	38.04
cloth4	36.34	36.41	38.04
lampshade1	41.40	41.50	40.16
rocks1	38.96	39.05	38.79
wood2	38.30	38.73	38.50

Table 2. Computing time of various alpha matting methods. Each method is written in C++ with Visual Studio2012 on Windows 7 64bit, and tested on Intel Core i7 920 2.93GHz.

Prop.	Robust Matting
30 ms	38000 ms

4. CONCLUSION

In this paper, we proposed a filter based matting for view synthesis, and were evaluated by using about 30 data sets from Middlebury. The results showed that the proposed method improves the subjective quality at object boundaries and has the higher PSNR (0.40 dB) than the non matting method [1], and has the higher PSNR (0.20 dB) than the conventional matting of robust matting [9].

In addition, its computational speed is 100x faster than the robust matting. The code used in this paper is uploaded in `http://nma.web.nitech.ac.jp/fukushima/research/viewsynthesis.html`

In our future works, we will extend the matting by using stereo matting [11] to improve the accuracy of matting. Also, instead of using ground truth, we use estimated depth maps, which have accurate boundaries, by using side information of manual user input [12], or boundary refinement method [13].

5. REFERENCES

[1] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for ftv," *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 65–72, Jan. 2009.

[2] S. Zinger, L. Do, and P.H.N. de With, "Free-viewpoint depth image based rendering," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5, pp. 533–541, July 2010.

[3] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[4] J. Wang and M. F. Cohen, "Image and video matting: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, Jan. 2007.

[5] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 600–608, Aug. 2004.

[6] S.-C. Chan, Z.-F. Gan, K.-T. Ng, K.-L. Ho, and H.-Y. Shum, "An object-based approach to image/video-based synthesis and processing for 3-d and multiview televisions," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 821–831, June 2009.

[7] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *Proc. Computer Vision and Pattern Recognition*, vol. 2, pp. II–264 – II–271, June 2001 .

[8] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, June 2013.

[9] J. Wang and M. F. Cohen, "Optimized color sampling for robust matting," in *Proc. Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[10] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, Apr.–June 2002.

[11] M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann, "A stereo approach that handles the matting problem via image warping," in *Proc. Computer Vision and Pattern Recognition*, pp. 501–508, June 2009.

[12] M. O. Wildeboer, N. Fukushima, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "A semi-automatic multi-view depth estimation method," in *Proc. SPIE Visual Communications and Image Processing*, pp. 77 442B–8, Aug. 2010.

[13] T. Matsuo, N. Fukushima, and Y. Ishibashi, "Weighted joint bilateral filter with slope depth compensation filter for depth map refinement," in *International Conference on Computer Vision Theory and Applications*, Feb. 2013.