VIEW SYNTHESIS WITH HIERARCHICAL CLUSTERING BASED OCCLUSION FILLING

Ji Dai, Truong Nguyen

University of California, San Diego
Department of Electrical and Computer Engineering
9500 Gilman Drive, La Jolla, CA 92093

1. ABSTRACT

This paper presents a depth image based rendering algorithm for view synthesis task. We address the challenging occlusion filling problem with a hierarchical clustering approach. Depth distribution of neighboring pixels around each occlusion is explored and from which we determine the number of surrounding depth planes with agglomerative clustering. Pixels in the most distant plane are picked as candidates to restore that occlusion. The proposed algorithm is evaluated on Middlebury stereo dataset and Microsoft Research 3D video dataset. Results show that our method ranks among the best performers.

Index Terms— View Synthesis, Free Viewpoint, Depth Image Based Rendering, Occlusion Filling, Hierarchical Clustering

2. INTRODUCTION

Affordable virtual reality devices and accurate real-time viewpoint tracking sensors have made possible a new class of immersive viewing experiences, allowing user to interactively observe a dynamic scene from arbitrary perspectives. This trend also creates demands on producing related visual contents such as free viewpoint video, panorama video and etc. While multi-camera systems are commonly deployed in capturing these contents, user might pick a viewpoint where no physical camera is installed. In such case, the view needs to be synthesized.

Depth image based rendering (DIBR) is a popular approach for view synthesis problem [1]. Standard DIBR performs 3D warping with texture and depth information from the reference viewpoints to create the new view. Various inpainting techniques are applied to deal with the occlusions in warped view. These missing pixels are not visible in any input images, and therefore to restore them in a visually acceptable way is challenging.

This paper explores one of the most typical DIBR problem as shown in Fig.1: Given a pair of rectified images and corresponding disparity maps, how to synthesize the view from a new vantage point on the baseline? Note that our method can also be generalized to unrectified cases and we show that in Sec.6.2. A pipeline of proposed algorithm is provided in Fig.2. We first generate two sets of intermediate color images and depth maps from two reference views individually using 3D warping. Pixels warped from significant depth edges in the reference views are processed to remove ghost contours. The intermediate results are then merged together with alpha blending. We propose a hierarchical clustering based occlusion filling technique. For each occluded area, its neighboring pixels are clustered into groups based on the depth distribution. Each group is regarded as a depth plane. The occluded area is considered as part of the farthest plane and only valid pixels in that plane are used for restoration. Depth map is filled first and assists the inpainting for color image. Proposed algorithm obtained decent results on Middlebury and Microsoft 3D video datasets.

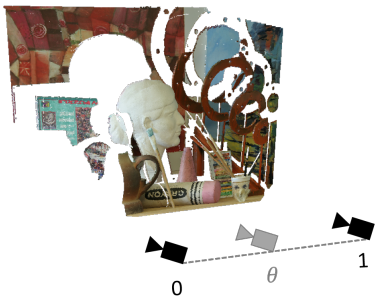


Fig. 1. A typical DIBR problem: given views at 0 and 1, how to synthesize view at $\theta \in (0, 1)$?

3. RELATED WORK

View synthesis has been an active research field over a long period and a comprehensive review of related techniques can be found in [2]. One common approach is using model-based rendering [3, 4] , which requires building a 3D scene model, typically polygon mesh.

Another mainstream approach is DIBR, which avoids the explicit modeling step and replaces it with a 3D warping process. Zinger et al. [5] present a standard DIBR pipeline.

The paper indicates some major challenges in DIBR such as dealing with unreliable depth estimation and occlusion filling. For artifacts caused by unreliable depth estimation, [6, 7] use layered representation to label the reliability of pixels on edges. The layers are warped individually and then merged together. A list of occlusion filling methods and their performance comparison is available in [8].

Under similar problem setting as ours, [9, 10] use depth information to assist the occlusion inpainting process. Tran et al. [11] improve it with a lattice structured conditional random field and graph cuts minimization in recovering unreliable and occluded pixels. Lim et al. [12] present an exemplar-based inpainting framework. Jain et al. [13] propose an efficiency-oriented algorithm aiming for real-time performance with minimal performance loss.

4. VIEW SYNTHESIS

4.1. 3D Warping

A general formulation of 3D warping is presented in [14]. If reference views are rectified and new view locates on the baseline, the problem is reduced to a trivial task of shifting the pixel in the reference views horizontally by its scaled disparity. Given color images $I_{0,1}$ and disparity maps $d_{0,1}$ at reference viewpoints, the intermediate color images $I^t_{0,1}$ and depth maps $D^t_{0,1}$ are generated as:

$$\begin{aligned} I^t_0(i, j - \theta * d_0(i,j)) &= I_0(i,j) \\ I^t_1(i, j + (1-\theta) * d_1(i,j)) &= I_1(i,j) \\ D^t_0(i, j - \theta * d_0(i,j)) &= f_0 \cdot b / d_0(i,j) \\ D^t_1(i, j + (1-\theta) * d_1(i,j)) &= f_1 \cdot b / d_1(i,j), \end{aligned}$$

$f_{0,1}, b$ are the focal length and stereo baseline. For warping destinations (i, j') with non-integer column index $j' \notin \mathbb{Z}$, we warp it to both $(i, \lceil j' \rceil), (i, \lfloor j' \rfloor)$ to mitigate crack artifacts. If multiple pixels are warped to the same destination, the one with smallest depth value is retained. Depth map is converted back to disparity map only for displaying purpose in the paper.

4.2. Ghost Contour Removal

Edges are usually sharp in depth image. However, their counterparts in color image tend to have a transition width. Edge pixels with background depth but contaminated with foreground color will cause ghost contour artifacts if not handled properly (see Fig.4(b)). We first detect the significant edges in the two reference views by applying Canny edge detector [7] with a threshold τ_e to the depth maps. Detected edges are dilated and pixels within are divided into background and foreground (Fig.4(a)). Pixels in I^t_0 and I^t_1 which are warped from these background edge pixels are replaced by median filtering their neighbors. Fig.4(c) shows the same region after correction.

4.3. Merge Intermediate Results

I^t_0, I^t_1 and D^t_0, D^t_1 are merged into I^t and D^t. If pixel $\boldsymbol{x}$ is visible in both intermediate results and $|D^t_0(\boldsymbol{x}) - D^t_1(\boldsymbol{x})| \leq \tau_d$, we blend them by

$$\begin{aligned} I^t(\boldsymbol{x}) &= (1-\theta) \cdot I^t_0(\boldsymbol{x}) + \theta \cdot I^t_1(\boldsymbol{x}) \\ D^t(\boldsymbol{x}) &= (1-\theta) \cdot D^t_0(\boldsymbol{x}) + \theta \cdot D^t_1(\boldsymbol{x}) \end{aligned}$$

The information from the closer reference view is assigned with larger weight. If $|D^t_0(\boldsymbol{x}) - D^t_1(\boldsymbol{x})| > \tau_d$, we retain the pixel from intermediate result $i = \arg\min_i D^t_i(\boldsymbol{x})$. The intuition behind is closer objects occlude farther objects.

5. HOLE FILLING

Holes in I_t, D_t can be caused by two main reasons. One is the insufficient sampling rate in reference images: as viewpoint changes, objects occupy more pixels than they do in the reference viewpoints; the other is occlusion: regions blocked in the reference views are revealed in the new view.

We notice that holes may locate in the middle of one depth plane or at the intersection of multiple depth planes (see Fig.5(a)). In the first case, the hole is considered as part of the plane. In the second case, we observe that the hole is very likely to be part of the background occluded by the foreground objects in reference viewpoints. Therefore it makes sense to label the hole as part of the farthest neighboring depth plane.

5.1. Hierarchical Clustering

We explore the depth distribution of neighboring pixels around each hole to determine the number of surrounding depth planes. For a hole H_i, we apply agglomerative hierarchical clustering to its 3rd-order neighboring pixels based on their depth value: each pixel starts as a cluster. Two clusters A, B will be merged if

$$\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} |D^t(a) - D^t(b)| \leq \tau_d.$$

The merging order is based on a greedy manner, meaning closest clusters merge first. The merging process stops when all pairwise distances among remaining clusters exceed τ_d. Fig.5(b) shows the number of remaining clusters around each hole. Each cluster is regarded as a depth plane.

5.2. Depth Hole Filling

Pixels in the farthest remaining cluster $\mathcal{C}(H_i)$ will help restoring H_i. Filling starts from pixels directly at the border with these selected pixels and gradually propagate towards the remaining area of the hole. For pixel $\boldsymbol{x}$ inside holes H_i:

$$D^t(\boldsymbol{x}) = \text{ median } (D^t(\boldsymbol{y})),$$

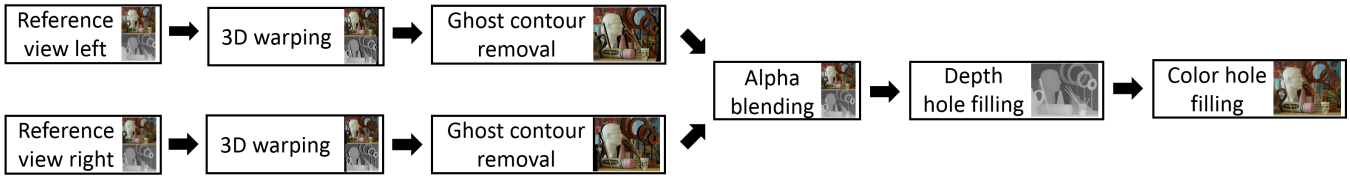
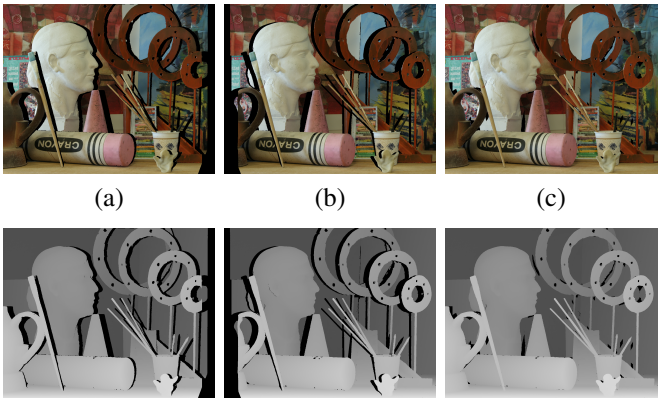


Fig. 2. Pipeline of the proposed algorithm: outputs at each step are shown.



(a) (b) (c)

(d) (e) (f)

Fig. 3. (a),(b),(d),(e) show the color images and disparity maps of two intermediate results from set *Art* in Middlebury dataset. (c),(f) show merged color image and disparity map.

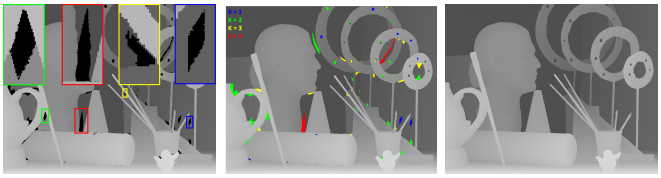
where $\boldsymbol{y} \in \mathcal{N}^3(\boldsymbol{x}) \cap \mathcal{C}(H_i)$, $\mathcal{N}^3$ denotes 3rd-order neighbor. Once $\boldsymbol{x}$ is filled, it is included in $\mathcal{C}(H_i)$.

5.3. Color Hole Filling

Holes in the color image are filled in a similarly directional manner. Instead of median filter, we use a variation of bilateral filter [15].

(a) (b) (c)

Fig. 4. (a) shows the dilated significant edges in reference view 0. Foreground pixels are colored blue and background pixels are colored green. (b) shows a region with ghost contours and (c) shows the same region after correction.



(a) (b) (c)

Fig. 5. (a) shows holes surrounded by different number of depth planes. (b) shows numbers of remaining clusters after hierarchical clustering (blue = 1, green = 2, yellow = 3 and red = 4). (c) shows filled disparity map

$$I^t(\boldsymbol{x}) = \frac{1}{Z} \sum_{\boldsymbol{y} \in \mathcal{N}^3(\boldsymbol{x})} w_s(\boldsymbol{y}) w_d(\boldsymbol{y}) I^t(\boldsymbol{y})$$

$$Z = \sum_{\boldsymbol{y} \in \mathcal{N}^3(\boldsymbol{x})} w_s(\boldsymbol{y}) w_d(\boldsymbol{y}),$$

Z is a normalization term. w_s here is the weight function depicting the spatial closeness between $\boldsymbol{y}$ and $\boldsymbol{x}$:

$$w_s(\boldsymbol{y}) = e^{-\|\boldsymbol{y}-\boldsymbol{x}\|_2^2/\sigma_s}$$

w_d is the weight function relating to the depth closeness:

$$w_d(\boldsymbol{y}) = \begin{cases} e^{-|D^t(\boldsymbol{y})-D^t(\boldsymbol{x})|^2/\sigma_d}, & |D^t(\boldsymbol{y}) - D^t(\boldsymbol{x})| \le \tau_d \\ 0, & |D^t(\boldsymbol{y}) - D^t(\boldsymbol{x})| > \tau_d \end{cases}$$

w_d prevents $\boldsymbol{x}$ from being contaminated by pixels inside other depth planes, which usually represent other objects. The filled color image undergoes median filtering on the edges to smooth out the sawtooth artifacts. Fig.6(a) shows the final output color image.

6. EXPERIMENTAL RESULT

6.1. Middlebury Stereo Dataset

We evaluated the proposed algorithm on the Middlebury stereo dataset (2005,2006) [20]. The dataset contains 27 sets of multi-view images. Each set includes a pair of rectified color images and corresponding disparity maps. Ground truth color images for three in-between views ($\theta = 0.25, 0.5, 0.75$)

Middlebury Dataset	Jain et al. [13]		Tran et al. [16]		Ramachandran et al. [10]		Proposed	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Art	31.67	0.95	32.66	0.95	30.22	0.94	**32.86**	**0.98**
Books	30.10	0.93	30.92	0.93	28.74	0.92	**32.55**	**0.96**
Cloth1	35.04	0.96	35.99	0.97	33.66	0.94	**38.28**	**1.00**
Dolls	31.61	0.95	33.05	0.95	30.90	0.94	**33.64**	**0.98**
Laundry	31.66	0.95	32.13	0.95	31.32	0.94	**32.81**	**0.98**
Moebius	33.42	0.95	34.30	0.94	32.76	0.93	**35.48**	**0.98**
Monopoly	29.80	0.95	32.19	0.95	28.77	0.93	**33.56**	**0.99**
Plastic	37.91	0.98	37.77	0.98	37.95	0.98	**40.32**	**1.00**
Reindeer	32.79	0.95	33.70	0.94	not reported	not reported	**34.85**	**0.99**
Wood1	36.29	0.94	37.47	0.94	not reported	not reported	**38.59**	**1.00**
Average	33.03	0.95	34.02	0.95	31.84	0.93	**35.30**	**0.98**

Table 1. PSNR and SSIM for proposed method and three state-of-art methods

Fig. 6. Left image shows the final output color image. Right image shows the ground truth image.

Fig. 7. Left image shows a synthesized frame of *Breakdancers* sequence. Right image shows the ground truth.

are also provided. In Table 1, we report PSNR and SSIM scores for 10 sample sets ($\theta = 0.5$) of proposed and 3 state-of-art algorithms. The results show that our method outperforms others in all reported sets and achieves an average PSNR of 35.30dB and SSIM of 0.98, leading the second best algorithm by 1.28dB in PSNR and 0.03 in SSIM.

6.2. Microsoft Research 3D Video Dataset

We extended the experiment to the Microsoft Research 3D video dataset [6]. The dataset includes *Ballet* and *Breakdancers* video sequences. Each sequence provides 100 frames of color images and depth maps for 8 cameras. The cameras are positioned on an arc spanning about 20 degrees from one end to the other. We select camera 3 and 5 as reference viewpoints and camera 4 as new viewpoint. The warping step in Sec.4.1 was replaced with the general 3D warping process and θ in Sec. 4.3 equals $\frac{\|\boldsymbol{T_l}-\boldsymbol{T_v}\|}{\|\boldsymbol{T_l}-\boldsymbol{T_v}\|+\|\boldsymbol{T_v}-\boldsymbol{T_r}\|}$, where $\boldsymbol{T_l}, \boldsymbol{T_r}, \boldsymbol{T_v}$ are the translation vectors for left/right reference view and new view respectively. We report average PSNR and SSIM scores over 100 frames and comparison with other methods in Table 2. A sample frame is shown in Fig.7. Unlike most reported algorithms, which utilize interframe information to enforce temporal consistency, we treated each frame independently to be persistent with the problem setting in this paper. Results indicate the proposed algorithm has the best score for *Ballet* and the second best for *Breakdancers*. We plan to incorporate temporal information in future work to further improve the performance of the proposed algorithm.

	Ballet		Breakdancers	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Oh [9]	32.50	0.87	31.75	0.83
Loghman [17]	30.36	0.92	31.64	0.91
VSRS [18]	30.23	0.90	31.17	0.89
Liu [19]	32.52	0.94	**33.33**	**0.92**
Proposed	**32.55**	**0.96**	31.77	0.90

Table 2. Comparison results on Microsoft 3D dataset: use view 3 and view 5 to generate view 4.

Complete results of aforementioned two experiments are available at http://jidai.me/view-synthesis/.

7. CONCLUSION

We propose a hierarchical clustering based framework for the challenging occlusion filling problem in the view synthesis task. The proposed algorithm is tested on rectified (Middlebury) and unrectified (Microsoft Research 3D video) datasets. Evaluation shows our method achieves good results under both schemes.

8. REFERENCES

[1] Christoph Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[2] Daniel Scharstein, *View synthesis using stereo vision*, Springer-Verlag, 1999.

[3] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 69, 2015.

[4] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons, "Towards high-resolution large-scale multi-view stereo," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1430–1437.

[5] Sveta Zinger, Luat Do, and PHN de With, "Free-viewpoint depth image based rendering," *Journal of visual communication and image representation*, vol. 21, no. 5, pp. 533–541, 2010.

[6] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG)*. ACM, 2004, vol. 23, pp. 600–608.

[7] Karsten Muller, Aljoscha Smolic, Kristina Dix, Philipp Merkle, Peter Kauff, and Thomas Wiegand, "View synthesis for advanced 3d video systems," *EURASIP Journal on image and video processing*, vol. 2008, no. 438148, pp. 11, 2008.

[8] Shafik Huq, Andreas Koschan, and Mongi Abidi, "Occlusion filling in stereo: Theory and experiments," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 688–704, 2013.

[9] Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Picture Coding Symposium, 2009. PCS 2009*. IEEE, 2009, pp. 1–4.

[10] Geetha Ramachandran and Markus Rupp, "Multiview synthesis from stereo views," in *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2012, pp. 341–345.

[11] Lam C Tran, Christopher J Pal, and Truong Q Nguyen, "View synthesis based on conditional random fields and graph cuts," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 433–436.

[12] Hwasup Lim, Yong Sun Kim, Seungkyu Lee, Ouk Choi, James DK Kim, and Changyeong Kim, "Bi-layer inpainting for novel view synthesis," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1089–1092.

[13] Ankit K Jain, Lam C Tran, Ramsin Khoshabeh, and Truong Q Nguyen, "Efficient stereo-to-multiview synthesis," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 889–892.

[14] Ilkoo Ahn and Changick Kim, "A novel depth-based virtual view synthesis method for free viewpoint video," *IEEE Transactions on Broadcasting*, vol. 59, no. 4, pp. 614–626, 2013.

[15] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[16] Lam C Tran, Can Bal, Christopher J Pal, and Truong Q Nguyen, "On consistent inter-view synthesis for autostereoscopic displays," *3D Research*, vol. 3, no. 1, pp. 1–10, 2012.

[17] Maziar Loghman and Joohee Kim, "Segmentation-based view synthesis for multi-view video plus depth," *Multimedia Tools and Applications*, vol. 74, no. 5, pp. 1611–1625, 2015.

[18] "Software for view synthesis," http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/mpeg2/VS.htm.

[19] Jing Liu, Chunpeng Li, Xuefeng Fan, Zhaoqi Wang, Min Shi, and Jie Yang, "View synthesis with 3d object segmentation-based asynchronous blending and boundary misalignment rectification," *The Visual Computer*, vol. 32, no. 6-8, pp. 989–999, 2016.

[20] Heiko Hirschmuller and Daniel Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.