NOVEL VIEW SYNTHESIS WITH LIGHT-WEIGHT VIEW-DEPENDENT TEXTURE MAPPING FOR A STEREOSCOPIC HMD

Thiwat Rongsirigul, Yuta Nakashima, Tomokazu Sato, and Naokazu Yokoya

Nara Institute of Science and Technology
{thiwat.rongsirigul.tk5, n-yuta, tomoka-s, yokoya}@is.naist.jp

ABSTRACT

The proliferation of off-the-shelf head-mounted displays (HMDs) let end-users enjoy virtual reality applications, some of which render a real-world scene using a novel view synthesis (NVS) technique. View-dependent texture mapping (VDTM) has been studied for NVS due to its photo-realistic quality. The VDTM technique renders a novel view by adaptively selecting textures from the most appropriate images. However, this process is computationally expensive because VDTM scans every captured image. For stereoscopic HMDs, the situation is much worse because we need to render novel views once for each eye, almost doubling the cost. This paper proposes light-weight VDTM tailored for an HMD. In order to reduce the computational cost in VDTM, our method leverages the overlapping fields of view between a stereoscopic pair of HMD images and pruning the images to be scanned. We show that the proposed method drastically accelerates the VDTM process without spoiling the image quality through a user study.

Index Terms— Novel view synthesis, head-mounted displays, image-based rendering

1. INTRODUCTION

Recently, the proliferation of off-the-shelf head-mounted displays (HMDs) lets end-users enjoy virtual reality (VR) applications, such as video games and telepresence. Some of these applications aim to provide users with experiences of being at the remote place by virtually immersing them into a real-world scene. To achieve this, the applications synthesize a novel view of the real-world scene and present it to the users. In the present, such applications are studied in many scenarios, such as a virtual preview of a remote place [1].

A spectrum of methods for novel view synthesis (NVS) have been proposed in between two extrema: model-based and image-based (e.g. [2]) methods. The model-based methods render a reconstructed or handcrafted 3D model at a given view point. A texture may be then applied to the 3D model to give a realistic look and alleviate the imperfection of the 3D model in the case of automatic model reconstruction. On the other hand, the image-based methods basically interpolate densely captured camera images of the real-world scene without using scene geometry. These methods give a photo-realistic result; however, they cannot handle a large-scale scene because they usually require a huge number of images.

Fig. 1. A user using our proposed prototype system.

Among these methods for NVS, view-dependent texture mapping (VDTM) [3] has been studied for its photo-realistic image quality with the capability of reproducing view-dependent components, such as reflection. VDTM renders reconstructed or handcrafted 3D model similarly to the model-based methods, but it adaptively selects an image to be applied as a texture to the model for the user's current view point. Unlike purely image-based methods, VDTM requires fewer images in order to synthesize novel views while still preserving their image quality.

However, the image selection process in VDTM is computationally expensive since it needs to search through all captured images. In the case of a stereoscopic HMD, the computational cost for the selection is doubled as an HMD has two views for left and right eyes as shown in Fig. 1. On top of that, HMDs usually require a higher frame rate (i.e. over 60 frames per second) compared to a regular displays because the views presented to both eyes of the user must immediately follow his head movement. The combination of the extra computation and requirement may hinder from using the VDTM technique for HMDs.

This paper proposes a VDTM-based system with a low computational cost, or a light weight system, for NVS tailored for an HMD. Our system accelerates the image selection process by prioritizing the images to be searched. Moreover, our system leverages the overlap of left- and right-eyes' fields of view. The system synthesizes a novel view from scratch only for the left eye and transfer the result to the right-eye view. It

978-1-5090-6067-2/17/$31.00

then fills missing pixels in the right-eye view due to the slight difference in the fields of view by applying our light-weight VDTM process only to these pixels. The contribution of this paper can be summarized as follows.

- We design a light-weight VDTM method for an HMD, which synthesizes a pair of novel views for left and right eyes. To the best of our knowledge, this is the first attempt in this direction.
- In order to speed up image selection in VDTM, we propose to prioritize images to be searched based on the distance between the camera that captured one of input images and the user's current view point. We achieve roughly three times increase in performance compared to a conventional method.
- We further reduce the computational cost by almost half by exploiting overlapping fields of view of left and right eyes of a stereoscopic HMD.
- We demonstrate through our user study that the proposed system is able to achieve real-time rendering at nearly 90 frames per second on a stereoscopic HMD without spoiling the image quality.

2. RELATED WORKS

A wide variety of techniques for synthesizing novel views of real-world objects or scenes have been proposed in between two extreme ends of NVS techniques.

The model-based methods, on one end, use a 3D model of real objects/scenes. Since handcrafting 3D models is costly, some automatic techniques for 3D reconstruction can be used instead [4, 5, 6, 7]. For example, Jancosek and Pajdla proposed a method for multi-view stereo, whose output can be adopted for this purpose. Existing work for 3D reconstruction is summarized and compared in [8]. These automatic reconstruction techniques, however, usually give an imperfect 3D model.

On the other end, the image-based methods, which synthesize novel view images using only captured images without utilizing any geometrical information. McMillan proposed [9] to uses a plenoptic function originally proposed by Adelson [10]. Based on Adelson's plenoptic function, Levoy and Hanrahan proposed the light field rendering technique [2], and Gortler proposed the Lumigraph technique [11]. These image-based methods give photo-realistic results if the number of captured image is sufficiently large. However, in a large-scale scene, it is almost impossible to capture a sufficient amount of images to get an acceptable result from such methods.

In between these two extreme ends, several hybrid techniques have been proposed to reduce the number of images that the image-based methods require while still preserving a photo-realistic result. One such technique is Debevec's VDTM [3]. This technique significantly reduces the number of images required while still preserving the resulting image quality. Many researchers have studied this technique because of its capability of reproducing view dependent features such as specular reflection. Some existing works [12, 13, 14, 15, 16] obtain the 3D mesh model of the scene geometry through an automatic reconstruction technique while the other [17] simply uses a 3D point cloud representation of the scene. In original VDTM, Debevec used a simplified 3D model of the real scene environment [18]. Similarly, Buehler proposed [19] to use a simplified model as a proxy of the scene geometry instead of a fully reconstructed 3D model.

As NVS techniques for a stereoscopic HMDs, Chen [13] and Rakuten Inc. [1] proposed prototypes tailored for such devices. Chen [13]'s prototype uses KinectFusion [20] to obtain the geometry of small objects and applies texture. Rakuten Wedding [1] is a pure image-based method which interpolates between captured images without using a scene geometry.

Our system is based on VDTM and thus also lies between the image-based and model-based methods. We use an automatically reconstructed scene geometry. Due to imperfections of the reconstructed 3D model, we use VDTM [3] to achieve a photo-realistic result. Unlike Chen's prototype [13], our prototype is capable of reconstructing a real-world scene, but not small objects, captured by a standard camera. We also accelerate the original VDTM technique to achieve a real-time performance in a large real-world scene with a large number of input images.

3. OVERVIEW OF OUR SYSTEM

The VDTM technique is originally designed to reproduce the details of a scene from a simplified 3D model [18], which include view-dependent components, such as highlight on reflective surfaces. For this goal, given a 3D mesh model M of the scene and a set of images $S = \{I_n | n = 1, \ldots, N\}$, where N is the number of images, VDTM adaptively selects suitable images to pick out the color of a point or texture of a triangle of model M. This image selection process is usually expensive since it searches all images in S for every pixel or triangle.

The situation is much worse for stereoscopic HMDs as they have a pair of displays for left and right eyes, which requires running the image selection process twice. In addition to this, the user experience with HMDs depends much on the frame rate: HMDs usually track the user's head motion and update the view according to it. Insufficient frame rate may cause the user perceiving a significant difference between his/her head motion and what he/she watches, and this ends up with spoiling immersiveness. Our ideas to increase the frame rate are summarized as follows.

Image prioritization. Supposing the novel view image for one eye has K pixels, which usually ranges from sub-millions to a few millions depending on the HMD's display, the image selection process of the original VDTM technique

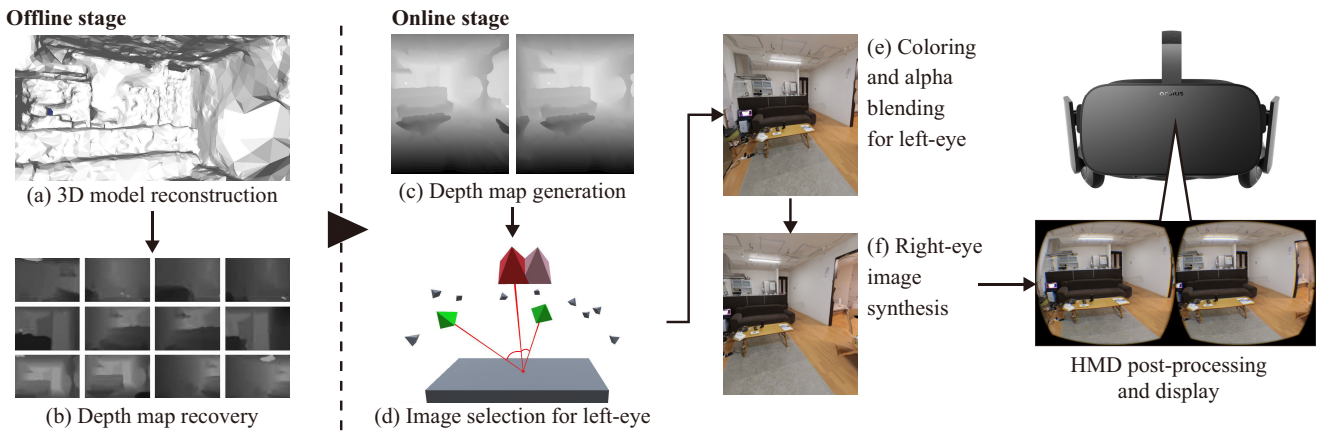

Fig. 2. System overview.

searches N images K times. Therefore, limiting the number N drastically reduces the computational cost for image selection. Inspired by [19], we prioritize the camera closer to the current virtual view point in term of its distance before image selection. More specifically, we sort the images in S in the ascending order of their distances to the current virtual view point, and then search the image until we find N' images that meet a certain criterion. By doing this, the number of images to be searched is drastically reduced.

One-time full rendering. We exploit the field-of-view similarity in left- and right-eye images of a stereoscopic HMD to skip rendering most pixels in the right-eye image. For the left-eye image, we perform the full VDTM process. For the right-eye image, since its field-of-view overlaps with the left-eye image, we copy the pixels of the left eye's rendering result to the corresponding pixels in the right-eye image. We then perform a full VDTM process to render pixels that are not covered by the left-eye's field-of-view. This allows us to mostly skip image selection for the right-eye image.

The proposed system consists of the offline and online stages as shown in Fig. 2. In the offline stage, we reconstruct a real scene as a 3D mesh model M from a set S of captured images using a multi-view stereo technique (e.g. [4]) (a). We then render the model from each captured image I_n's viewpoint with the standard rendering pipeline to recover the depth map D_n for later use (b). In the online stage, we render the model at given left-eye and right-eye virtual view points (C_{L} and C_{R}, respectively) again to obtain the depth maps D_{L} and D_{R} (c). Then the image selection process with image prioritization finds top two images that suit to picking the color to render the left-eye image (d). Due to differences in the lighting condition in each captured image, discontinuity in the texture may occur at the seam of different textures. To relieve such discontinuity, we color the corresponding pixel by applying alpha blending over the selected two images (e). We then synthesize the right-eye image using the rendered left-eye image to partially skip the image selection process (f). Finally, the HMD's post-processing is performed before the images are presented to the user through the HMD. Since the offline process only uses existing techniques, we detail the online stage in the following section.

4. VDTM FOR STEREOSCOPIC HMD

In the online stage, we firstly generate depth maps for left- and right-eye images D_{L} and D_{R} from the reconstructed mesh model M using a standard rendering pipeline. The view points and directions for both eyes are usually provided by the HMD's hardware, which are $\mathbf{R}_{\mathrm{L}}$ and $\mathbf{R}_{\mathrm{R}}$ for the rotation components as well as $\mathbf{t}_{\mathrm{L}}$ and $\mathbf{t}_{\mathrm{R}}$ for the translation components, together with its projection (virtual camera's intrinsic parameter) matrix $\mathbf{K}_{\mathrm{E}}$.

4.1. Image Selection for Left Eye

For pixel k ($k = 1, \ldots, K$) in the left-eye image, we find the two images in S that most suit to picking color up. Let $\mathbf{R}_n$ and $\mathbf{t}_n$ be the rotation and translation components of the camera C_n that captured the image I_n, and $\mathbf{K}_n$ the camera's intrinsic parameter matrix.

Using $\mathbf{R}_{\mathrm{L}}$, $\mathbf{t}_{\mathrm{L}}$, $\mathbf{K}_{\mathrm{E}}$, and the depth value $d_k^{\mathrm{L}} \in D_{\mathrm{L}}$ for pixel k in the left-eye image, we can regain the corresponding 3D position $\mathbf{p}_k$ in the model M's coordinate system, which we call the world coordinate system.

The original VDTM technique [3] firstly checks if $\mathbf{p}_k$ is visible in the image I_n in order for finding the two suitable images from S. For this, $\mathbf{p}_k$ is projected to D_n to find the corresponding depth value $d_{k'}^n$. We also transform $\mathbf{p}_k$ to the camera C_n's coordinate system and extract the third component z_k^n of the transformed vector, which corresponds to the depth. If $\mathbf{p}_k$ is visible in I_n, the difference between $d_{k'}^n$ and z_k^n should be small because they are both at the same point. Therefore, visibility v_k^n of $\mathbf{p}_k$ in I_n is given using a certain threshold ϵ by

$$v_k^n = \begin{cases} 1 & \text{if } |d_{k'}^n - z_k^n| < \epsilon \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

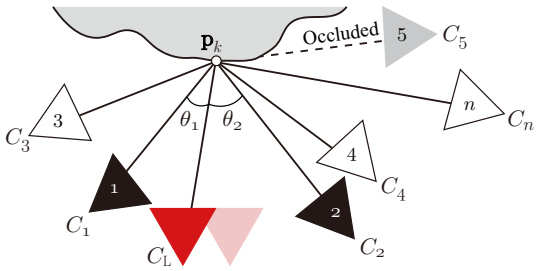

Fig. 3. An overview of the image selection process. Red: virtual view, Numbered Triangle: captured images. Black triangles indicate the selected top two images, and the number indicates the order the images are scanned.

Since the images' appearance is closer to a virtual view if they are captured from more similar viewing directions to the virtual view, original VDTM's image selection finds the closest viewing direction. Specifically, let the camera n's position be $\mathbf{a}_n = -\mathbf{R}_n^\top \mathbf{t}_n$ and the left-eye's position $\mathbf{a}_{\mathrm{L}} = -\mathbf{R}_{\mathrm{L}}^\top \mathbf{t}_{\mathrm{L}}$, the cosine of the angle θ_n formed by the left eye position, 3D position $\mathbf{p}_k$, and the camera C_n (as in Fig. 3) is given by

$$\cos\theta_n = \frac{(\mathbf{a}_n - \mathbf{p}_k)^\top (\mathbf{a}_{\mathrm{L}} - \mathbf{p}_k)}{\|\mathbf{a}_n - \mathbf{p}_k\| \, \|\mathbf{a}_{\mathrm{L}} - \mathbf{p}_k\|}. \tag{2}$$

The most suitable image is one that gives the largest value of $v_k^n \cos\theta_n$ with considering the visibility.

As mentioned above, this process is time consuming because we need to search all images in S to find the most suitable image for a single pixel. Our intuition to reduce this computational cost, which is inspired by [19], is that if the Euclidean distance between the camera n and the left-eye position is small, then $\cos\theta_n$ is large; therefore, we can prioritize images to be searched according to the distance. For this, we firstly sort the cameras in the ascending order of the distance $\|\mathbf{a}_n - \mathbf{a}_{\mathrm{L}}\|$, which does not depend on $\mathbf{p}_k$ and thus is done only once for every update of the image. Then, for pixel k, we compute $v_k^n \cos\theta_n$ in the ascending order of the distance. The criterion for terminating this process is that N images that satisfy $v_k^n \cos\theta_n > \xi$ are found. Among the images found, we select two images with highest $\cos\theta_n$. Since the cameras are sorted, it is more likely to find good images (i.e., a larger value of $v_k^n \cos\theta_n$) in an earlier stage of the search. Note that larger $\cos\theta_n$ does not necessarily mean a smaller distance. For example, if the camera is on the light ray from $\mathbf{p}_k$ to the left-eye position, then $\cos\theta_n$ is 1 while the distance can be arbitrarily large. This image prioritization has a good side effect that low resolution images (i.e., ones captured from farther viewpoints) are not selected.

4.2. Blending and Coloring for Left Eye

We color the pixel k in the left-eye image using selected two images with applying alpha blending over them for alleviating discontinuous seams between pixels whose selected images are different. The alpha value for the second image is calculated based on the angle θ_n of the two selected images. Let θ_1 and θ_2 be the angles of the first and second images respectively. The alpha value for the second image is given using a threshold T_b by

$$\alpha_2 = \begin{cases} \frac{0.5}{1-T_b}\left(\frac{\theta_1}{\theta_2} - T_b\right) & \text{if } \frac{\theta_1}{\theta_2} > T_b \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

In addition, if the projected point $\mathbf{p}_k$ lies near the edge of the first image, the alpha value also given to the second image based on the distance from the pixel to the image edge. For the first image, alpha value is simply determined by $\alpha_1 = 1 - \alpha_2$.

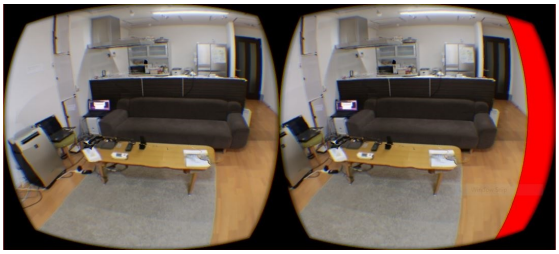

Fig. 4. Pixels that are not covered by the left-eye image is indicated in red in the right-eye image.

4.3. Right-eye Image Synthesis

In order to obtain the right-eye image, instead of repeating the same processes as the left-eye image, we exploit the rendering result for the left-eye image. For pixel k in the right-eye image whose corresponding 3D position in the world coordinate system is $\mathbf{p}_k$, we find the corresponding pixel's position $\mathbf{u}_k = (x_k \;\; y_k)^\top$ in the left-eye image by

$$\mathbf{u}_k = \Pi_{\mathbf{K}_{\mathrm{L}}}(\mathbf{R}_{\mathrm{L}}\mathbf{p}_k + \mathbf{t}_{\mathrm{L}}), \tag{4}$$

where $\Pi_{\mathbf{K}_{\mathrm{L}}}(\cdot)$ represents perspective projection of a 3D point using $\mathbf{K}_{\mathrm{L}}$. We copy the color at $\mathbf{u}_k$ in the left-eye image to the pixel k in the right-eye image. Due to the slight difference in eye positions, there are some pixels that are not covered by the left-eye image, which are indicated by red pixels in Fig. 4. We color these pixels by performing the VDTM process again.

5. EXPERIMENTAL RESULTS

We have implemented a prototype of our system and have conducted a user study to demonstrate the advantage of our system. Our prototype is running on a desktop Windows 10 PC with 16 GB RAM, Intel Core i7 4790K, and Nvidia GTX1080 GPU. It renders an image for each eye at 1332×1586 resolution to show through an Oculus Rift CV1 headset. We use OpenGL and Oculus SDK 1.3. Our prototype is implemented with C++ and GLSL. For user study, we have

Fig. 5. Examples of input images (top) and reconstructed 3D model (bottom).

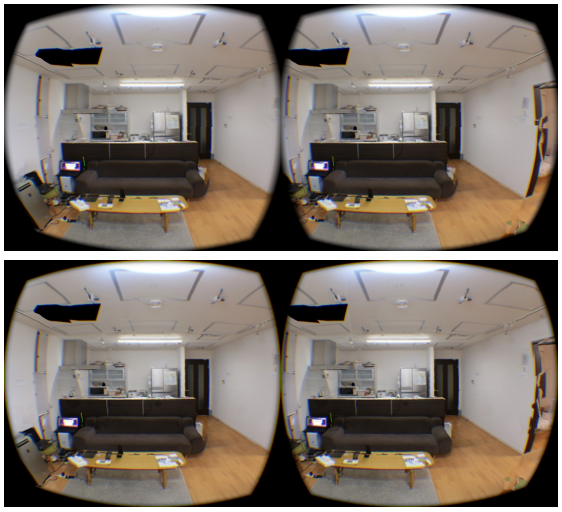

Fig. 6. Example of presented reconstructed scene from configuration ONE-3 (top) and TWO-UNL (bottom). UNL stands for unlimited.

built a dataset that contains 270 images of an in-room environment captured using a Canon EOS 6D camera, and a 3D mesh model with 39,697 triangles has been reconstructed from them using VisualSFM [21, 7, 22] and CMPMVS [4]. Figure 5 shows an example of input images and a portion of the reconstructed 3D mesh model, respectively. We empirically set the values of ϵ and T_b to 0.2 and 0.995, respectively.

Figure 6 shows example stereoscopic image pairs synthesized by our prototype, which uses image prioritization ($N' = 3$) and one-time full rendering (ONE-3), as well as one obtained by using the original VDTM technique separately applied for both eyes (TWO-UNL). The image quality deterioration by image prioritization and one-time full rendering is not significant in this example. At most view points, our prototype produces a good quality novel view image as shown in Fig. 7 (left). However, there are some noticeable seams in the texture as shown in Fig. 7 even though we adopt alpha blending. This is mainly because the limitation of the 3D reconstruction algorithms: Our automatically reconstructed 3D mesh model is not perfect, and the algorithm can not recover the scene geometry because point correspondences in homogeneous regions can not be found on reflective surfaces. Such 3D reconstruction errors cause inconsistent textures. Also,

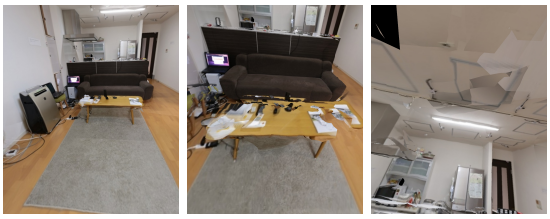

Fig. 7. Left-eye image (left) without failure in texturing and alpha blending, (middle) with errors in texturing, and (right) with errors in alpha blending.

Table 1. Comparison of averaged frame rate for each configuration. FPS stands for frame per second.

Configuration	FPS	Configuration	FPS
ONE-3	89.285	TWO-3	57.648
ONE-10	45.039	TWO-10	33.010
ONE-UNL	23.815	TWO-UNL	15.370

our alpha blending sometimes fails to smoothly connect textures from different images.

Table 1 shows the frame rates by 6 configurations of the VDTM process, which are combinations of image prioritization parameters ($N' = 3$, 10, and UNL, where UNL stands for unlimited) and whether one-time full rendering is applied (ONE: one-time full rendering and TWO: original VDTM for both eyes), for which further breakdown is provided in Fig. 8. According to Table 1, our prototype significantly improved the frame rate. This is achieved by significantly reducing the time needed to rendering the right-eye image by performing the one-time full rendering as shown in Fig. 8. Moreover, by image prioritization, we can improve the performance of traditional VDTM, which is also illustrated in Fig. 8.

In order to demonstrate the perceptual quality of our system, we recruited 13 participants in our user study, who never or rarely had experience on HMDs or VR before. Each participant was exposed through an HMD to stereoscopic image pairs by the 6 VDTM configurations. After the training session, in which all configurations are presented to the participants, they were asked to give scores ranging from 1 (bad) to 5 (good) to each configuration in regards to reconstructed image quality, smoothness of the video stream (if the frame rate is enough or not), and perceived reality of the presented scene. Figure 9 shows the average scores together with standard deviations. The high smoothness scores of our prototype (e.g. ONE-3) in Fig. 9 support our timing result in Table 1. We can also see that the quality of the image is nearly identical regardless of the configuration, which means that the frame rate is improved while preserving the image quality. We consider that this also makes the presented scene more perceptually realistic for the participants as shown in the realism scores.

6. CONCLUSION

We have proposed a light-weight VDTM-based novel view synthesis system designed for a stereoscopic HMD. We prior-

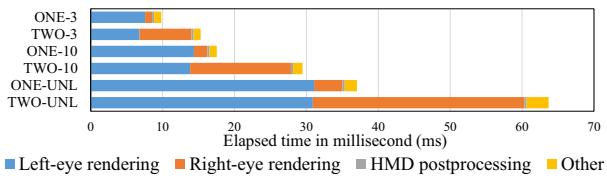

Fig. 8. Breakdown of timing results.

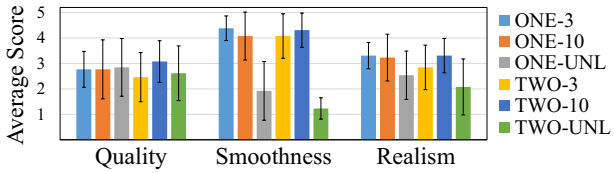

Fig. 9. Average subjective evaluation scores for each configuration.

itized the images that are searched in the image selection process in original VDTM based on the images' distances to current virtual eye positions. To further boost the performance, we take advantage of the overlapping fields of view between left- and right-eyes images: We perform the full VDTM process only for the left-eye image and then transfer the result to the right-eye image. Our system has been proven to increase the frame rate six times without much sacrificing the image quality. By improving the frame rate, the system provides the users with more realistic sensation. In future work, we will reduce the noticeable seam in synthesized images.

Acknowledgement This work is partly supported by Microsoft Research CORE Projects, and the texturing part is inspired by discussion with Prof. Katsushi Ikeuchi and Dr. Ambrosio Blonco at Microsoft Research.

7. REFERENCES

[1] RAKUTEN INC., "Saying 'I do'? First, get a virtual preview," 2016, `http://rakuten.today/blog/virtual-wedding.html` [Online; accessed 14-November-2016].

[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. ACM SIGGRAPH*, 1996, pp. 31–42.

[3] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Proc. Eurographics Rendering Workshop*, 1998, pp. 105–116.

[4] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 3121–3128.

[5] Y. Furukawa and J. Ponce, "Accurate, dense, afnd robust multi-view stereopsis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[6] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 1434–1441.

[7] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Int. Conf. 3D Vision*, 2013, pp. 127–134.

[8] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 519–528.

[9] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. ACM SIGGRAPH*, 1995, pp. 39–46.

[10] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, chapter 1. The MIT Press, 1991.

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. ACM SIGGRAPH*, 1996, pp. 43–54.

[12] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *Int. J. Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.

[13] C. F. Chen, M. Bolas, and E. Suma, "Real-time 3D rendering using depth-based geometry reconstruction and view-dependent texture mapping," *ACM Trans. Graph.*, pp. 84:1–84:2, 2016.

[14] F. Okura, Y. Nishizaki, T. Sato, N. Kawai, and N. Yokoya, "Motion parallax representation for indirect augmented reality," in *Proc. IEEE Int. Sympo. Mixed and Augmented Reality*, 2016, pp. 105–106.

[15] Y. Nakashima, Y. Uno, N. Kawai, T. Sato, and N. Yokoya, "AR image generation using view-dependent geometry modification and texture mapping," *Virtual Reality*, vol. 19, no. 2, pp. 83–94, 2015.

[16] K. Katagiri, Y. Nakashima, T. Sato, and N. Yokoya, "Novel view synthesis based on view-dependent texture mapping with geometry-aware color continuity," *Trans. the Virtual Reality Society of Japan*, vol. 21, no. 1, pp. 153–162, 2016.

[17] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 30:1–30:12, 2013.

[18] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. ACM SIGGRAPH*, 1996, pp. 11–20.

[19] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. ACM SIGGRAPH*, 2001, pp. 425–432.

[20] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Sympo. Mixed and Augmented Reality*, 2011, pp. 127–136.

[21] C. Wu, "VisualSFM : A visual structure from motion system," 2006, `http://ccwu.me/vsfm` [Online; accessed 14-November-2016].

[22] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)," 2006, `http://www.cs.unc.edu/~ccwu/siftgpu` [Online; accessed 14-November-2016].