Augmented Reality Image Generation with Virtualized Real Objects Using View-Dependent Texture and Geometry

Yuta Nakashima
NAIST
8916-5 Takayamacho, Ikoma
Nara, 630-0192 Japan
n-yuta@is.naist.jp

Tomokazu Sato
NAIST
8916-5 Takayamacho, Ikoma
Nara, 630-0192 Japan
tomoka-s@is.naist.jp

Yusuke Uno
NAIST
8916-5 Takayamacho, Ikoma
Nara, 630-0192 Japan

Naokazu Yokoya
NAIST
8916-5 Takayamacho, Ikoma
Nara, 630-0192 Japan
yokoya@is.naist.jp

Norihiko Kawai
NAIST
8916-5 Takayamacho, Ikoma
Nara, 630-0192 Japan
norihi-k@is.naist.jp

Abstract
Augmented reality (AR) images with virtualized real objects can be used for various applications. However, such AR image generation requires hand-crafted 3D models of that objects, which are usually not available. This paper proposes a view-dependent texture (VDT)- and view-dependent geometry (VDG)-based method for generating high quality AR images, which uses 3D models automatically reconstructed from multiple images. Since the quality of reconstructed 3D models is usually insufficient, the proposed method inflates the objects in the depth map as VDG to repair chipped object boundaries and assigns a color to each pixel based on VDT to reproduce the detail of the objects. Background pixel exposure due to inflation is suppressed by the use of the foreground region extracted from the input images. Our experimental results have demonstrated that the proposed method can successfully reduce above visual artifacts.

Author Keywords
AR with virtualized real objects, view-dependent texture, view-dependent geometry

ACM Classification Keywords
H.5.1 [Information Interfaces and Presentation]: Multimedia Information SystemsArtificial, augmented, and virtual realities

Figure 1: Examples of generated AR images. From top to bottom: CMPMVS [6], CMPMVS with VDT [3], and proposed method.

Introduction

Virtually arranging real objects in a real environment using augmented reality (AR) technology potentially has a wide range of applications such as virtual room planning [1]. Such applications usually require virtualizing the real objects, i.e., to create their 3D models so as to generate their arbitrary viewpoint images. One of the most straightforward approaches for virtualization is to hand-craft 3D models of the real objects. However, 3D modeling generally is a laborious task, requiring special skills. This may prevent ordinary developers and users from involving such applications. Therefore, automatic techniques for virtualizing real objects are strongly desired.

Recently, dense 3D reconstruction [4, 6] and image-based rendering [5, 8] have been studied, which can virtualize real objects from multiple images containing them. Among these techniques, dense 3D reconstruction is preferable to image-based rendering as it requires less input images than image-based rendering. However, even the state-of-the-art dense 3D reconstruction techniques usually suffer from insufficient accuracy, resulting in visual artifacts such as (i) loss of detailed shapes and texture and (ii) rough boundaries, as in the top image of Figure 1.

One approach to solve the above problems while using inaccurate 3D models is to use view-dependent texture (VDT) proposed by Debevec et al. [3]. VDT adaptively selects the image that best suits each triangle patch of the 3D model under a certain criterion and apply the corresponding image patch to the 3D model. As in Figure 1(middle), VDT recovers the detailed shape and texture. However, as can be seen in the figure, the generated image still suffer from problem (ii) because VDT does not change the shape of the 3D model at all. Also, it requires fine adjustment of the input image to the 3D model.

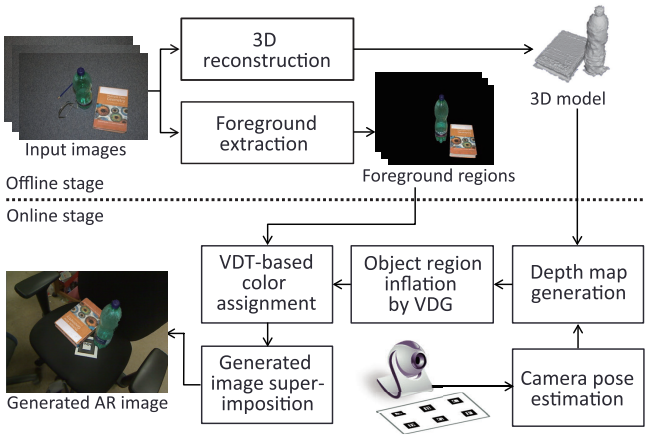


Figure 2: Overview of proposed method.

This paper proposes a method for AR image generation from automatically reconstructed 3D models as in Figure 1(bottom). The proposed method incorporates VDT with view-dependent geometry (VDG) and foreground extraction, where VDG, in this paper, is defined as to modify a depth map of 3D models given a viewpoint so that the depth map suits the viewpoint for generating an image of the virtualized real objects. Our main idea is to inflate the object regions in the depth map by VDG to alleviate the chipped boundaries. Background pixel exposure due to object region inflation is suppressed by ignoring the non-foreground region of the input images when VDT determines each pixel's color. The proposed method is suitable for objects whose images can be taken from anywhere around it. To verify its advantage, we visually demonstrate the quality of generated AR images. We also verify that the proposed method can work in real-time, which is essential property toward AR applications.

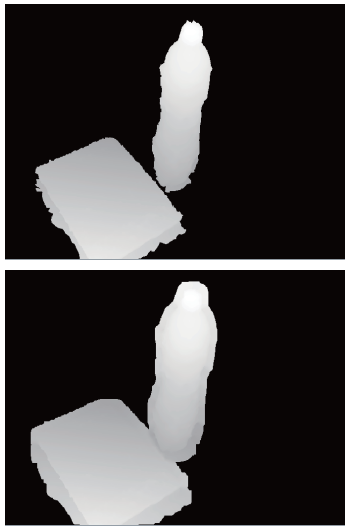


Figure 3: Original (top) and inflated (bottom) depth maps, where inflated one is generated by 20 iterations.

AR Image Generation with VDT and VDG

The proposed method consists of offline and online stages (Figure 2). Given set $S = \{I_1, \ldots, I_K\}$ of input images that contain real objects (referred to as the target), the offline stage reconstructs a 3D model of the target, e.g., using a multi-view stereo (MVS) method [4, 6, 10], and extracts its regions in the input images as foreground. The target in the reconstructed 3D model may be then extracted if it contains extra portion. For foreground extraction, we can use GrabCut [9] to reduce the burden to extract the foreground manually. Although both dense 3D reconstruction and foreground extraction may require manual operations, they are less laborious than hand-crafting 3D models. The online stage firstly estimates the pose of real camera C_{E} from the captured image I_{E} as the viewpoint for the virtualized target, e.g., using ARToolkit [7], and generates the depth map given the 3D model and the pose of C_{E}. Then, since the errors are inevitable even for the state-of-the-art 3D reconstruction techniques, the proposed method inflates the target regions in the depth map to alleviates visual artifacts around the boundaries of the target. We then select a suitable image I_k from S, based on which colors of the corresponding pixels are determined. Finally, we superimpose the generated image of the virtualized target to the captured image. The details of object region inflation by VDG, VDT-based color assignment, and generated image superimposition are given below.

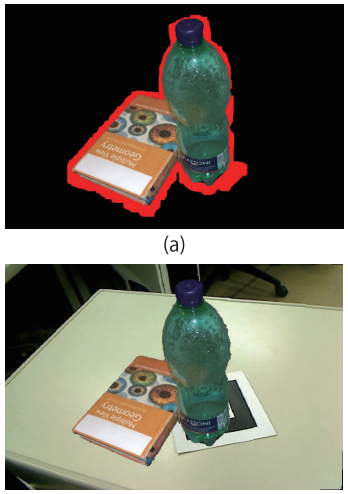
Object region inflation by VDG: To alleviate the problem of chipped boundary, the proposed method inflates the target region in the depth map so that the boundary in the generated image can be formed by the boundaries in the input images. Given the viewpoint, the online stage generates the depth map of the 3D model. Let Ω denote the set of pixels in target region, i.e., the image region onto which the 3D model of the target is projected, and thus the depth values are calculated over Ω. The depth value for pixel i is denoted by d_i. The target region is inflated by applying the following smoothing filter with the 3×3 uniform kernel to the depth map:

$$d'_i = \begin{cases} \dfrac{1}{|\Omega \cap \Phi_i|} \displaystyle\sum_{j \in \Omega \cap \Phi_i} d_j & \text{for } i \in \bar{\Omega} \\ d_i & \text{otherwise} \end{cases}, \qquad (1)$$

where Φ_i is the set of pixels that the kernel centered at pixel i covers. $|\Omega \cap \Phi_i|$ is the number of pixels in set $\Omega \cap \Phi_i$ and $\bar{\Omega}$ is the complementary set of Ω. This keep the inside of target region Ω unchanged so as not to smooth the boundary within the targets. To get sufficient inflation, the filter is applied iteratively. Figure 3 shows examples of an original depth map and an inflated one.

VDT-based color assignment: To reproduce the detail of the target, we assign a color to each pixel in the inflated target region based on the input images considering occlusion.

First, the proposed method finds the image in S that suits for picking the color of pixel i in the inflated target region. Given the intrinsic and extrinsic camera parameters of C_{E}, we can regain 3D position $\mathbf{p}_i$ of pixel i whose depth value in the depth map after inflation is d'_i, where the world coordinate can be set arbitrarily. Let $\mathbf{t}_{\mathrm{E}}$ and $\mathbf{t}_k$ denote the translation of C_{E} and C_k, respectively, where $\mathbf{t}_k$ is the camera that shot k-th image $I_k \in S$. The suitable image is selected so that it can fulfill the following requirements: (i) To reproduce the detail of the target, as described in [3], the angle formed by $\mathbf{t}_{\mathrm{E}} - \mathbf{p}_i$ and $\mathbf{t}_k - \mathbf{p}_i$ must be as small as possible among all k's as long as it fulfills the requirement (ii). (ii) Since $\mathbf{p}_i$ is a point on the 3D model and thus can be occluded in some images in S, $\mathbf{p}_i$ must be visible in the selected image. We can easily find I_k that



(a)

(b)

Figure 4: Examples of (a) generated image and (b) AR image. Red pixels in (a) represent background label.

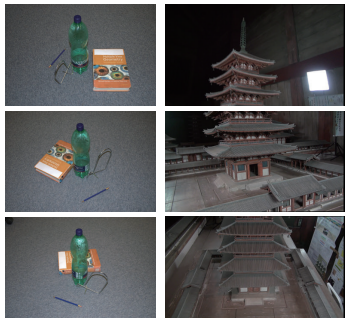


Figure 5: Image datasets for DS1 (left) and DS2 (right).

fulfills the requirement (i) because this is equivalent to find k that gives the largest value of $s_{k,i}$ defined by

$$s_{k,i} = (\mathbf{t}_{\mathrm{E}} - \mathbf{p}_i) \cdot (\mathbf{t}_k - \mathbf{p}_i)/(|\mathbf{t}_{\mathrm{E}} - \mathbf{p}_i|\,|\mathbf{t}_k - \mathbf{p}_i|). \quad (2)$$

To verify the visibility of 3D position $\mathbf{p}_i$, we generate the depth map of the 3D model using the camera parameters of C_k. This depth map generation can be done at the offline stage because the camera parameters and the 3D model are fixed. At the online stage, we project $\mathbf{p}_i$ onto I_k again using the camera parameters of C_k. The depth value corresponding to $\mathbf{p}_i$ is denoted by $d_{k,i}$ and the projected position on I_k is denoted by $(x_{k,i}, y_{k,i})$. Position $(x_{k,i}, y_{k,i})$ identifies the corresponding depth value $e_{k,i}$ in the depth map generated for C_k. If $\mathbf{p}_i$ is occluded, $d_{k,i}$ is further from C_k than $e_{k,i}$. Therefore, we judge that $\mathbf{p}_i$ is visible when $d_{k,i} = e_{k,i}$ and $\mathbf{p}_i$ is in the view frustum of C_k. The latter condition can be verified using C_k's camera parameters.

In order to achieve real-time VDT-based color assignment, we reduce the computational burden by avoiding one-by-one visibility verification as follows: Firstly, we calculate $s_{k,i}$ and verify visibility if $s_{k,i}$ exceeds the current greatest value obtained previously. If $\mathbf{p}_i$ is visible, current greatest value $s_{\hat{k},i}$ is updated to $s_{k,i}$ This process is repeated from $k = 1$ to K to find k^* that gives the greatest value of $s_{\hat{k},i}$.

Finally, we assign a color to pixel i using the foreground regions of I_{k^*}. If the pixel at $(x_{k^*,i}, y_{k^*,i})$ in I_{k^*} is in the foreground regions, its color is assigned to pixel i. Otherwise, a label representing background is assigned to it. An example of a generated image is shown in Figure 4(a), where the pixels to which background labels are assigned are represented by red. This color assignment applied to the inflated depth map alleviates chipped boundaries because the boundaries of the generated image is formed by actual boundaries in the foreground regions of the input images.

Generated view superimposition: Here we generate an AR image by superimposing the generated image on the image I_{E} captured by C_{E}. The pixels in the inflated target region are superimposed on I_{E}, but the pixels with background labels are ignored. An example of the generated image is shown in Figure 4(b).

Experimental results

We experimentally verified the visual quality of the proposed method. Two image datasets, i.e., DS1 and DS2, were used in this experiment as input images (Figure 5), whose specification is summarized in Table 1. To reconstruct the 3D models from these datasets, we used CMPMVS [6], where the intrinsic and extrinsic camera parameters for input images were estimated by Zhang's method [11] and VisualSFM [2], respectively. Since the 3D models originally output from CMPMVS usually contain non-object portions, we manually extracted the target (Figure 6). For foreground extraction, we used GrabCut [9]. The seed region for initializing GrabCut was given manually, and we interactively refined the target regions. For camera pose estimation at the online stage, we adopted ARToolkit [7]. For DS2, we manually selected 59 images from the dataset to reduce the computational cost in VDT-based color assignment. The depth maps were generated from the 3D models using OpenGL.

To demonstrate the superiority of the proposed method, we show generated images by two baseline methods, i.e., (A) a model-based method and (B) a VDT-based method, as well as (C) the proposed method. In the model-based method, the 3D model of the target was rendered with the color assigned to each vertex by the dense 3D reconstruction technique. The VDT-based method applies the original VDT technique to the 3D model, where suitable texture was selected for each triangle of the 3D model.

Table 1: Specification of datasets. Numbers of (i) images and (ii) triangle patches.

	DS1	DS2
(i)	24	571
(ii)	36,051	19,408

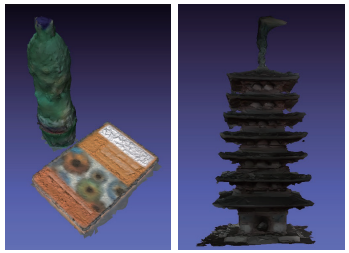


Figure 6: Obtained 3D models for DS1 (left) and DS2 (right).

Figure 7 shows AR images by the baseline and proposed methods. The model-based method gave a blur-like visual artifacts. This is because it did not apply texture to the model and the colors assigned to the vertices were not sufficient to reproduce the detail. Also, the result indicated that the 3D models reconstructed by CMPMVS were inaccurate, producing chipped boundaries.

In the VDT-based method, the blur-like visual artifacts were alleviated owed to the VDT technique. However, the problem of chipped boundaries was still significant. The AR images by the VDT-based method also exposed background pixels due to the inaccurate 3D model.

The proposed method overcame the above problems. Rough boundaries were smoothed by VDG by inflating the target regions, while foreground extraction alleviated excessive background pixel exposure.

However, the proposed method introduced another cause of rough boundaries, i.e., errors in alignment of the 3D models and the input images incorrectly assigned colors in background regions to the target regions, which is noticeable in the red rectangle in Figure 8. This problem is potentially solved by establishing a criterion to find images with large alignment errors and excluding them. Another problem of the proposed method was texture transition. The original proposal of the VDT technique [3] reduces the discontinuity in texture by determining the color based on weighted average of multiple texture images. However, in this work, we omitted this because of its computational cost, and clear discontinuity in texture was observable in the proposed method as in the blue rectangle and its close view in Figure 8.

The frame rates of the model-based, VDT-based, and the proposed methods were summarized in Table 2, where the PC used in this experiment was equipped with Intel Core i7 2600K 3.40 GHz, 8 GBytes RAM, and NVIDIA GeFore GTX 560 Ti, whose OS and Graphic API were Windows 7 64 bit and OpenGL, respectively. Although the frame rates of the VDT-based and the proposed methods were comparable, they were currently not sufficient for the smooth motion. Current implementations of the VDT-based and proposed methods use only CPU for most part of its computation. Using the GPU may improve the frame rate, allowing us to implement smooth texture transition as well.

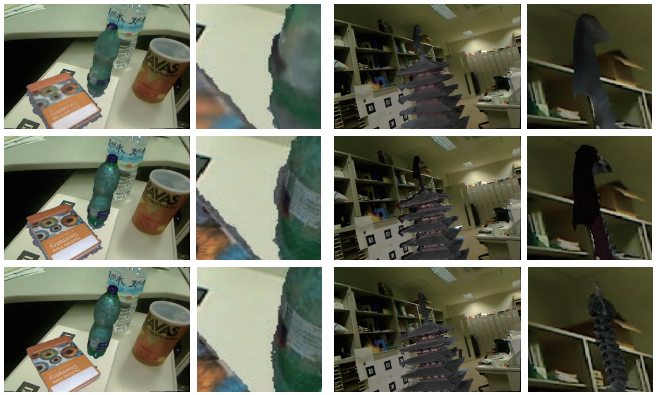


Figure 7: Examples of generated AR images. From top to bottom rows: Model-based method, VDT-based method, and proposed method. From left to right columns: generated AR images for DS1 and their close views, and generated AR images for DS2 and their close views.

Summary

This paper has presented a method for generating AR images with real objects whose 3D models are obtained by dense 3D reconstruction techniques. Usually, 3D models based on such techniques are not very accurate, and thus they produce significant visual artifacts. The pro-

Table 2: Frame rates. (A) model-based, (B) VDT-based, and (C) proposed methods.

	DS1	DS2
(A)	94.48 fps	95.34 fps
(B)	11.18 fps	6.40 fps
(C)	8.32 fps	4.32 fps

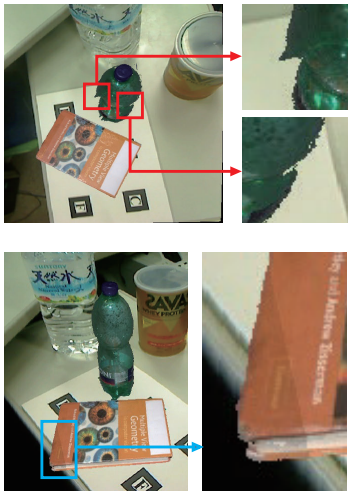


Figure 8: Examples of rough boundaries introduced due to foreground extraction indicated by red rectangles and their close views (top), and discontinuous texture transition indicated by blur rectangle (bottom).

posed method reduces visual artifacts by using the VDG and VDT techniques, where VDT covers the lack of the detailed shape, and VDG smoothes rough boundaries of the 3D models by leveraging extracted foreground. Our experimental results successfully demonstrated the superiority of the proposed method compared with two baseline methods. The proposed method drastically reduces the burden to hand-craft detailed 3D models of real objects and thus potentially encourages AR applications with virtualized real objects. The applications of the proposed method include virtually arranging pieces of furniture and AR sightseeing, in which historical heritages that have been currently lost and preserved only as miniatures are presented as in Figure 9. Our future work is implementing smooth texture transition as well as the use of GPU.

Figure 9: AR sightseeing in a historic site: Example application of proposed method.

Acknowledgements

This work is supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (A) No. 23240024. Some images in this work are by the courtesy of Todaiji Temple, Japan.

References

[1] Augmented Reality Solutions. *VizAR*. http://youreality3d.com/.

[2] Changchang, W. *VisualSFM: A visual structure from motion system*, 2011. http://www.cs.washington.edu/homes/ccwu/vsfm/.

[3] Debevec, P., Yu, Y., and Borshukov, G. Efficient view-dependent image-based rendering with projective texture-mapping. In *Proc. 9th Eurographics Rendering Workshop* (1998), 13 pages.

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

[5] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. The lumigraph. In *Proc. ACM SIGGRAPH '96* (1996), 43–54.

[6] Jancosek, M., and Pajdla, T. Multi-view reconstruction preserving weakly-supported surfaces. In *Proc. 2011 IEEE Conf. Computer Vision and Pattern Recognition* (2011), 3121–3128.

[7] Kato, H., and Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. 2nd IEEE/ACM Int. Workshop on Augmented Reality* (1999), 85–94.

[8] Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., and McMillan, L. Image-based visual hulls. In *Proc. ACM SIGGRAPH '00* (2000), 369–374.

[9] Rother, C., Kolmogorov, V., and Blake, A. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Proc. ACM SIGGRAPH '04* (2004), 309–314.

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

[11] Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence 22*, 11 (2000), 1330–1334.