https://doi.org/10.2352/ISSN.2470-1173.2018.04.SDA-112

A Refocus-Interface for Diminished Reality Work Area Visualization

Momoko Maezawa, Shohei Mori, and Hideo Saito; Keio University; Yokohama, Japan

Abstract

In this paper, we present a refocus interface to set the parameters used for diminished reality (DR)-based work area visualization and a multiview camera-based rendering scheme. The refocus interface allows the user to determine two planes — one for setting a virtual window, through which the user can observe the background occluded by an object, and the other for a background plane, which is used for the subsequent background rendering. The background is rendered considering the geometric and appearance relationships of the multiview cameras observing the scene. Our preliminary results demonstrate that our DR system can visualize the hidden background.

Introduction

Various handheld tools are used for efficient manual operations in fields like woodworking, cooking, and surgery. However, this benefit comes with the drawback of visual occlusion of the working area, as the devices tend to be large due to their driving parts, joints, batteries, and so on. For example, readers may have experienced situations where a gripped electric drill shielded the work target, thereby making direct observation of the work target from the front impossible. One classical solution to this problem is observing the working area from another viewpoint through mirrors or cameras. However, this method requires cumbersome mental viewpoint conversions from the different view to the eye view; in the case of mirrors, mirror image conversion is also necessary. To overcome this problem, computer-aided viewpoint conversions may be substituted for mental viewpoints.

Augmented reality (AR) allows the information of the hidden background to be directly overlaid onto the real environment [1, 2], but the information is described in indirect forms, such as digital annotations. Contrary to this concept of augmentation, visualization technology for making things less noticeable or invisible is called diminished reality (DR) [3, 4, 5]. While our recent DR method achieves high-quality background recovery compared with existing methods, the rendering results have jitters induced by the RGB-D camera [8]. Thus, in this paper, we present a DR system that does not use an RGB-D camera as an alternative for DR-based work area visualization methods.

To achieve a method for DR-based background visualization without an RGB-D camera, here, we address the two following issues: 1) how to determine the region to be removed and recovered (i.e., the region of interest (ROI) and background surface detection); and 2) how to render the background based on the retrieved region information (i.e., background recovery). These are described below.

ROI and Background Surface Detection: For the ROI and background surface detection, many researchers have used automated [3, 6] or semi-automated methods [7]. Compared with such approaches, in our case (i.e., work area visualization), we suppose that the removal region can be placed in a fixed position, just like a fixed loupe on a workers desk. Thus, we place an AR Magic Lens [9] as a virtual loupe through which we can see the background. The position is determined when the user sweeps the lens manually to show the refocus images, using multiview images as guidance (i.e., the refocused image gives a fine focus image when the focal plane is placed at a certain object surface). We also use this approach for estimating the background surface.

Background Recovery: For background recovery, we modify an existing rendering scheme known as unstructured lumigraph rendering (ULR) [10] for our purpose. This image-based rendering method provides an arbitrary viewpoint image by calculating weights for each camera, a camera blending field (CBF), based on the geometric relationships between the calibrated cameras and a geometric proxy, such as a polygon mesh for the scene surfaces. We added an appearance-based term to this formulation to determine whether the camera is observing the background or the occluding object in the working space.

Our contributions regarding these issues in DR can be summarized as follows:

- A Magic Lensinspired refocus interface to set DR-relevant parameters; and
- The geometric- and appearance-based blending field calculation of ULR.

Refocus-based Visualization

Overview

Our system requires the user to set up multiview cameras and parameters related to image-based rendering. Figure 1 shows the system workflow. First, the user must arrange the multiview cameras to capture the work area. Then, the system automatically calibrates the cameras using a well-known computer vision technique. After the procedure, the system is ready for rendering and starts to render the scene. Second, the user changes his/her virtual focal plane, which is rendered in the calibrated multiview camera environment, to find a fine focused depth to set a Magic Lens window, where he/she can see through the hidden space. Third, the user again changes his/her virtual focal plane to set the background plane for visualization via the multiview image-based refocusing. Given the depth information induced by the focal plane set by the user, the system renders the scene without occluding objects in the Magic Lens window.

Setup and Calibration

The system first calibrates multiple cameras to obtain the intrinsic and extrinsic parameters of each camera. First, the system performs feature point extraction and matching between the im-

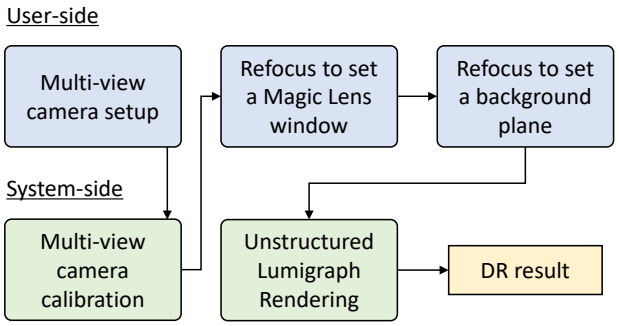


Figure 1. *System workflow*

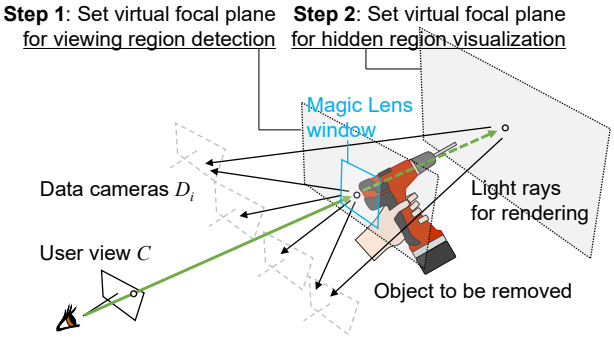


Figure 2. *Refocus user interface for work area visualization*

ages to obtain 2D-2D correspondences in images I^{D_i}. Then, the system performs bundle adjustment [11] with the 2D-2D correspondences.

Refocus-based Parameter Setting

To visualize the work area, we need two types of depth information. The first is for setting the Magic Lens window, and the other is for setting the work area depth (Figure 2).

Setting a Magic Lens Window

Our magic lens is a virtual window floating in a 3D space, into which the user can see through the work area. To set the window, the user must sweep the space along with his/her viewing direction and find a depth giving him/her a fine focus on a certain object (i.e., the object to be removed) while the system is rendering the scene. The sweeping can be controlled using the mouse drag or key binding approach. Once the depth is determined, the user draws a window in an arbitrary shape (e.g., by mouse dragging). Based on this shape, a binary mask image I^M is generated to represent window and non-window pixels in the user's view. The system, therefore, renders the scene only at the window pixels.

Setting the Work Area Depth

The user needs to set the depth representing the work area surface. Here, we assume that the Magic Lens window is small enough that the work area region through the window can be expressed as a plane. Thus, the user sweeps the virtual focal plane again and stops the sweeping when he/she finds a well-focused background on which to set the background plane. We represent the background plane as a set of grid 3D positions $\mathbf{X}_j$ and include this in our rendering procedure, as discussed in the next section.

CBF Calculation

A CBF is a map of weights or ratios to blend M color cameras D_i in a user's view, C (i.e., blending weights of data cameras in [10]) We consider the geometric and appearance features of the multi-view images in our image-based rendering scheme. Thus, we calculate geometry- and appearance-related terms when calculating the blending fields. Algorithm 1 shows the proposed rendering scheme.

Geometric Terms

We introduce the CBF proposed by Buehler *et al.* [10] as geometric terms. The CBF counts for the angular, resolution, and field of view (FoV) weights represented in the following equations.

$$w_{ang} = exp(-(1-\mathbf{d}^C \cdot \mathbf{d}^{D_i})^2/\sigma_{ang}), \tag{1}$$

$$w_{res} = \begin{cases} 1 - L/d(\mathbf{t}^{D_i}, \mathbf{X}_j) \ (L >= 0) \\ 1 \ (otherwise) \end{cases}, \tag{2}$$

$$w_{fov} = R(\psi(D_i, \mathbf{X}_j)), \tag{3}$$

where $\mathbf{d}^C$ and $\mathbf{d}^{D_i}$ are the viewing direction of C and D_i, $\sigma_{ang} = 0.01$, $d(\cdot,\cdot)$ calculates distance between two vectors, $L = d(\mathbf{t}^{D_i}, \mathbf{X}_j) - d(\mathbf{t}^C, \mathbf{X}_j)$ in which $\mathbf{t}^C$ and $\mathbf{t}^C$ are the 3D position of C and D_i, $\psi(\cdot,\cdot)$ calculates the projection to an input camera, and $R(\cdot)$ returns 1 if an input point is within an image space and 0 otherwise.

Finally, given the user-defined balancing parameters, α and β, the following equation gives the weight for $\mathbf{X}_j$ regarding D_i:

$$w_j^{D_i} = w_{fov}(\alpha w_{ang} + \beta w_{res}). \tag{4}$$

Appearance Term

In addition to the geometric terms, we add an appearance-based term to weight certain cameras. The appearance term gives priority to cameras if they share a similar appearance in the FoV assuming most data cameras observe the background rather than the occluding object. Algorithm 2 shows the procedure to calculate the appearance weight w_{app}. In the pseudo code, the function $sim(\cdot,\cdot)$ calculates the similarity of given image patches (e.g., normalized cross-correlation) and $\sigma_{app} = 0.1$. With these weights, we can re-write the weight in equation 4 as follows.

$$w_j^{D_i} = w_{fov}(\alpha w_{ang} + \beta w_{res} + \gamma w_{app}), \tag{5}$$

where γ is a user-defined control parameter like α and β.

Experimental Results

Overview and Setup

Here, we compare the results of PhotoShop's Content-Aware Fill, the synthetic aperture photography (SAP) of Vaish *et al.* [12], and our approach to show the effectiveness of our method. For the comparison, all the methods used the same Magic Lens window.

Algorithm 1: Proposed rendering procedures

$I^{D_i}(\mathbf{x})$**:** Color at $\mathbf{x} \in \mathbb{R}^2$ of C and D_i respectively
$\mathbf{t}^C, \mathbf{d}^C$**:** Position and viewing direction of C
$\mathbf{t}^D, \mathbf{d}^{D_i}$**:** Position and viewing direction of D_i
$\mathbf{X}_j$ **:** j_{th} 3D position of the focal plane
α, β, γ**:** User-defined parameters for balancing weights
$w_j^{D_i}$ **:** Resultant blending weight of D_i for $\mathbf{X}_j$

foreach X_j **do**
  foreach D_i **do**
    $w_{ang} \leftarrow CalcAngularWeight(\mathbf{d}^C, \mathbf{d}^{D_i})$
    $w_{res} \leftarrow CalcResolutionWeight(\mathbf{t}^C, \mathbf{t}^{D_i}, \mathbf{X}_j)$
    $w_{fov} \leftarrow CalcFieldOfViewWeight(\mathbf{X}_j)$
    $w_{app} \leftarrow CalcAppearanceWeight(I^{D_i}, \mathbf{X}_j)$
    $w_j^{D_i} \leftarrow w_{fov}(\alpha w_{ang} + \beta w_{res} + \gamma w_{app})$
  end
  $Sort(w_j^{D_i})$
end
$ULR(\mathbf{M}^C, \mathbf{P}^C, \mathbf{M}^{D_i}, \mathbf{P}^{D_i}, D_i, I^{D_i}, w_j^{D_i})$

Algorithm 2: Appearance weight calculation

$\mathbf{X}_j$**:** j_{th} 3D position of the focal plane
$I^{D_i}(\mathbf{x})$**:** Color at $\mathbf{x} \in \mathbb{R}^2$ of D_i
w_{app} **:** Resultant appearance weight

$w_{app} \leftarrow 0$
$\mathbf{x}_j^{D_i} \leftarrow \psi(D_i, \mathbf{X}_j)$
Create a patch W^{D_i} around $\mathbf{x}_j^{D_i}$
foreach D_k **do**
  $\mathbf{x}_j^{D_k} \leftarrow \psi(D_k, \mathbf{X}_j)$
  if $R(\mathbf{x}_j^{D_k}) = 1$ **then**
    Create a patch W^{D_k} around $\mathbf{x}_j^{D_k}$
    $w_{app} \leftarrow w_{app} + exp(-(1 - sim(W^{D_i}, W^{D_k})^2/\sigma_{app})$
  end
end
$w_{app} \leftarrow w_{app}/M$

We used GoPro 4 Silvers (480*times*640 resolution) to record the frames (Figure 3). The system was implemented using Windows 10, Visual Studio 2015, C++, and OpenGL shading language 3.3. We obtained intrinsic and extrinsic parameters of the user camera C and the data cameras D_i via the bundle adjustment described in the "Setup and Calibration" section. We used a 30×40 planar grid for the geometric proxy and placed in a 3D space as a manually controllable focal plane to perform SAP and our method. We used the four closest cameras for each vertex $\mathbf{X}$ to synthesize a virtual view.

Results and Discussions

Figure 4 shows the results of Photoshop's Content-Aware Fill, SAP, and our method. In these results, a user created a rectangular-shaped Magic Lens window, and then the system visualized the background using each method. As a result, the user can see through the background via the green rectangular window. Although a ghost of the user's fingers is slightly visible in the result of the proposed method, almost the whole hand in the window is invisible, and it is more visible in the other methods. Content-Aware Fill provides a complete removal of the background, but the background is not consistent. On the other hand, we should note that the image quality of the multi-view-based approaches is susceptible to camera calibration errors.

Conclusion

In this paper, we presented an interface representing a combination of the refocus and Magic Lens approaches, and we formulated a CBF using geometric and appearance terms for DR work area visualization. We demonstrated that our DR system can reveal backgrounds occluded by an object. Future work will further discussions via quantitative evaluations.

Acknowledgments

This work was supported in part by a Grant-in-Aid from the Japan Society for the Promotion of Science Fellows Grant Number 16J05114.

References

[1] R. T. Azuma, "Recent advances in augmented reality," *IEEE Comput. Graph. Appl. (CG&A)*, vol. 21, pp. 34–47, 2001.

[2] M. Goto, Y. Uematsu, H. Saito, S. Senda, A. Iketani, "Task support system by displaying instructional video onto AR workspace," Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR), pp. 13–16, 2010.

[3] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, M. A. Otaduy, "Visuo-haptic mixed reality with unobstructed tool-hand integration," *IEEE Trans. Vis. Comput. Graph. (TVCG)*, vol. 19, issue 1, pp. 159–172, 2013.

[4] V. Buchmann, T. Nilsen, M. Billinghurst, "Interaction with partially transparent hands and objects," Proc. Australasian User Interface Conference, vol. 40, pp. 17–20, 2005.

[5] S. Mori, S. Ikeda, H. Saito, "A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects," *IPSJ Trans. on Computer Vision and Applications (CVA)*, vol. 9, no. 17, DOI: 10.1186/s41074-017-0028-1, 2017.

[6] Z. Li, Y. Wang, J. Guo, F. L. Cheong, Z. S. Zhou, "Diminished reality using appearance and 3D geometry of Internet photo collections," Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR), pp. 11–19, 2013.

[7] J. Herling, W. Broll, "High-quality real-time video inpainting with PixMix," *IEEE Trans. Vis. Comput. Graph. (TVCG)*, Vol. 20, pp. 866–879, 2014.

[8] S. Mori, M. Maezawa, H. Saito, "A work area visualization by multi-view camera-based diminished reality," *Trans. on MDPI Multimodal Technologies and Interaction*, vol. 1, issue 3, no. 18, pp. 1–12, DOI: 10.3390/mti1030018, 2017.

[9] D. Baricevic, C. Lee, M. Turk, T. Hollerer, D. A. Bowman, "A hand-held AR magic lens with user-perspective rendering," Proc. Int.

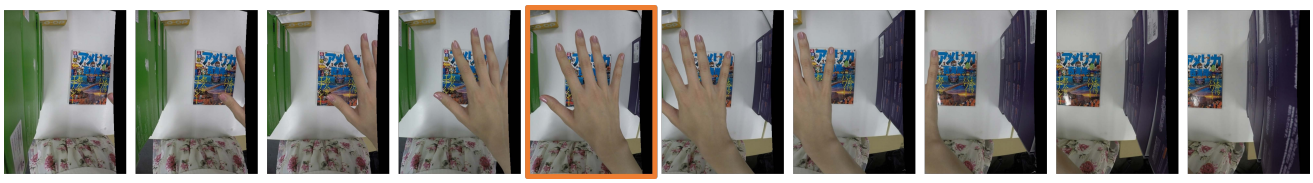
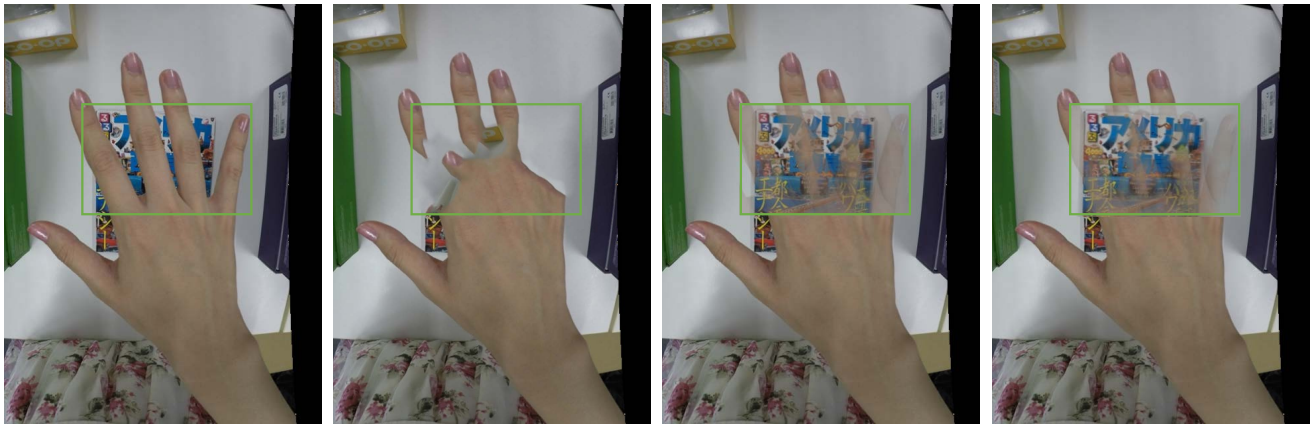


Figure 3. *Input images of the camera array. The image with orange frames is used as the user view.*



(a) User view (b) Photoshop Content-Aware Fill (c) SAP (d) The proposed method

Figure 4. *Resultant images. The green rectangles show the Magic Lens window.*

Symp. on Mixed and Augmented Reality (ISMAR), pp. 197–206, 2012.

[10] C. Buehler, M. Bosse, L. McMillan, S. Gortler, M. Cohen, "Unstructured lumigraph rendering," Proc. the Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 425–432, 2001.

[11] M. I. A. Lourakis, A. A. Argyros, "A software package for generic sparse bundle adjustment," *Trans. Math Softw. (TOMS)*, vol. 36, no. 1, pp. 1–30, 2009.

[12] V. Vaish, B. Wilburn, N., Joshi, M. Levoy, "Using plane + parallax for calibrating dense camera arrays," Proc. Computer Vision and Pattern Recognition (CVPR), pp. 2–9, 2004.

Author Biography

Momoko Maezawa received her B.S. degree in engineering from Keio University, Japan, in 2017. She is currently in a master student at Keio University.

Shohei Mori received his B.S., M.S., and Ph.D. degrees in engineering from Ritsumeikan University, Japan, in 2011, 2013, and 2016, respectively. He was part of the JSPS Research Fellowship for Young Scientists (DC-1) until 2016. He is currently completing a JSPS Research Fellowship for Young Scientists (PD) at Keio University and working as a guest researcher at Graz University of Technology.

Hideo Saito received his Ph.D. degree in electrical engineering from Keio University, Japan, in 1992. Since then, he has been on the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project in the Robotics Institute, Carnegie Mellon University as a visiting researcher. Since 2006, he has been a full professor in the Department of Information and Computer Science, Keio University.