An Instant See-Through Vision System Using a Wide Field-of-View Camera and a 3D-Lidar

Kei Oishi* Keio University
Shohei Mori† Keio University
Hideo Saito‡ Keio University

Abstract

Diminished reality (DR) enables us to see through real objects occluding some areas in our field of view. This interactive display has various applications, such as see-through vision to visualize invisible areas, work area visualization in surgery and landscape simulation. In this paper, we propose two underlying problems in see-through vision, in which hidden areas are observed in real time. First, see-through vision methods require a common area to calibrate every camera in the environment. However, the field of view is limited and many approaches rely on a time-consuming calibration, sensors, or fiducial markers. Second, see-through vision applications assume that the background is planar to ease image alignment. We therefore present a place-and-play see-through vision system using a wide field-of-view RGB-D camera. We validated the accuracy and the robustness of our system and showed results in various environments to show the applicability.

Keywords: Diminished reality, fish-eye camera, 3D-Lidar.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information System—Artificial, augmented, and virtual realities

1 Introduction

Diminished reality (DR) is an active research field whose goal is to remove real objects from the real world visually [5]. This interactive display has various applications, such as work area visualization in surgery and landscape simulation. One of the DR techniques aiming at seeing through remote regions occluded by walls is called see-through vision [3, 1, 7]. The advantage of see-through vision is that it can recover dynamically changing backgrounds based on live video resources from cameras at different viewpoints (e.g., surveillance cameras).

However, the application range is rather limited due to two technical issues. First, see-through vision methods require a common area to calibrate every camera in the environment. However, the field of view (FoV) of each camera is limited, and many approaches rely on a time-consuming calibration using sensor-, feature-, and fiducial marker-based methods. Second, see-through vision applications assume that the background consists of several planes. The conventional techniques approximate backgrounds as several planes to ease camera pose estimation and image warping from the background observer camera to the user camera. While this approximation comes from an assumption that see-through vision is used in urban areas, it limits the application range.

To address these problems, we present a place-and-play see-through vision system using an RGB-D camera with a wide FoV(Figure 2).Our RGB-D camera consists of a fish-eye camera and a 3D-Lidar and enables us to have a common region of view for the user and hidden view observers to be calibrated online. As a result, this system configuration results in a see-through vision without pre-calibration between the environment and the cameras.

Our contributions in this paper include:

- Proposing an instant see-through vision system, which does not require calibration between the environment and cameras except a wide FoV RGB-D camera placement
- Feature matching between the fish-eye background observer camera and the user pinhole camera
- Demonstrating accuracy, robustness, and applicability using real live videos captured in indoor and outdoor 3D scenes

2 Related Work

Kameda *et al.* proposed the use of surveillance cameras calibrated in advance as hidden view observer cameras [3]. Their method visualizes an invisible space due to building by converting images obtained by the observer cameras into user camera images. In the image conversion, they assume that the area to be visualized is a plane or far away to be approximated as a plane. In this method, a CAD model is used to visualize the hidden backgrounds and to estimate user camera pose from observer cameras. However, it is not practical to have a CAD model of the environment and creating it is time-consuming. Barnum *et al.* acquired a common region of view between a user and a hidden view observer camera using an additional camera observing both regions visible from the user and the hidden view observer camera [1]. This method also assumes pre-calibrated cameras and a planar background when transforming background image to the user view. Tsuda *et al.* used a fiducial marker placed on a wall to be diminished to estimate the camera pose [7]. This method generates see-through vision images by rendering acquired 3DCG model of background buildings.

All the methods described above estimate user poses online but assume all background observer cameras and the environment are calibrated in advance. On the other hand, we acquire a common

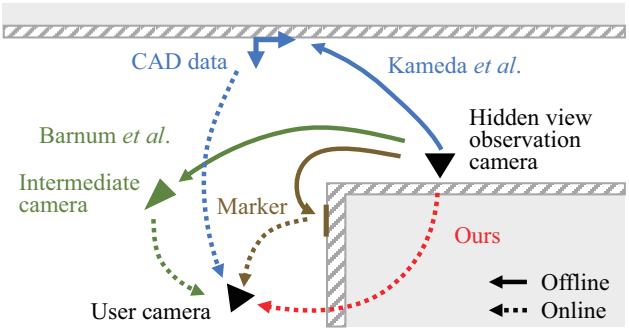

Figure 1: Comparison of the number of coordinate system transformations in the conventional methods and our proposed method. Only our proposed method performs direct correspondence between a user camera and a hidden observer camera.

*e-mail: oishi@hvrl.ics.keio.ac.jp
†e-mail: mori@hvrl.ics.keio.ac.jp
‡e-mail: saito@hvrl.ics.keio.ac.jp

978-0-7695-6327-5/17 $31.00 © 2017 IEEE
DOI 10.1109/ISMAR-Adjunct.2017.99

Figure 2: RGB-D camera composed of a fish-eye camera and a 3D-Lidar

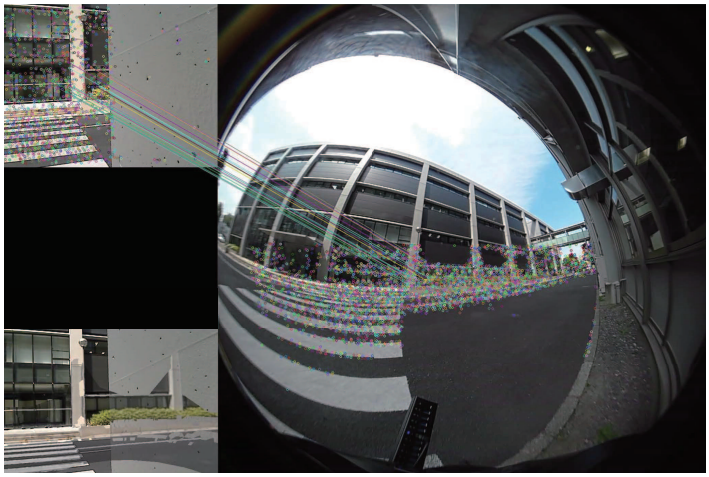

Figure 3: Feature matching between a camera and a fish-eye camera. Left top: User camera input, right: fish-eye camera input, Left bottom: See-through result

field of view between a hidden view observer camera and a user camera using a wide FoV RGB-D camera as an observer camera. In other words, the fish-eye camera enables direct correspondence between the cameras, and 3D-Lidar enables the reconstruction of the 3D background structure. Figure 1 summarizes the number of the online and offline coordinate transformations of the conventional method and our methods.

3 The Proposed Framework

The user camera pose is calculated based on 3D positions of feature points accompanying matches with the user and a hidden view observer camera. However, the difference in appearance between a fish-eye image and a pinhole image due to distortions is too large to match feature points in the images using feature descriptors. Thus, we first convert the fish-eye image to a pinhole camera image (i.e., we simulate a pinhole camera with the same internal parameter as that of the user camera in a celestial coordinate space). Then, we obtain a see-through vision image by overlaying a textured model generated from a point cloud and a fish-eye image to the user view.

3.1 User Camera Pose Estimation

The ideal fish-eye camera has an equidistant projection. In practice, however, actual fish-eye cameras do not follow an ideal projection model. We therefore estimate the internal parameters of a fish-eye camera using a model proposed by Scaramuzza *et al.* [6]. Based on these, we calculate rays, which correspond to each pixel of the fish-eye image. To render a virtual camera image C_v, we set the camera to the origin of the fish-eye camera C_f. Given the rays of each pixel of the fish-eye camera $\mathbf{p}_f = (X_f, Y_f, Z_f)^{\mathrm{T}}$, each pixel of the virtual camera $\mathbf{p}_v$ is calculated as follows.

$$\mathbf{p}_v \approx \mathbf{A}\mathbf{R}\mathbf{p}_f \tag{1}$$

where $\mathbf{A}$ and $\mathbf{R}$ are 3 by 3 intrinsic parameter and rotation matrices associated with the virtual camera, respectively.

We estimate the user camera pose by solving the perspective n-point (PnP) problem with 3D positions of corresponding points between the virtual and the user camera. At this stage, RANSAC is performed to remove outliers and obtain reliable correspondence. Figure 3 shows an example of the resulting feature matches.

3.2 See-through Image Generation

The number of point clouds from a 3D-Lidar is too small to fill in the region of interest in the user view. We therefore generate triangular meshes from a point cloud and fill the mesh with fish-eye image pixels. To remove deformed triangular meshes, we generate meshes only if the length of a side is smaller than a threshold and satisfies the following condition.

$$\mathbf{T} \leftarrow \{t \mid l_t^i < \alpha ||\mathbf{g}_t||, t \in \mathbf{W}\} \tag{2}$$

where $\mathbf{T}$ is the output triangular meshes, α is a user given constant, l_t^i $(i = 0, 1, 2)$ is the length of a side of a triangular mesh t among all triangular meshes $\mathbf{W}$, and $\mathbf{g}_t$ is the center of gravity of a triangular mesh t, respectively.

4 Performance Validation

We validate out method in terms the following three items.

- **Real-time performance** We measured the processing speed throughout the see-through vision process.
- **Robustness** We validated the robustness of our camera pose estimation in an outdoor scenario.
- **Applicability** We applied our see-through vision system in various indoor and outdoor environment.

4.1 Setup

Here, we assume a blind spot visualization scenario. Table 1 summarizes cameras and a 3D-Lidar used in this experiment. We used a Windows 10 PC with an Intel Core i7 5820K 3.3GHz CPU and NVIDIA GeForce GTX 960 GPU. In this experiment, we used the SURF [2] descriptor of GPU implementation for acceleration.

4.2 Real-time Performance

We put a hidden view observer camera behind a building at around 100cm height, as shown in Figure 4 and captured the building wall with the user camera. We selected a 3D scene in which a person is walking in front of a building, which is difficult for the conventional methods to perform see-through vision. Figure 6 shows the results of our see-through vision. These results show that our method can generate see-through images in the 3D scene.

In this experiment, we rendered four virtual viewpoints facing different directions. The average frame rate was 1.76 fps. The most time-consuming process was user camera pose estimation, which accounted for approximately 80% of the total processes. The breakdown of the timing was approximately 52% (approximately 13% per image) for rendering virtual cameras, approximately 18% (approximately 5% per image) for matching, approximately 10% for

Table 1: Device specifications

Device name	Specifications
HOKUYO YVT-X002 (3D-Lidar)	Max 10,360 points, Scanning range: $210^\circ \times 40^\circ$
Kodak PIXPRO SP360 4K (Fish-eye camera)	Image size: 1440×1440, FoV: $235^\circ \times 235^\circ$
Logicool C905 (USB camera)	Image size: 640×480, FoV: $61^\circ \times 46^\circ$

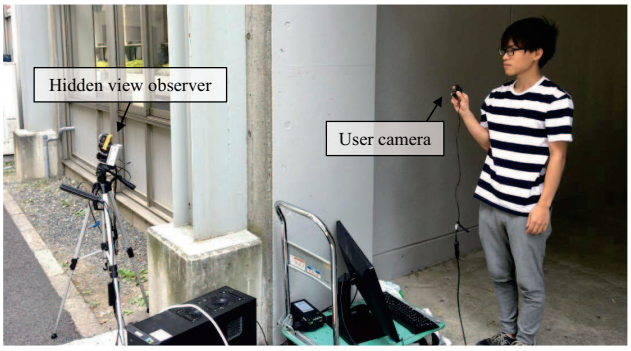

Figure 4: Experimental setup

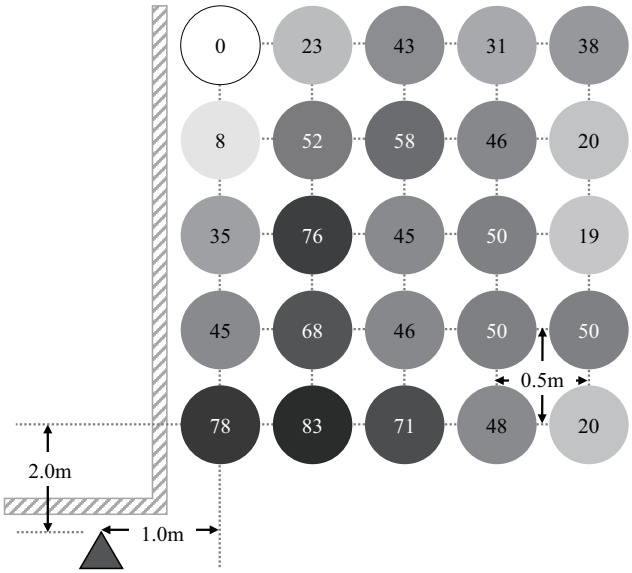

Figure 5: Success rate measurement results of the proposed see-through vision system.

solving the PnP problem. Although the virtual camera rendering process occupies most of the time, we believe that the processing speed would be improved by parallelization techniques, such as GPU implementation (e.g., OpenGL), which is currently implemented on CPU.

4.3 Robustness

We hypothesized that the success rate of our see-through vision image generation drops as the distance from a hidden view observer camera becomes large since we render virtual viewpoints at the projection center of the fish-eye camera. Therefore, we systematically changed the relative position of the user camera and measured the success rate at each point. The success rate was defined as the ratio of the number of frames, in which the hidden background was correctly estimated, within 10 seconds during the execution of see-through vision.

We measured the success rate at each point on the grid of interval 0.5 m in the area of 2 m^2 located 1 m next to and 2 m behind the observer camera. The user camera was directed so that the view is blocked by the wall by 50 % or more. Figure 5 depicts the results of this experiment. The number shown in the circle is the rate of correctly estimated user camera poses at the position (i.e., success rate). As expected, we found that the robustness falls depended on the distance from the observer camera. We observed extremely low scores at positions close to the wall due to large occlusions by the wall impeding correct feature matching. We suggest that placing virtual cameras close to the positions where the success rate is low would increase the matching.

4.4 Applicability

We also performed the proposed method in four other places, including two indoor and two outdoor scenes. Figure 7 shows the results. The left figure shows the input, and the right one shows the see-through images. These results show the applicability of our method. To generate these results, we placed the wide FoV RGB-D camera behind the occluding wall and soon the see-through vision results are obtained. Our preparation step is just simple to place a hidden view observer (i.e., RGB-D camera with a wide FoV). In other words, compared to the conventional methods, our method is highly portable and applicable since it does not require pre-calibration of the background observer camera.

5 Discussion

Limiting the region of interest and matching colors of cameras will improve the quality of our see-through vision since we currently overlay all of the reconstructed raw color pixels onto the user view. We can improve the frame rate as well up to the highest one among the cameras by separating processes for color and depth data via multi-threading, or in a similar manner to [4]. Cameras with different exposure timing will also improve frame rate [8]. One of the extensions of our system will be, therefore, to use multiple cameras as background image resources (e.g. pedestrians' cameras).

6 Conclusion

We presented a see-through vision method using a hidden view observer camera with a wide FoV (i.e., RGB-D camera composed of a fish-eye camera and a 3D-Lidar). To achieve see-through vision using this camera, we proposed an online user camera pose estimation method using feature point matching in a common FoV between the user and the hidden view observer camera. Unlike conventional methods, our method could acquire direct correspondences to the environment and therefore remove the pre-calibration process, which results in the place-and-play see-through vision system. The outdoor and indoor experiments using real data demonstrated the computational performance, robustness of the camera pose estimation, and wide application range of the proposed method. Our future work will include improving the system's performance based on GPU implementation and feature point matching between fish-eye and pinhole images.

Acknowledgements

This work was supported in part by JSPS Grant Numbers 16J05114.

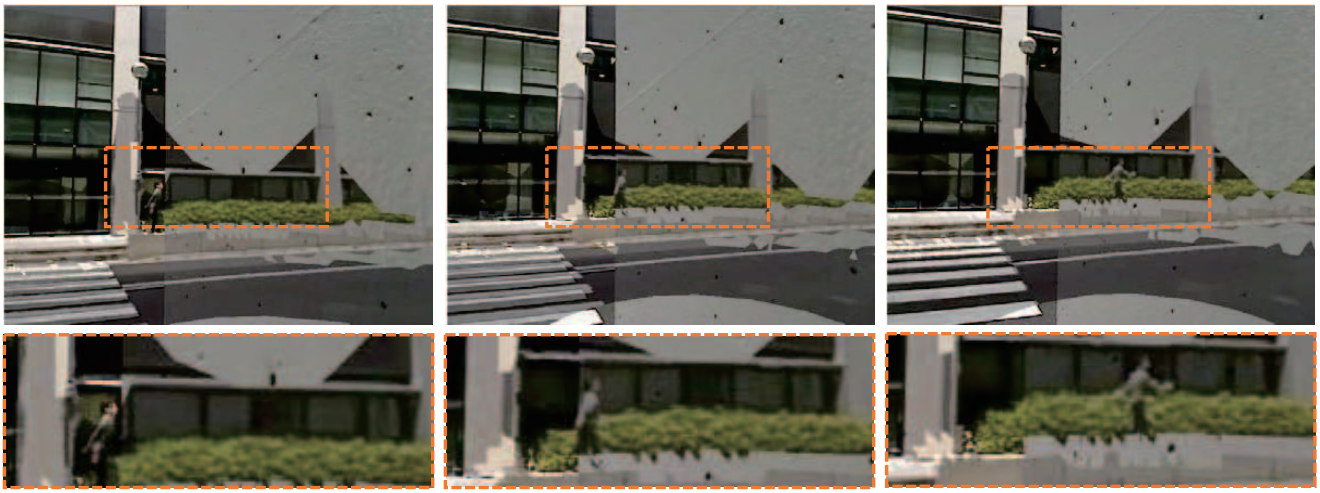

Figure 6: Results of the proposed see-through vision system. Note that a person is walking behind the wall (i.e., the background scene changes dynamically). Bottom row images show the enlarged image of the region enclosed in the orange rectangle in the top row images.

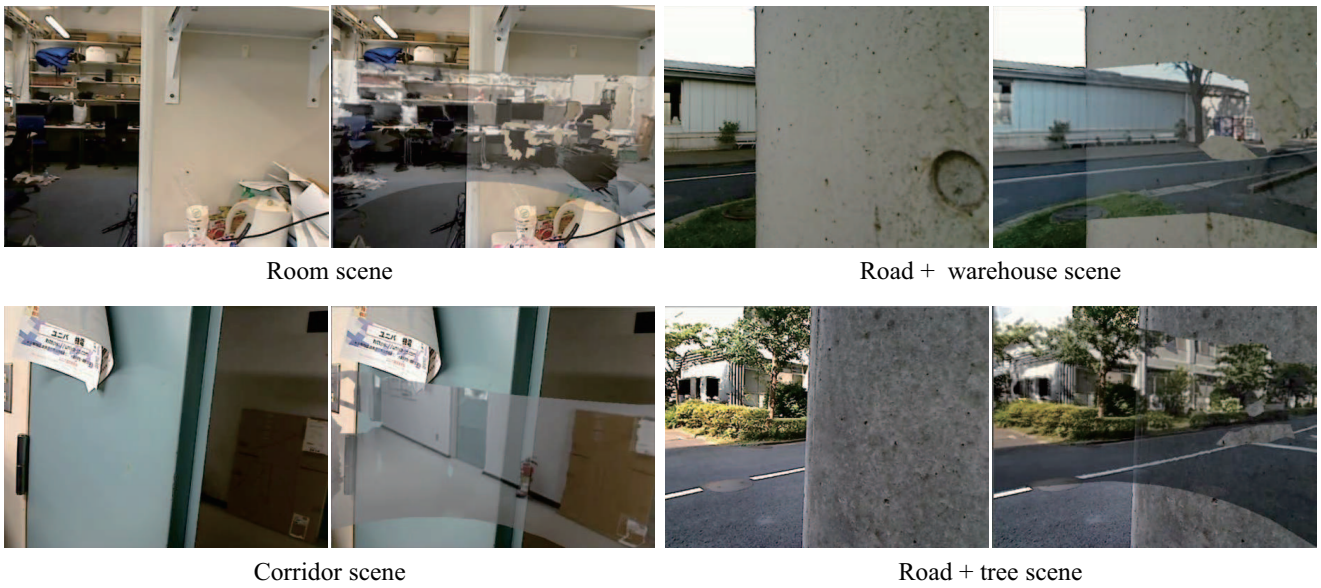

Figure 7: Results in the indoor and outdoor scenes. The left images show indoor scenes, and the right images show outdoor scenes.

REFERENCES

[1] P. Barnum, Y. Sheikh, A. Datta, and T. Kanade. Dynamic seethroughs: Synthesizing hidden views of moving objects. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*, pages 111–114, 2009.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] Y. Kameda, T. Takemasa, and Y. Ohta. Outdoor see-through vision utilizing surveillance cameras. In *Proceedings of IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 151–160, 2004.

[4] J. Lu, H. Benko, and A. D. Wilson. Hybrid hfr depth: Fusing commodity depth and color cameras to achieve high frame rate, low latency depth camera interactions. In *Proceedings of CHI Conference on Human Factors in Computing Systems*.

[5] S. Mori, S. Ikeda, and H. Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 2017.

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, 2006.

[7] T. Tsuda, H. Yamamoto, Y. Kameda, and Y. Ohta. Visualization methods for outdoor see-through vision. *IEICE Transactions on Information and Systems*, 89(6):1781–1789, 2006.

[8] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–294 – II–301, 2004.