

Detour Light Field Rendering for Diminished Reality Using Unstructured Multiple Views

Shohei Mori*
Keio University
Yokohama, Japan

Momoko Maezawa†
Keio University
Yokohama, Japan

Naoto Ienaga‡
Keio University
Yokohama, Japan

Hideo Saito§
Keio University
Yokohama, Japan

ABSTRACT

Instructor's perspective videos are useful for presenting intuitive visual instructions for trainees in medical and industrial settings. In such videos, the instructor's arms often obstruct the trainee's view of the work area. In this article, we present a diminished reality method for visualizing the work area hidden by an instructor's arms by capturing the work area with multiple cameras. To achieve such diminished reality, we propose detour light field rendering (DLFR), in which light rays avoid passing through penalty points set in the unstructured light fields reconstructed from multiple viewpoint images. In DLFR, the camera blending field used in an existing free-viewpoint image generation method known as unstructured lumigraph is re-designed based on our use cases. In this re-design, lesser weights are given to light rays as they pass close to given penalty points. Experimental results demonstrate that using DLFR, the appearance of an undesirable object can be removed from an image in real time.

Index Terms: Diminished Reality, Light Field Rendering, Computational Photography, Multiple Views

1 INTRODUCTION

Instructor's perspective videos and overview videos have the potential to present intuitive visual instructions for trainees in medical and industrial settings. These instruction videos allow trainees to dispense with mental viewpoint conversion, therefore, trainees are able to concentrate on the instructions. However, in these videos, the instructor's body often occludes the work area. To deal with this problem, we propose a diminished reality (DR) method based on light fields [4] to visualize a hidden work area.

DR research aims to visually remove undesirable objects in a user's view by overlaying them with recovered background images. Existing DR methods reconstruct a hidden view using multiple video streams from cameras placed within the environment [10, 6], previously taken photos [5, 7], or image patches in the user's view [3]. We take a similar approach to synthetic aperture photography (SAP) [4, 9]. SAP uses a set of regularly arranged cameras and makes objects invisible by shallow refocusing with a simulated wide aperture or by masking the objects from all cameras.

To relax restrictions of camera placement and masking, we propose detour light field rendering (DLFR). The proposed method is a light field rendering in which light rays detour points in the light fields that we refer to as penalty points. By putting a penalty point on an instructor's hand, we can avoid reconstructing light rays of that area in a synthesized view.

*e-mail:mori@hvrl.ics.keio.ac.jp

†e-mail:momoko.maezawa@hvrl.ics.keio.ac.jp

‡e-mail:ienaga@hvrl.ics.keio.ac.jp

§e-mail:saito@hvrl.ics.keio.ac.jp

2 DETOUR LIGHT FIELD RENDERING

To achieve DLFR, we re-designed the camera blending fields (CBFs) from the method of Buehler *et al.* [1], which is a generalized form of free-viewpoint image generation methods using unstructured cameras based on known scene geometry. The following sections describe the re-designed CBFs and view synthesis using the CBFs.

2.1 Camera Blending Field with Penalty Points

A CBF is a map of blending weights of M data cameras, D_m ($m = 1, 2, \dots, M$), in a virtual view C . In the proposed method, penalty weights are given to data cameras receiving light rays closely passing through penalty points \mathbf{p}_k^O ($k = 1, 2, \dots, K$), that is, light rays that detour the points are weighted instead. We assume that data cameras surround a target work area to remove an occluding object (such as a hand in Fig.1) in a generated virtual view C . Given poses \mathbf{T}^{D_m} and projection matrices \mathbf{P}^{D_m} of data cameras, the blending weights of each data camera to each 3D point \mathbf{p}_i^G ($i = 1, 2, \dots, N$) are calculated as follows.

$$w_{ang} = \theta_i^{C, D_m} \quad (1)$$

$$w_{fov} = \pi(\mathbf{P}^{D_m} \mathbf{T}^{D_m} \tilde{\mathbf{p}}_i^G) \quad (2)$$

$$p_{obj} = \sum_{k=1}^K (1 - \theta_i^{D_m, O_k}) \quad (3)$$

$$w_{i,m} = p_{obj} w_{fov} w_{ang} \quad (4)$$

where, $\pi(\tilde{\mathbf{u}})$ is a function that returns 1 when the 2D point \mathbf{u} exists within the image plane, otherwise 0. θ^{C, D_m} and θ^{D_m, O_k} are calculated as follows.

$$\theta_i^{C, D_m} = \exp(\text{acos}(\mathbf{d}_i^C \cdot \mathbf{d}_i^{D_m}) / \sigma_\alpha) \quad (5)$$

$$\theta_i^{D_m, O_k} = \exp(\text{acos}(\mathbf{d}_i^{D_m} \cdot \mathbf{d}_i^{O_k}) / \sigma_\beta) \quad (6)$$

where \mathbf{d}_i^C , $\mathbf{d}_i^{D_m}$, and $\mathbf{d}_i^{O_k}$ is a position of C , a normalized vector from a virtual camera c , D_m , and \mathbf{p}_k^O to \mathbf{p}_i^G , respectively. σ_α and σ_β are user controllable values.

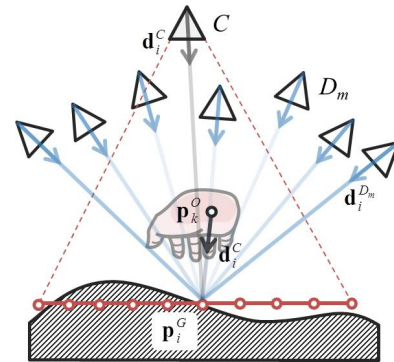


Figure 1: Setup illustration



Figure 2: Comparison between DR results. From left, data camera 5 (D_5), Content-Aware Fill, the proposed method $k = 0$, the proposed method ($k = 1$), and the proposed method ($k = 1$) with RBAB masking

2.2 View Synthesis Without Occluders

3D points \mathbf{p}_i^G make up triangle meshes known as geometric proxies [1]. These triangle meshes can be considered an approximation of a surface of a background or focal plane. Based on the calculated CBF, images of q most highly weighted data cameras are projected onto each mesh using projective texture mapping [8] and are blended with an alpha blending scheme [2].

3 RESULTS AND DISCUSSION

3.1 Setup

We present experimental results to show that an undesirable object can be visually removed in real time using DLFR. In the following evaluation, we used ten data cameras D_m (480×640 resolution) surrounding the work area, a static desktop scene. σ_α and σ_β were both set to 1.0.

3.2 Implementation

We obtained intrinsic and extrinsic parameters of a virtual camera C and data cameras D_m by bundle adjustment (Agisoft PhotoSan) and undistorted the images in advance. The positions of penalty point \mathbf{p}_k^O were manually assigned. Planar geometric proxy \mathbf{p}_i^G ($N = 30 \times 40$) was placed in a 3D space as a manually controllable focal plane. We used the four closest cameras for each \mathbf{p}_i^G to synthesize a virtual view.

The proposed method was implemented on a Windows 10 64-bit laptop with an Intel Core i7 4500U 1.8 GHz CPU, an Intel HD Graphics 4400 GPU, and a 16.0 GB memory. The CBF calculation (Section 2.1) and view synthesis based on the CBF (Section 2.2) were implemented on the CPU and GPU, respectively. The system was implemented using C++ and OpenGL shading language 3.3.

3.3 Results

Figure 2 presents a comparison of results from Photoshop Content-Aware Fill; the proposed method with $k = 0$, which can generate results that are similar to SAP [9]; the proposed method with ($k = 1$); and the proposed method ($k = 1$) with region blending across the border (RBAB) masking [7]. Although a ghost of the fifth finger is partially seen in the result of the proposed method, almost all of the hand is invisible compared to the other methods. However, the recovered view of the proposed method was slightly more blurred than that of SAP because data cameras far from the virtual view were selected for view synthesis. This indicates that the image quality is susceptible to camera calibration errors. The frame rates from the CBF calculation and view synthesis was about 113 fps, thus indicating that the proposed method works in real time.

3.4 Limitations

Object removal becomes challenging when the hand gets close to the background because the background cannot be completely seen by the data cameras. Therefore, in future work, we will consider adding temporal pre-observation of the background to the system, that is, a hybrid approach of background pre-observation [7] and real-time observation.

4 CONCLUSION

In this paper, we proposed a DR framework that uses DLFR to visualize occluded work areas in an instructor’s perspective video. The light field via unstructured multiple views is designed to avoid passing through spaces determined by penalty points. Therefore, an instructor’s hand within the space is invisible in the video. The experimental results show that DLFR is capable of removing the appearance of a hand from an image in real time as well as DLFR’s limitations. In future work, we will develop a DLFR system with automatic hand detection and multi-video streaming.

ACKNOWLEDGEMENTS

This work was supported in part by a Grant-in-Aid from the Japan Society for the Promotion of Science Fellows Grant Number 16J05114 and a Grant-in-Aid for Scientific Research (S) Grant Number 24220004.

REFERENCES

- [1] C. Buehler, *et al.*. Unstructured lumigraph rendering. In *Proc. SIGGRAPH*, pages 425–432, 2001.
- [2] P. Debevec, *et al.*. Efficient viewdependent image-based rendering with projective texture-mapping. In *Proc. Eurographics Rendering Workshop*, pages 85–92, 1998.
- [3] N. Kawai, *et al.*. Diminished reality for ar marker hiding based on image inpainting with reflection of luminance changes. In *ITE Trans. on MTA*, volume 1, pages 343–353, 2013.
- [4] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.
- [5] Z. Li, *et al.*. Diminished reality using appearance and 3d geometry of internet photo collections. In *Proc. ISMAR*, pages 11–19, 2013.
- [6] S. Meerits and H. Saito. Real-time diminished reality for dynamic scenes. In *Proc. IWDR*, pages 53–59, 2015.
- [7] S. Mori, *et al.*. Efficient use of textured 3d model for pre-observation-based diminished reality. In *Proc. IWDR*, pages 32–39, 2015.
- [8] M. Segal, *et al.*. Fast shadows and lighting effects using texture mapping. In *Proc. SIGGRAPH*, pages 294–252, 1992.
- [9] V. Vaish, *et al.*. Using plane + parallax for calibrating dense camera arrays. In *Proc. CVPR*, pages 2–9, 2004.
- [10] S. Zokai, *et al.*. Multiview paraperspective projection model for diminished reality. In *Proc. ISMAR*, pages 217–226, 2003.