Light Field Occluder Removal

Shannon Kao

Motivation

Occluding objects are a common problem in photography. In some cases, scenes may have accidental occlusion, as in dirty lenses or architectural artifacts; in others, occlusion may be unavoidable, as in systems with fixed cameras, such as surveillance networks. Removing such objects, then, is the task of selecting the occluding region, and filling in the scene behind it.

While occluders necessarily represent gaps in information about the scene, there are ways to augment data collection in order to conduct the in-painting of occluded regions. In traditional cameras, photographs from multiple angles provide this information. With the advent of light field cameras, though, it has become possible to remove occluders with the data from a single 4D light field image.

The goal of this project is to remove occluders, which may be of varying shapes and sizes, from a light field image, given that the occluder is at a different depth from the region of interest.

Related Work

Conventional Occluder Removal
Removal of occluding objects is a commonly-researched subject, and there are many methods addressing the problem in conventional cameras.

Favaro [4] presents an algorithm that requires multiple, highly-textured images in order to see beyond occlusions in a scene. Gu [5] provides a method that requires several calibration images to remove image artifacts due to dirty camera lenses. Yamashita [6] targets occluding fences, specifically, using information from a series of images at different focal depths to remove these artifacts.

However, capturing multiple photographs for occlusion removal is a nontrivial task, as it requires a certain level of precision and consistency between photographs. The light field camera, however, records this data in a single image, removing the need for multiple photographs. It would be ideal to leverage this additional data for more accurate, occlusion removal, from a single photograph.

Light Field Occluder Removal
The obvious approach to occluder removal is simply to mask the occluded areas, and use texture synthesis [3] or image inpainting [1] to fill in the unknown regions. Unfortunately, these approaches require a certain density of data to be effective, and relevant information is often scattered across light fields. Simple integration operations fail for the same reasons [2].

Project Overview

Median Filter

Using the method proposed by Yatziv [7], I will first implement a median image-based occlusion removal. The first step of any occlusion removal system is to detect the occluder. This can be done using standard computer vision techniques, resulting in a masked image with holes where the occluder was. This method then computes a median image from the stack of 4D light field data. The median image is used as the initial image estimation, and texture synthesis techniques [3] are applied to fill in the necessary data for each layer in the light field image.

Focal Stack Propagation

The median image method is not optimal for plenoptic cameras, so I will attempt to expand on it using a focal stack propagation algorithm as proposed by Broad [8]. This system similarly defines a mask over the occluded regions, but instead of a median image, calculates an all-in-focus image representing the scene. This all-in-focus region is completed, then the data is propagated back to different layers of the light field image using a Gaussian filter.

Implementation

Goal

Produce a system that, given a light field image with occlusion, can identify the occluder, mask the regions, then fill in the necessary data to produce an un-occluded final image.

Data

I will be using the light field images provided by the class (mentioned in lecture) to test my system.

Milestones

I am hoping to build off the Homework 5 depth estimation code in order to implement this occlusion removal.

1. Implement occlusion detection (1 week)
2. Median image based occlusion removal (1 week)
    a. Generate the median image
    b. Implement texture synthesis based on the median image
3. Focal stack propagation (1 week)
    a. Generate the all-in-focus image
    b. Generate the depth image
    c. Apply texture synthesis to both (a) and (c)
    d. Calculate the appropriate Gaussian filters for each stack, then propagate the synthesized data

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. "Image inpainting," In Proc. SIGGRAPH 2000, New Orleans, July 2000.

[2] A. Duci, A. Yezzi, S. Mitter, and S. Soatto, "Region matching with missing parts," Proc. of the ECCV, LNCS, Springer Verlag, May 2002.

[3] A. Efros and W. Freeman. "Image quilting for texture synthesis and transfer," In Proc. SIGGRAPH 2001, Los Angeles California, August, 2001.

[4] P. Favaro and S. Soatto, "Seeing beyond occlusions (and other marvels of a finite lens aperture)," in CVPR, 2003.

[5] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing Image Artifacts Due to Dirty Camera Lenses and Thin Occluders," Proc. of SIGGRAPH Asia, Dec 2009.

[6] A. Yamashita, A. Matsui, and T. Kaneko, "Fence removal from multi-focus images," in Int'l Conf. on Pattern Recognition, 2010.

[7] L. Yatziv, G. Sapiro, and M. Levoy, "Lightfield completion," in Int'l Conf. on Image Processing, 2004.

[8] T. Broad and M. Grierson, “Light Field Completion Using Focal Stack Propagation,” in Proc. of SIGGRAPH, Jul 2016.