15-663 Project Report – Coded Aperture

Jingguo Liang
jingguol@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

ABSTRACT

In this project, I will explore and implement the coded aperture technique proposed by Levin et al. [2] and discussed in the Coded Photography lecture. This technique makes a simple modification to the conventional camera, which is the insertion of a patterned occluder into the aperature of the camera lens, which we call *coded aperture*. And it enables us to recover depth information and a high-resolution image from a single photo taken with the coded aperture. The depth information recovered may be further utilized on features such as image re-focusing or 3D reconstruction.

1 INTRODUCTION

During the semester, we have either been exposed to or implemented many computational photography techniques that enables us to extract depth information of the scene from the image(s) taken. However, most of them are either impractical or very difficult to perform on commercial cameras.

Hasinoff and Kutulakos introduced the technique Confocal Stereo in their article [1], which recovers 3D shapes through photos taken with different focal lengths and aperture sizes. While this technique produces results in very fine resolutions, the procedure requires that the scene and the camera stays static during the whole process, making it very difficult to perform on outdoor scenes, or with casual shooting scenarios with commercial cameras.

We are also introduced to light fields photography [3], with which we can generate images with new camera positions and focal lengths by interpolating an array of photos taken by a lens array. It is clear that it is very difficult to use this technique without the presence of dedicated equipment, either a lens array or a plenoptic camera. Furthermore, the output renderings from light fields only have a limited resolution, eliminating it as a choice when a high-quality photo is desired.

There are other techniques mentioned in the lectures, such as depth from focus and focal flow. All those techniques either require multiple input photos, or a specific piece of equipment that is not commonly used in normal photo shooting. Coded aperture, on the other hand, is able to generate depth information and a high-resolution all-in-focus image simultaneously, without needing multiple shots or special equipment, making it a solid choice for common commercial cameras.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 METHOD

In this section, I will make a brief introduction to the method proposed by Levin et al. [2]. Although my implementation is not identical to the one described in the article, they share the same main idea and the article serves as a good baseline for my implementation.

2.1 Notation

I will follow the convention of notation used by Levin et al. in this report, where lower case symbols are used to represent signals in the spatial domain and upper case symbols are used to represent signals in the frequency domain.

2.2 Imaging Model

Consider the thin lens model shown in figure 1. For a point located on the focal plane, which has a distance d from the lens, all the rays emanating from this point passing through the lens will be focused on a single point on the sensor plane. If the point moves to a location off the focal plane, which has distance d_k from the lens, the light rays emanating from it will no longer focus to a single point on the focal plane. Instead, they will be mapped to a region on the sensor plane, which is known as the *circle of confusion*. This results in off-focus objects appearing to be blurred in the photo taken.

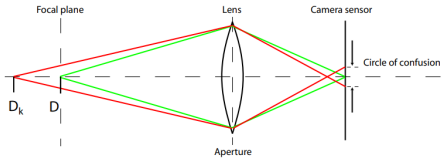


Figure 1: Thin lens model

Image credit: Levin et al.

We can model this process using a convolution. For objects at distance d_k, the image formed y will be

$$y = f_k * x$$

where x is the in-focus, sharp image, f_k is the convolution kernel corresponding to the aperture at this distance, and $*$ denotes convolution.

2.3 Gaussian Prior of Images

Though affected by noises from various sources, real-world images have statistics very different from random noises. I will only briefly introduce the prior assumption and its implications made by Levin et al., without going deeply into the maths and derivations.

From the observation that real-world images have a sparse derivative distribution, Levin et al. makes the prior assumption that the derivatives in a sharp image will follow a Gaussian distribution with a zero mean.

$$P(x) \propto \prod_{i,j} e^{-\frac{1}{2}\alpha((x(i,j)-x(i,j+1))^2+(x(i,j)-x(i+1,j))^2)} = \mathcal{N}(0,\Psi)$$

We also assume that noise in neighboring pixels are independent and follows a Gaussian distribution $n \sim \mathcal{N}(0,\eta^2 I)$, where eta is a implementation specific constant. Through some derivations, we have $P_k(y)$ also following a Gaussian distribution: $P_k(y) \sim \mathcal{N}(0,\Sigma_k)$, where Σ_k is the covariance matrix. In the frequency domain, the distribution of the blurry image y becomes:

$$P_k(Y) \propto \exp\left(-\frac{1}{2}E_k(Y)\right) = \exp\left(-\frac{1}{2}\sum_{v,\omega}|Y(v,\omega)|^2/\sigma(v,\omega)\right)$$

where $\sigma(v,\omega)$ are diagonal entries of $\bar{\Sigma_k}$, which is Σ_k in the frequency domain.

2.4 Filter Selection

We then use the Kullback-Leibler divergence to evaluate how good a filter is.

$$D_{KL}(P_{k_1}(y), P_{k_2}(y)) = \int_y P_{k_1}(y)(\log P_{k_1}(y) - \log P_{k_2}(y))dy$$

Intuitively, we want the image distributions $P_{k_1}(y)$ at distance k_1 and $P_{k_2}(y)$ at distance k_2 to be as different as possible, so that we could differentiate between the distances by observing the image. Distributions that maximizes the Kullback-Leibler divergence above will satisfy the criteria. It will be blurry at k_1 with a high probability and blurry at k_2 with a low probability.

Transforming this measurement into frequency domain, we have

$$D_{KL}(P_{k_1}, P_{k_2}) = \sigma_{v,\omega}\left(\frac{\sigma_{k_1}(v,\omega)}{\sigma_{k_2}(v,\omega)} - \log\left(\frac{\sigma_{k_1}(v,\omega)}{\sigma_{k_2}(v,\omega)}\right)\right)$$

which implies that the differences between two distributions will be large when the ratio of their frequencies is large.

With this measurement, we are able to search for good filter patterns from all the available ones. There are only several more limitations here. First, the pattern must be binary, since we do not have translucent material for the occluder. Second, it must be connected, so that pieces will not be hanging in the middle. And finally, use only the center region of the aperture to avoid radial distortion as much as possible.

Levin et al. conducted a random search over the 13 patterns with 1mm holes. They found out that symmetrical patterns generally performs better than asymmetrical ones on the KL divergence, and they usually have fewer zeros in their frequency domains. They finally chosen the pattern shown in figure 2, which is also the pattern that I impelmented.

2.5 Deblurring

With the Gaussian prior assumption

$$P_k(x|y) \propto \exp\left(-\left(\frac{1}{\eta^2}\left|C_{f_k}x-y\right|^2 + \alpha\left|C_{g_x}x\right|^2 + \alpha\left|C_{g_y}x\right|^2\right)\right)$$

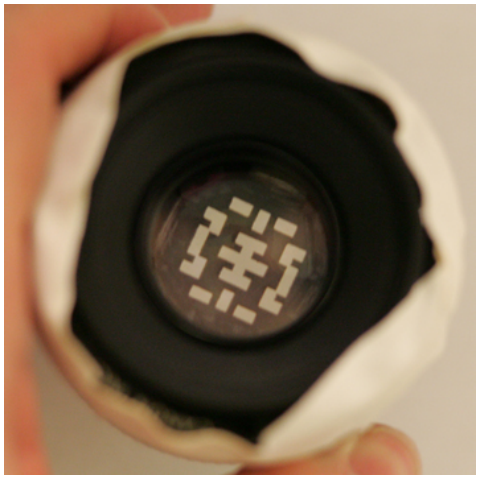


Figure 2: Occluder pattern

Image credit: Levin et al.

where α is an implementation-specific constant regularizing the smoothness of the image, and C_f is the matrix such that $C_f x = f * x$. Then, we have

$$\begin{aligned} x^* &= \text{argmax } P_k(x|y) \\ &= \text{argmin } \frac{1}{\eta^2}\left|C_{f_k}x-y\right|^2 + \alpha\left|C_{g_x}x\right|^2 + \alpha\left|C_{g_y}x\right|^2 \end{aligned}$$

which is a least square optimization problem that can be solved in closed form.

The Gaussian prior that we have tends to over-smooth the result. Levin et al. proposed another image prior to make the result sharper, which I will not discuss it here.

2.6 Handling Depth Variation

The previous step outputs a deblurred, sharp image provided that every pixel in the image has the same depth, such that the whole image can be deblurred for a single blurring kernel. However, in absolute majority of scenearios pixels in an image will have different depths. With a series of blurring kernels, which are the coded aperture from different distances, we can compute the reconstruction error

$$e = \frac{1}{\eta^2}\left|C_{f_k}x^* - y\right|$$

where x^* is the deblurred image using kernel f_k. Pixels deblurred and re-blurred using a filter inconsistent with its true depth will have a high reconstruction error in its surrounding area. Thus, for every pixel, we can sum over the reconstruction errors in its neighborhood to obtain its energy E:

$$E(y(i)) = \sum_{j \in W_i} e_k(j)^2$$

where W_i denotes a window around pixel i. And we can determine the approximate correct depth d by

$$d(i) = \text{argmin}_k\ \lambda_k E_k(y(i))$$

where λ_k are coefficients learnt from training data. With the depth map, we can reconstruct an all-in-focus image simply by taking pixels from deblurred images corresponding to pixel depths.

3 IMPLEMENTATION, RESULT, AND DISCUSSION

In this section, I will demonstrate my work following approximately the structure of the previous section, discuss how my implementation differs from that in the article by Levin et al., and discuss the significance of the results.

3.1 Lens and Occluder

Figure 3 shows the coded aperture that I implemented. I have implemented the coded aperture occluder with a pattern resembling the one presented by Levin et al., which is the pattern they found out that maximizes the KL distance. The lens I used for the coded aperture is a Canon EF 50mm f/1.8 II lens, borrowed from Yannis. I was able to disassemble the lens and insert the occluder to the aperture. The aperature is approximately 21mm in diameter, and I used the 11mm × 11mm area in the center for the pattern, to avoid radial distortion.

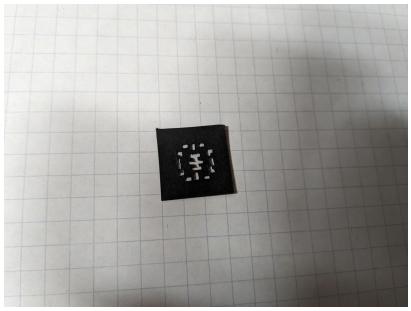
In assignment 1, where we used black paper for pinhole cameras, the image quality suffered from the low quality of the cuts, especially the coarse edges of the holes. Fibers of the paper remained on the edges after the cut, partially occluding the pinhole and introducing diffraction. I switched to cardstock paper this time, which is harder and thicker, and used finer art knives for the cutting. While most of the coarse edges vanish, a few still exists on the intersections of two holes. Additionally, the shape of the pattern is less regular than the implementation by Levin et al., which is likely to made by 3D printing or laser cutting.

Unfortunately, the lens does not attach to the camera because they are of different brands. For the remaining parts of the implementation, I have to use the data provided by Levin et al. together with their article.

3.2 Kernel Calibration

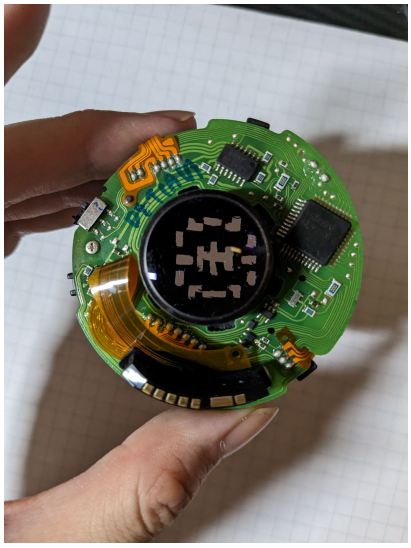
Before we proceed to the remaining parts, there is some important data that we must acquire, which is the blurring kernel for the coded aperture, at different distances. This should be done by calibrating the coded aperture through the following method proposed by Levin et al.

Make a drawing of some random curves, as shown in figure 4, and hang it to the wall. Without inserting the occluder, fix the camera at some distance from the drawing (approximately 2m), adjust and fix the focal length and aperture such that the camera creates a sharp image. Then, insert the occluder. Without changing the focal length and aperture settings, incrementally move the camera away from the drawing, taking a photo of it every step. Then, for each photo taken with the coded aperture, select a region in both the sharp image without occluder and the blurred image such that their content will align (note that I have the four markers on the edges to help on this). Then, using an appropriate filter size (21 is usually a good choice), solve the over-constraint linear system, where the sharp image convolved with the filter equals the blurred image.



(a)

(b)



(c)

Figure 3: Coded Lens Implementation

(a) the occluder; (b) the disassembled lens; (c) aperture with the inserted occluder

As my lens failed to attach to the camera, this step had to be skipped. I instead used the filter data provided by Levin et al, which correspond to their example photos.

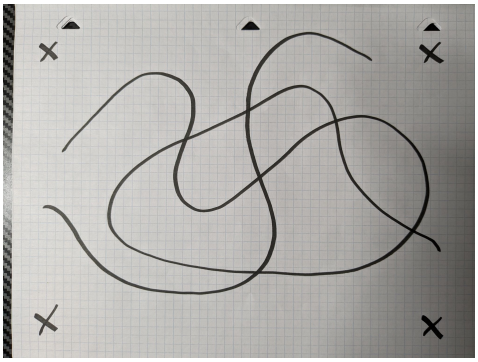


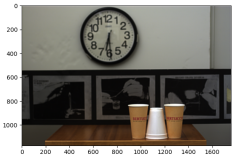
Figure 4: Curve drawing for calibration

3.3 Delurring

Instead of using the blurring method introduced by Levin et al., which involves solving a linear system, I used the Wiener deconvolution method discussed in the lecture. After some experimentation, I have chosen the reciprocal of the signal to noise ratio $\frac{1}{SNR} = 0.01$, for the best result. Additionally, I also tried the Richardson-Lucy deconvolution that is introduced in Levin et al., which they found to produce less satisfactory results than other deconvolution methods.

Figure 5 shows examples of the deconvolved images from the two deconvolution methods. We can observe strong ring artifacts from the image with Wiener deconvolution, which is expected when the deconvolution filter does not match to the true filter when the image was taken. The only thing observed in the Richardson-Lucy deconvoluted image is a blurrier version of the sharp image, which renders it useless for the following steps. However, it is to be noted that while Richardson-Lucy deconvolution is an iterative algorithm, I only run it for 3 iterations, as it takes excessive amount to time to run. It might yield better results on more iterations, but the running time will be long.

(a) Wiener deconvolution



(b) Richardson-Lucy deconvolution

Figure 5: Deconvolved images

3.4 Depth Map and All-in-focus Image

Figure 6 shows several groups of images taken from the coded aperture, the recovered depth map, and the recovered all-in-focus image. We observe that the results are not the most satisfactory. While part of the depth information is correctly extracted, it appears that a large portion of the pixels are categorized to a same depth in the middle. And the incorrectly evaluated depths result in artifacts in the recovered all-in-focused image, which are essentially part of the ringing artifacts created by deblurring with incorrectly scaled kernels.

Figure 6: Result

For each group, images are listed in the order of: (1) original photo taken with coded aperture; (2) recovered all-in-focus image; (3) recovered depth map.

There are two possible reasons for the artifacts. The most probable one may be the coefficients λ_k in the equation for depth estimation

$$d(i) = \text{argmin}_k \ \lambda_k E_k(y(i))$$

Levin et al. state that those coefficients are learnt through a maching learning method of training on a set of training images with known depth profiles. They are trained such that the scale misclassification

error is minimized. As I do not have the training data to obtain their accurate values, I naively initialized them all as 1, giving them equal weights in the depth estimation.

The second potential problem is the deconvolution algorithm, which is the only remaining difference of my implementation from the implementation by Levin et al. As I use Wiener deconvolution, I discard the Gaussian prior which has a regularization term in it that tries to keep the image smooth. However, I do not think this is the main reason for the artifacts, since Wiener deconvolution also handles noises, and I was able to produce meaningful deconvoluted images.

4 FUTURE WORK

The depth map and all-in-focus images extracted from coded aperture photos allows us to extend many features upon it. The depth-map enables us to perform a 3D reconstruction of the scene, only taking one image with a slightly modified commercial camera. We can also create re-focused images by applying corresponding blur filters to chosen pixels based on the depth map.

There are also limitations to this technique. The most prominent limitation is the quality or resolution of the recovered depth map. Depending on the window size when we are estimating reconstruction error, the recovered depth map may either be very noisy or has a very low resolution. Furthermore, as mentioned by Levin et al., occasionally human input is required to accurately recover the depth map. Another problem is the coded aperture calibration. The calibration must be performed per lens per pattern per focal distance and per aperture, which is a heavy burden on manufacturers, if this technique is to be commercialized.

REFERENCES

[1] Samuel W. Hasinoff and Kiriakos N. Kutulakos. 2008. Confocal stereo. *International Journal of Computer Vision* 81, 1 (2008), 82–104. https://doi.org/10.1007/s11263-008-0164-2

[2] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. 2007. Image and Depth from a Conventional Camera with a Coded Aperture. *ACM Trans. Graph.* 26, 3 (jul 2007), 70–es. https://doi.org/10.1145/1276377.1276464

[3] Marc Levoy and Pat Hanrahan. 2023. *Light Field Rendering* (1 ed.). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3596711.3596759