Assignment 2: Coded Apertures
Lab session 3

1 Introduction

In this assignment, you will experiment and familiarize with the use of coded apertures, a class of camera filters that aim to recover images that are not captured perfectly in focus, either because the objects are outside of the focal distance [3, 1] or suffer from motion blur [2]. As seen in class, the capture of an image I' can be modeled as the convolution of the point spread function (PSF) k of the camera lens and the perfectly focused image I, plus a noise term σ (Figure 1):

$$\underbrace{I'}_{\text{Capture}} = \underbrace{k}_{\text{Lens PSF}} \star \underbrace{I}_{\text{Image}} + \underbrace{\sigma}_{\text{Noise}} \tag{1}$$

The blur on the image depends on two factors: i) the distance D of the object with respect to the focal plane distance d, and ii) the type of mask and the diameter of the aperture, which determines the shape of the PSF.

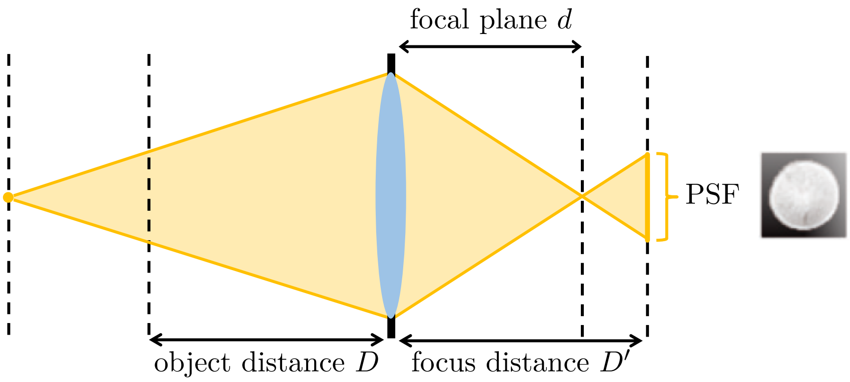

Figure 1: Image capture model.

We have seen that spatial convolutions can be expressed as simple multiplications in the frequency domain. With $\mathcal{F}$ being the Fourier transform, the image capture process becomes:

$$\mathcal{F}[I'] = \mathcal{F}[k] \cdot \mathcal{F}[I] + \mathcal{F}[\sigma] \tag{2}$$

Starting with the blurred image and knowing the PSF of the camera, it is possible to recover the original image by deconvolution:

$$I = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[I'] - \mathcal{F}[\sigma]}{\mathcal{F}[k]}\right] \tag{3}$$

However, this inversion method is not always perfect. Some information can be irrecoverable after the convolution with the PSF, whenever it has zero or close to zero values for some frequencies, resulting in detail loss and ringing artifacts on the recovered image. Coded apertures try to address these problems by avoiding singularities in the frequency domain, as we will see in this session.

2 Blur simulation and recover of focused image

Select one image included in the `images` folder and simulate the blurring and focusing of the image using a circular aperture with the code in `deblurring_demo.m`. In the code, identify how the defocusing process is implemented using convolution and how the focused image is recovered by inverting the process (Figure 2 shows an example).

Focused Defocused Recovered

Figure 2: Example of an image captured with a circular aperture (left), defocused (middle), and refocused (right).

Once you have familiarized yourself with the code, perform the following experiments and briefly describe them in your report:

1. Select different noise levels (`sigma`) and blur sizes (`blurSize`), and analyze their impact on the image you have selected.

2. The code in `deblurring_demo.m` uses a Wiener deconvolution with a natural image prior, incorporating knowledge about natural images into the deconvolution with a frequency prior. Test other deconvolutions such as Lucy (`deconvlucy` in MATLAB) o Wiener without priors (`deconvwnr` in MATLAB). Experiment with different number of iterations or the noise levels. You can look up these functions' parameters in MATLAB's online documentation or using `help deconvlucy` and `help deconvwnr`.

3 Power spectra

As you can see in Section 2, the recovered images using a circular aperture loose considerable detail and they are full of ringing artifacts. Visualize the input images and the apertures in frequency domain using `power_spect_demo.m` and explain how this is affecting the reconstruction (hint: check whether all the frequencies are preserved after computing the convolution).

4 Coded apertures

Coded apertures are designed to recover focused images losing the least possible amount on information in the convolution and deconvolution processes. You can see the difference comparing the results in Figure 2 using a circular aperture and the results shown in Figure 3 using a coded aperture.

Repeat the tasks described in Section 2 and Section 3 using the apertures introduced by Zhou et al. [3] and Raskar et al. [2]. Analyze how the power spectra of these apertures behaves compared with the circular aperture.

Focused | Defocused | Recovered

Figure 3: Example of an image processed with Zhou et al.'s coded aperture [3].

5 Color images

So far, we have worked with gray-scale images. Modify the code to work on color images by computing each color channel independently. An example is shown in Figure 4.

Focused | Defocused | Recovered

Figure 4: Example of a color image processed with Raskar et al.'s coded aperture [2].

6 Deliverable

Your deliverable should be an archive file (e.g., a ZIP file) with the following items:

1. All your MATLAB code, including commented code, implementing the different sections of this assignment, as well as a README file explaining how to use the code. Comment your design choices in the code, if any.

2. A PDF report of up-to two pages(plus figures) with all the intermediate and final results obtained, the explanations on how you got your results, and why you decided to use certain parameters or methods and not others.

7 Evaluation

The maximum score for this assignment is 10 points:

Blur simulation (Section 2)	3
Power spectra (Section 3)	2
Coded apertures (Section 4)	4
Color images (Section 5)	1
Total	10

8 Acknowledgments

The material in this assignment has been adapted from the public code published by Zhou et al. [3] as supplementary data to their work.

References

[1] Belen Masia et al. "Perceptually-Optimized Coded Apertures for Defocus Deblurring". In: *Computer Graphics Forum* 31.6 (2012).

[2] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. "Coded Exposure Photography: Motion Deblurring Using Fluttered Shutter". In: *ACM Transactions on Graphics* 25.3 (2006). DOI: 10.1145/1141911.1141957.

[3] Changyin Zhou, Steve Lin, and Shree Nayar. "Coded aperture pairs for depth from defocus". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009. URL: https://www.microsoft.com/en-us/research/publication/coded-aperture-pairs-depth-defocus/.