TEL AVIV UNIVERSITY

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING

The Zandman-Slaner Graduate School of Engineering

PHASE CODED APERTURE FOR EXTENDED DEPTH OF FIELD IMAGING AND DEPTH MEASUREMENT CAPABILITIES FROM A SINGLE IMAGE

By

Harel Haim

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY

in partial fulfillment of the requirements for the degree of

"DOCTOR OF PHILOSOPHY"

November 2017

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
The Zandman-Slaner Graduate School of Engineering

PHASE CODED APERTURE FOR EXTENDED DEPTH OF FIELD IMAGING AND DEPTH MEASUREMENT CAPABILITIES FROM A SINGLE IMAGE

By

Harel Haim

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY
in partial fulfillment of the requirements for the degree of
"DOCTOR OF PHILOSOPHY"

Under the supervision of Prof. Emanuel Marom and Prof. Alex Bronstein

November 2017

Acknowledgments

First and foremost, I would like to thank my two advisors Professor Emanuel Marom and Professor Alex Bronstein. The two of you provided me the guidance I needed while supporting me to pursue my own ideas.

Emanuel, you were my advisor and mentor since I first started my master degree and you are one of the main reasons I decided to leave behind my career in Hi-Tec and follow my true dream of becoming a scientist. Thank you for all your hard work, devotion and for always being available to discuss and advise. Your knowledge, expertise, and creativity alongside your impeccable work ethic, is inspiring.

As Emanuel provided the optical aspect of my 'computational photography' studies, Alex facilitated the second half, with his infinite knowledge in computational processing. Alex, thank you for pushing me to learn and investigate new fields which allowed me to expand my studies even more and opened my mind to new possibilities. Your innovative mindset and firsthand experience with commercial applications provided an important perspective and encouraged me to pursue projects with more impact.

Special thanks also to my associate and friend Shay Elmalem who has been working beside me those last years and dedicated countless hours helping me build most of the prototype cameras I used in my research. I had the privilege of co-writing my last paper in my PhD studies with Shay, which allowed me to enter the exciting field of deep learning. I wish to thank Dr. Raja Giryes who co-authored this latest paper. I very much enjoyed our collaboration and I want to thank you Raja for your guidance, insight and support. I also wish to thank several fellow graduate students, who I had the pleasure working beside them on the FPGA project: Tal Remez, Or Litany and Shachar Yoseff.

Finally, I would like to thank my loving wife and best friend Netta. Thank you for encouraging me to follow my dream. You knew exactly when I needed to be pushed and when to relax, gave me advice when I needed one and, most important, for believing in me in every step of the way. Your love and support made this PhD possible.

Abstract

The advent of focal plane arrays revolutionized the field of photography. The availability of smartphones, first appearing on the market in the last decade, made everyone a “photographer”, and as for 2017, in USA there are roughly more smartphone cameras then there are people. Digital image quality is determined by the optics, the focal plane array sensor and the processing stage. For smartphone, the limited available volume allocated to the lens makes it very difficult to improve the image quality by optical solutions. As both conventional optics and sensor technologies have reached their peak, most of the advancements in that field have shifted to the domain of image processing.

The field of Computational Photography (CP) has grown fast those last years, attracting the attention of top technological companies such as Google, Apple and Samsung. In CP, one manipulates the image acquisition stage to allow an efficient post processing stage for image processing and computer vision applications. In the scope of this dissertation, a computational camera design, aimed specifically for small-scale camera implementation, was investigated. Such camera provides extended depth-of-field (EDOF) imaging, and allows estimating a depth map from a single frame. Those features can be used for functions such as refocusing and 3D modeling, and can be incorporated in applications such as augmented reality and autonomous vehicles.

One of the most challenging issues in imaging systems is the restoration of out-of-focus (OOF) images. The problem is notoriously ill-posed, since information is lost in the process. A symmetric binary phase mask offers a low-cost optical solution for increasing the camera's depth-of-field (DOF), providing acceptable quality for machine vision applications such as barcode reading and face detection. An RGB phase mask, whereby one acquires unique responses for the red, green and blue channels, resulting in a simultaneous acquisition of three perfectly registered images, each with a different out-of-focus characteristic has been thoroughly investigated in this thesis.

A major part of this dissertation is focused on methods for fusing the three RGB channels into a single-color image with extended DOF and improved appearance, via a specially tailored efficient post-processing algorithm based on sparse representation model. The computational stage has also been implemented on an FPGA module, thus providing an end-to-end, real-time imaging system that can blindly handle scenes that contain objects at different distances from the camera, making it ideal for natural every-day scenes.

In the last part of this dissertation, two methods for estimating a depth map using a single frame will be presented. Since the RGB mask provides depth dependent color response, those color cues can be utilized for estimation the focus setting for each pixel in the image, which can be easily translated into absolute metrical values. The first method is based on the sparse model that was used for EDOF imaging. The second method is focused on fast implementation of depth map estimation using convolutional neural network (CNN). The presented simulation and experimental results demonstrate real-time performance with accuracy depth estimation.

Table of Contents

List of Acronyms and Abbreviations

ATF – Amplitude Transfer Function

AWGN – Additive White Gaussian Noise

BN – Batch Normalization

CM – Confidence Map

CNN – Convolutional Neural Networks

CONV – Convolutional

CP – Computational Photography

CRF – Continuous Random Field

CS – Compressive Sensing

DCT – Discrete Cosine Transform

DFD – Depth from Defocus

DFF – Depth from Focus

DLIM – Diffraction Limited

DOF – Depth-Of-Field

DOFO – Depth-Of-Focus

DTD – Describable Textures Dataset

EDOF – Extended Depth-of-Field

FCN – Fully Connected Networks

FISTA – Fast Iterative Shrinkage Thresholding Algorithm

FPGA – Field-Programmable Gate Array

ISP – Image Signal Processor

ISTA – Iterative Shrinkage Thresholding Algorithm

LASSO – Least Absolute Shrinkage and Selection Operator

MAD – Mean Absolute Difference

MAP – Maximum a Posteriori

MAPE – Mean Absolute Percentage Error

MP – Matching Pursuit

MRF – Markov Random Field

MSE – Mean Square Error

MTF – Modulation Transfer Function

OMP – Orthogonal Matching Pursuit

OOF – Out-Of-Focus

OTF – Optical Transfer Function

PSF – Point Spread Function

PSFDE – Point Spread Function Derivative Energy

PSNR – Peak Signal-to-Noise Ratio

ReLU – Rectified Linear Unit

SGD – Stochastic Gradient Descent

SSIM – Structural Similarity

SVD – Singular Value Decomposition

List of Figures

List of Tables

1. Contribution and thesis outline

This dissertation is organized as followed:

Chapter 2 is dedicated to background review. The first section (2.1) presents the basic concepts in optics that were used in the scope of this dissertation. Section 2.2 reviews the related studies in the field of computational photography, and Section 2.3 reviews passive and active depth acquisition methods. The following sections deal with the fundamentals of the processing methods that were used in the course of this research: sparse representation and dictionary learning is presented in Section 2.4 and Section 2.5 presents the basic concept in deep learning using convolutional neural network.

Chapters 3-5 present the methodologies used in the scope of the research, each contributing to its respective field.

Chapter 3 describes the optical aspect of this research. Section 3.1 presents the optical properties of a binary phase mask, in particular, depth dependent phase mask (Section 3.2). The main contribution provided in this chapter is presented in Sections 3.3 and 3.4. The mask optimization process (3.3) is used for designing a depth sensitive phase mask for various optical configurations, as well as for compensating for spherical aberrations (3.4).

Chapter 4 is dedicated for the EDOF imaging system computational process using sparse models. First, a non-blind deblurring method (4.2) is described as a reference to the novel blind method presented in Section 4.3. The experimental stage of real-life scenes is presented in 4.4. The end-to-end, real-time imaging system is presented in 4.5 with the utilization of the FPGA module. The last contribution in this chapter is dedicated to depth estimation using a sparse model (4.6) which, when combined with the all-in-focus image that our EDOF system produces, enables capabilities such as image refocusing.

Chapter 5 takes the depth estimation one step forward with the introduction of CNN model in our phase mask optical system. The CNN utilized the color cues exhibited in the image captured with an RGB mask, and generates state-of-the-art performance in real-time computational time. The two main contributions provided in this chapter are the integration of mask design within the CNN training process (Section 5.3) alongside real-time performance with accurate metrical results for both simulated and real life images (Sections 5.4 and 5.5).

2. Background

2.1 Basic concepts of optical imaging

2.1.1 Ray optics

Ray optics, or geometrical optics describes light as rays, travelling in optical media. Although ray optics does not predict many optical phenomena, it is very useful for describing and designing imaging systems.

The basic imaging model describes light traveling from an object plane located at a distance d_{obj} from a lens with a focal length f, to an image plane located at a distance of d_{img} from the lens plane as illustrated in Fig. 1. Perfect imaging refers to a scenario where all light rays exiting form a point at the object plane are conformed into a single point at the image plane. In this scenario, both image plane and the object plane support the well know 'thin lens equation' [1]:

$$\frac{1}{d_{obj}}+\frac{1}{d_{img}}=\frac{1}{f}. \tag{1}$$

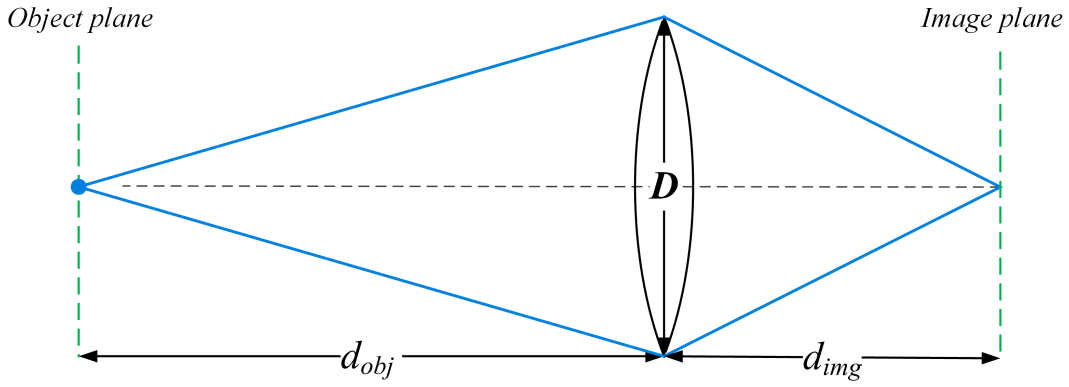


Fig. 1: Perfect imaging. The object and image distance from the pupil satisfies the 'thin lens equation'.

When the imaging condition is satisfied, the image obtained by geometrical optics is the scaled version of the object:

$$u_{img}(x,y)=u_{obj}(x/M,y/M) \tag{2}$$

where M, the lateral magnification, is given by:

$$M=-\frac{d_{img}}{d_{obj}} \tag{3}$$

In the scope of this section (unless noted differently), a paraxial approximation of a thin lens model is used whereby lens aberrations are neglected.

When the image and object planes do not satisfy the condition in Eq. (1), the resulting output image will be blurred. In ray optics, imaging is considered still 'in-focus' when a point in the object plane results in a blurred spot smaller than a pixel size when detected by a pixelated detector array. This concept defines two important terms: Depth-of-Focus (DOFO) and Depth-of-Field (DOF). Consider a point object located at a distance u from the lens plane. The ideal image plane that satisfies Eq. (1) is at a distance v as illustrated in Fig. 2. The term DOFO defines the range whereby the image plane can be shifted by Δv while the spot size is still within the width of a pixel. Using simple geometric considerations, DOFO can be defined as:

$$\Delta v = \frac{p \cdot v}{D} \tag{4}$$

where p is the pixel width and D is the lens diameter.

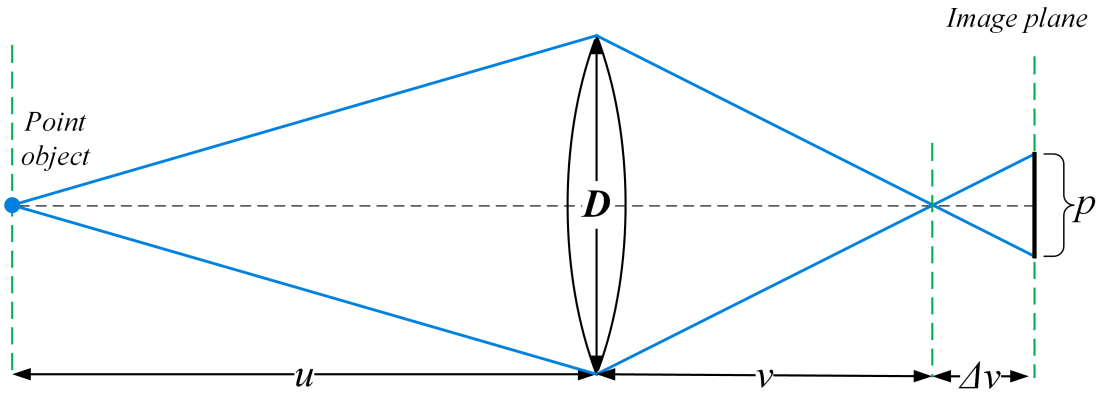


Fig. 2: Depth-of-focus. The range whereby the image plane can be shifted by Δv while the spot size is still within the width of a pixel.

The term DOF refers to a scenario in which the image plane is fixed at a certain distance v corresponding to a nominal object distance u as illustrated in Fig. 3. The DOF defines a depth range whereby the object distance may vary without loss of resolution in the image plane, so that all objects stay in focus. Using Eq. (1) the derivative $\frac{dv}{du}$ can approximate as:

$$\frac{\Delta \mathrm{v}}{\Delta \mathrm{u}} \approx \frac{d\mathrm{v}}{d\mathrm{u}} = \frac{-f^2}{(\mathrm{u}-f)^2} \xrightarrow{\mathrm{u}>>f} -\frac{f^2}{\mathrm{u}^2} \tag{5}$$

By rearranging this approximation, using Eq.(3) and Eq. (4), the DOF can be expressed as:

$$|\Delta \mathrm{u}| = \left(\frac{\mathrm{u}}{f}\right)^2 \Delta \mathrm{v} = \left(\frac{\mathrm{u}}{f}\right)^2 \frac{\mathrm{v}}{D} p \xrightarrow{\mathrm{v}\approx f} \left(\frac{\mathrm{u}}{\mathrm{v}}\right)^2 \frac{f}{D} p = \boxed{\frac{1}{M^2} \cdot F_{\#} \cdot p} \tag{6}$$

where $F_{\#} = f / D$ is the well know f-number parameter which provides the light collection ability of the camera. As the aperture diameter increases more light is collected at the expense of the DOF.

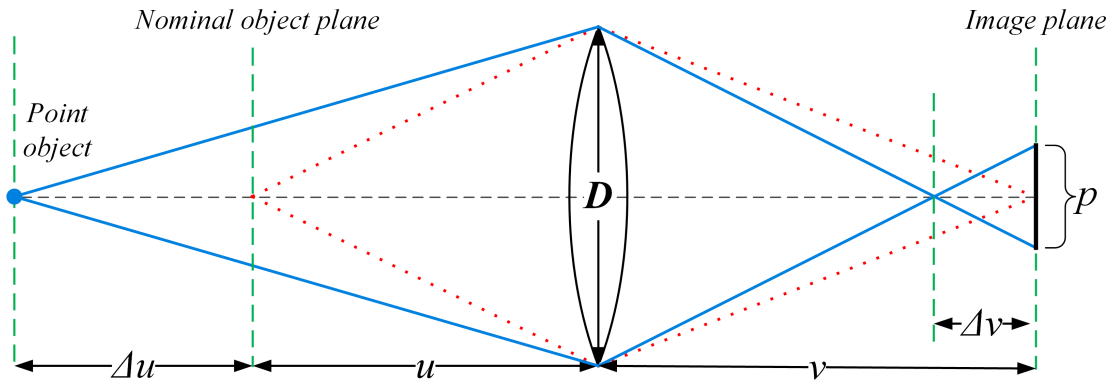


Fig. 3: Depth-of-field. The range whereby the object distance may vary while the spot size is still within the width of a pixel.

Another useful term in DOF evaluation is the hyperfocal distance. Consider a case where the image plane is located at the focal plane, meaning perfect imaging for an object at infinity; the hyperfocal distance is the minimum object distance which will still be in focus in terms of a single pixel spot size, as illustrated in Fig. 4. The hyperfocal distance u_{hyp} is defined as:

$$u_{hyp} = \frac{f \cdot D}{p} \tag{7}$$

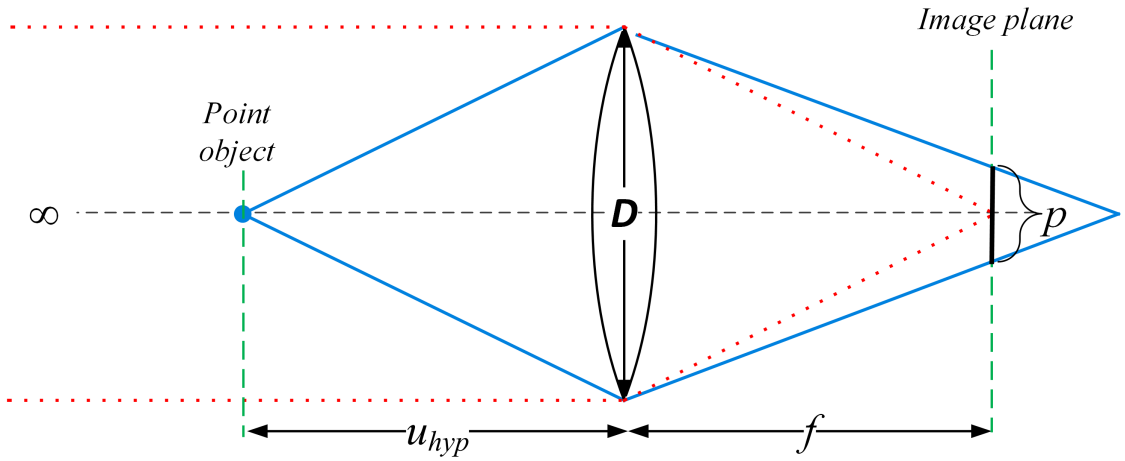


Fig. 4: Hyperfocal distance. The distance of which a point object image spot size is less than a pixel, as the image plane is set to the lens focal length.

2.1.2 Frequency analysis of optical imaging system

Ray optics cannot describe that perfect imaging of light rays coming out of a point source with 'aberration-free' lens, do not produce a perfect point on the image plane, but rather a blurred spot, due to Diffraction limitations. On the other hand, the scalar model of wave optics predicts this phenomenon and is very useful for evaluating imaging systems performance.

In this study the imaging system is described as a single "black box" element with a certain transfer function. For simplicity, we will assume the imaging system to consist of a single lens with a focal length of f as shown in Fig. 5. The object plane, lens plane and image plane coordinates are (ξ,η), (x,y) and (u,v) respectively; z_o is the distance from the object to the lens and z_i is the distance from the lens to the image plane.

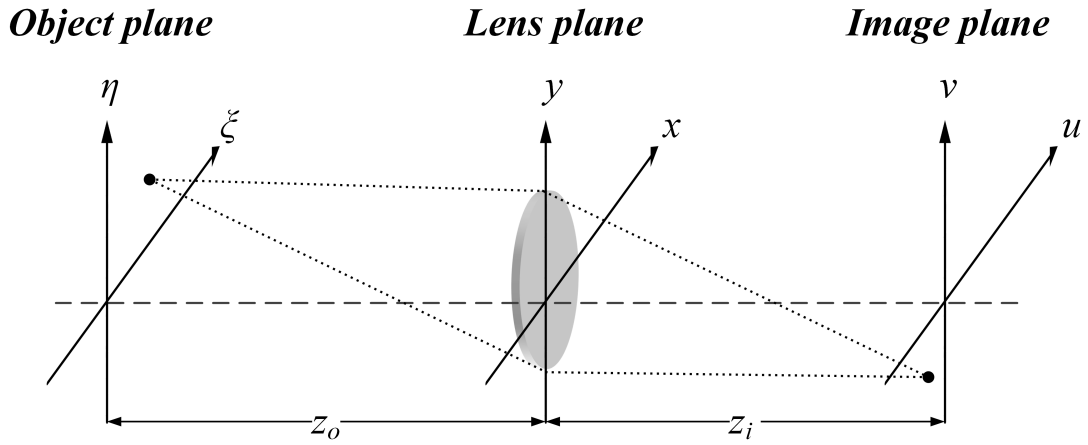


Fig. 5: Basic imaging system using a thin lens.

Coherent imaging exists in a controlled environment usually using lasers as a light source. Coherent illumination treats monochromatic illumination, harmonically time dependent, whereby the wave front phase at an arbitrary point is the phase of all the points in that wave front. For such case, the system is linear in complex amplitude:

$$U_i(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_c(u-M\xi;v-M\eta)U_o(M\xi,M\eta)\,d\xi d\eta \tag{8}$$

where $U_i(u,v)$ is image amplitude, $U_o(\xi,\eta)$ is the object amplitude distribution, $M=-(z_i/z_o)$ is the magnification factor and h_c is the amplitude transfer function (ATF) of the imaging system, whereby the lens aperture function is $P(x,y)$:

$$h_c(\xi,\eta;\lambda)=\frac{A}{\lambda z_i}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x,y)\exp\left\{-j\frac{2\pi}{\lambda z_i}(\xi x+\eta y)\right\}dxdy \tag{9}$$

"Incoherent illumination" means spatially incoherent quasi-monochromatic illumination, harmonically time dependent, in which the phase of an arbitrary point is uncorrelated to the phase at any other point. This type of illumination is most common in natural scenes and therefore camera performance is evaluated under this regime. Imaging systems for such case are linear with respect to the intensity and not with respect to the field amplitude, as in the case of coherent illumination.

For incoherent illumination, the output image I_i can be expressed by [1] :

$$I_i(u,v) = \kappa \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h\left(u-\tilde{\xi};v-\tilde{\eta}\right) I_g\left(\tilde{\xi},\tilde{\eta}\right) d\tilde{\xi}d\tilde{\eta} \tag{10}$$

where:

$$I_g\left(\tilde{\xi},\tilde{\eta}\right) = I_o\left(\frac{\tilde{\xi}}{M},\frac{\tilde{\eta}}{M}\right) \qquad ;\left(\tilde{\xi},\tilde{\eta}\right) = \left(M\xi, M\eta\right) \tag{11}$$

is the geometrical-optics ideal image intensity, κ is a real constant and h is the imaging system intensity point spread function (PSF):

$$h\left(\xi,\eta;\lambda\right) = \left|h_c\left(\xi,\eta;\lambda\right)\right|^2 \tag{12}$$

The properties and limitations of an imaging system, that is linear with respect to the intensity, can be analyzed in the spatial frequency domain. Imaging system performance can be best analyzed using the Optical transfer function (OTF). For incoherent imaging systems, the OTF describing the system response in terms of spatial frequencies, is the normalized Fourier transform of the intensity impulse response:

$$OTF\left(f_x,f_y\right) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(u,v)\exp\left(-j2\pi\left(f_x u + f_y v\right)\right) dudv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(u,v)\, dudv} \tag{13}$$

It can be shown, using Eq. (9), that the OTF is the normalized autocorrelation of the system pupil $P(x,y)$:

$$OTF\left(f_x,f_y\right) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P\left(x+\frac{\lambda z_i f_x}{2}, y+\frac{\lambda z_i f_y}{2}\right)\cdot P^*\left(x-\frac{\lambda z_i f_x}{2}, y-\frac{\lambda z_i f_y}{2}\right) dxdy}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|P(x,y)\right|^2 dxdy} \tag{14}$$

In the special case of an ideal circular pupil with no aberrations, the OTF is:

$$OTF(f) = \begin{cases} \frac{2}{\pi}\left[\arccos\left(\frac{f}{f_c}\right) - \frac{f}{f_c}\cdot\sqrt{1-\left(\frac{f}{f_c}\right)^2}\right] & f \le f_c \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $f = \sqrt{f_x^2 + f_y^2}$ and f_c is the cutoff frequency of the incoherent illuminated system:

$$f_c = \frac{D}{\lambda z_i} \quad (16)$$

with D as the pupil diameter.

The Modulation transfer function (MTF) is the absolute value of the OTF and is one of the most common ways for describing imaging system characteristics. For a given single frequency target, such as a sinusoidal pattern that span from zero to one, the MTF provides the attenuation factor of that sinusoidal pattern. It is equivalent to the contrast:

$$Contrast = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \quad (17)$$

where $I_{\max}, I_{\min}$ are the maximum and minimum values of the output sinusoidal as illustrated in Fig. 6.

In the special case of an ideal circular pupil as mentioned above, all frequencies which are equal or higher than the maximum spatial frequency f_c are not transmitted. Since they are above the cut-off limit, their contrast is zero.

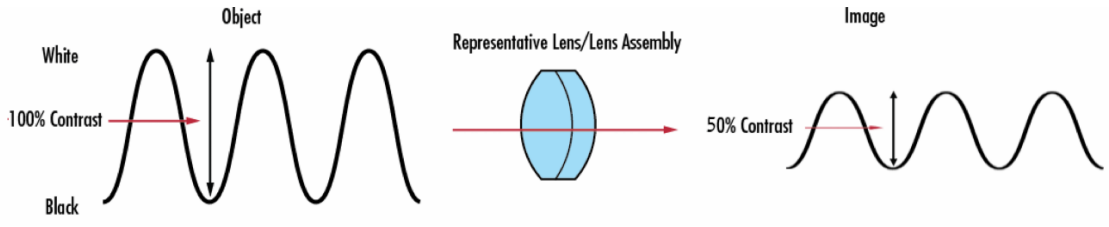


Fig. 6: Attenuation of contrast level for a sinusoidal input.

2.1.3 Out-of-focus aberration effects on image quality

As stated in Eq. (9) the PSF is proportional to the Fourier transform of the lens aperture function *P* only if the imaging condition is satisfied:

$$\frac{1}{z_0} + \frac{1}{z_i} - \frac{1}{f} = 0 \quad (18)$$

When this condition is not satisfied, the imaging system will suffer from out-of-focus (OOF) aberration which degrades the image quality, meaning lower contrast level and loss of information.

The out-of-focus error can be analytically described as a wavefront error or a phase error in the pupil plane [1]. This phase error is expressed by the addition of a quadratic phase term in the aperture of an otherwise ideal imaging system. In the presence of out-of-focus aberration, the generalized pupil can thus be expressed as:

$$\tilde{P}(x,y) \quad = \quad P(x,y)\cdot\exp\left[j\frac{\pi}{\lambda}\left(\frac{1}{z_0}+\frac{1}{z_{img}}-\frac{1}{f}\right)\left(x^2+y^2\right)\right] \tag{19}$$

where z_{img} is the detector position when the object is in nominal position z_n; z_o is the actual object position. Clearly when in nominal position the bracketed term is null; this is the in-focus condition. In case of a circular aperture with radius *R* we define a defocus parameter Ψ as:

$$\Psi = \frac{\pi R^2}{\lambda}\left(\frac{1}{z_0}+\frac{1}{z_{img}}-\frac{1}{f}\right)=\frac{\pi R^2}{\lambda}\left(\frac{1}{z_{img}}-\frac{1}{z_i}\right)=\frac{\pi R^2}{\lambda}\left(\frac{1}{z_o}-\frac{1}{z_n}\right) \tag{20}$$

The generalized pupil in this case will be:

$$P_{\Psi}(x,y) \quad = \quad \tilde{P}(x,y)=P(x,y)\cdot\exp\left[j\Psi\cdot\left(\frac{x^2+y^2}{R^2}\right)\right] \tag{21}$$

The defocus parameter value Ψ denotes the maximum phase error at the aperture edge. For $\Psi > 1$ the image will suffer from contrast loss; for $\Psi > 4$ it will experience severe information loss and even reversal of contrast for some frequencies as exhibited by the MTF curves in Fig. 7. The contrast reversal is demonstrated in Fig. 8 using a spoke target. As we go closer to the center of the target the contrast is reduced until it reaches the point where the black lines turn to white and vice versa.

For a circular aperture, the diffraction limit maximum spatial frequency (or the cut-off frequency) is $f_c = \frac{2R}{\lambda z_i}$; the resolution of the optical system increases as the aperture radius increases. At the same time, the DOF decreases as the defocus parameter Ψ increases, per Eq. (20), thus reducing the resolution of the OOF objects.

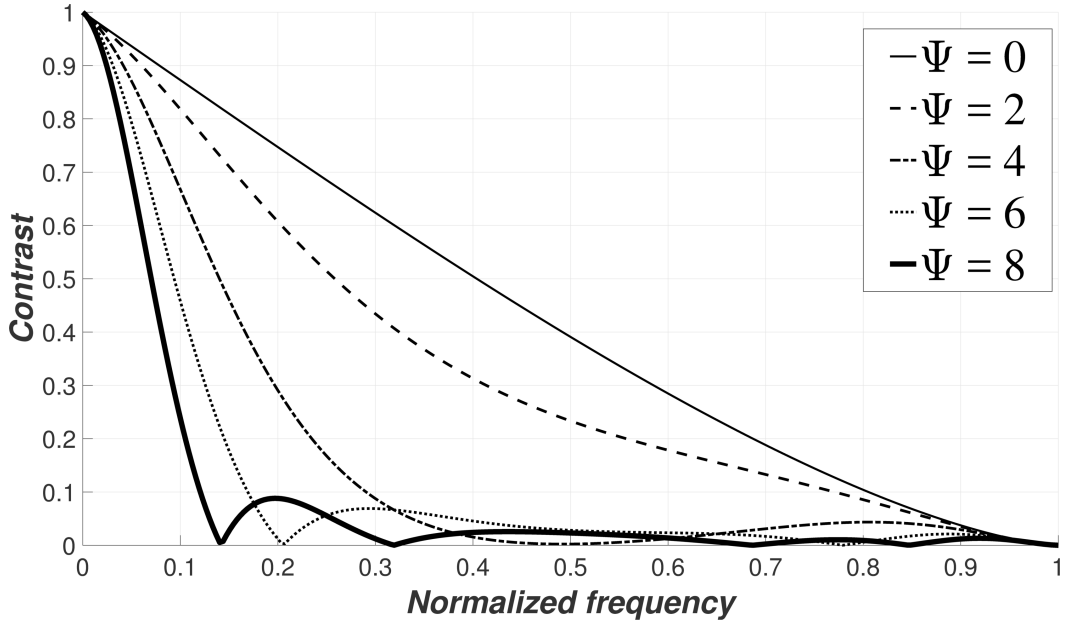


Fig. 7: MTF for a circular aperture for different value of the defocus parameter.

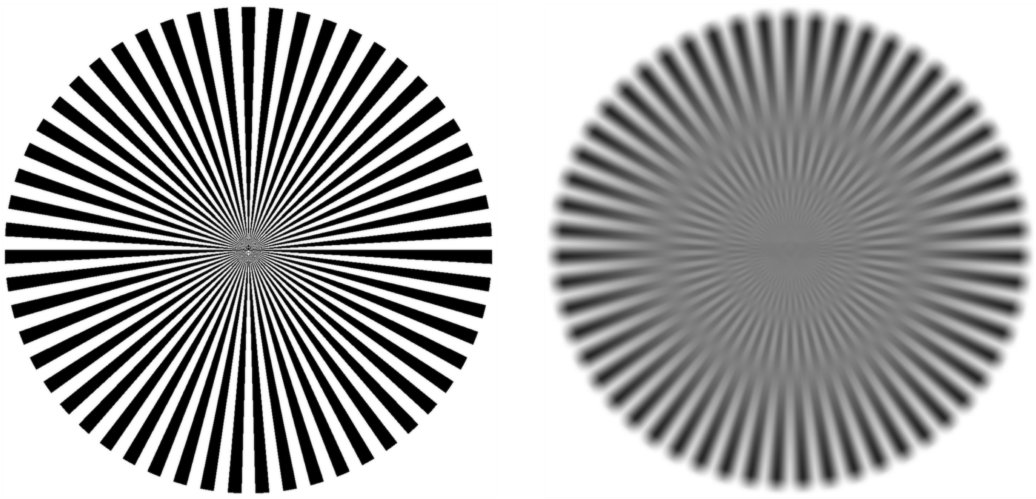


Fig. 8: Spoke target: imaging results for in-focus (left) and for severe OOF (right). Notice the contrast reversal, which appears on the right target.

In digital systems, a small error is permissible as long as the image size of a point source in the object plane is smaller than a single pixel in the detector plane. The range in which the object can move within the error limitation is called the depth-of field (DOF) which was also discussed in Section 2.1.1.

2.1.4 Optical aberrations

As described earlier in the Ray optics section, in case of an ideal lens, light from any point of an image will come to a focus on a single point at the focal plane. In practice, light does not focus to a single point, in the presence of lens aberrations. Those aberrations can be divided into two categories, Chromatic and Monochromatic aberrations.

Chromatic aberrations are caused by dispersion, which is the variation of the lens refractive index with wavelength. Effectively, this means that the focal distance will change for each wavelength. To overcome this type of aberration, a special lens design involving the use of several lenses with different dispersion, is required.

Monochromatic aberrations are caused by the geometry of the lens. When light is refracted by spherical surfaces the rays do not all converge to a point, even if they are of one wavelength. Monochromatic aberrations can arise from surfaces with irregularities, but they also naturally arise from spherical refracting surfaces.

To determine lens aberrations, one should apply Snell's law to every incoming ray and trace out the ray path accurately, taking in account the geometry of the system and dispersion. This is a very accurate way of describing the image formed by the lens and it fully includes the effects of monochromatic and chromatic aberrations. However, it is very time consuming, and unfortunately this action cannot be condensed into simple mathematical expressions. Ray tracing software, such as ZEMAX, take this approach.

Another way to analyze the lens aberrations is via wavefront analysis. The wavefront aberrations are the differences in the optical path between an actual wavefront and the ideal wavefront, as a function of position on the wavefront [2], [3].

The Zernike polynomials provide an analytical tool for evaluating lens aberration [4]–[8] and system optimization [9]. The wavefront aberrations in lenses having a circular pupil, can be expressed as Zernike polynomials:

$$W(\rho,\theta) = \sum_{m,n} R_n^m(\rho)\left[\alpha_n^m \cos(m\theta) + \alpha_n^{-m} \sin(m\theta)\right] \tag{22}$$

where ρ is the normalized radius of the exit pupil, θ is the azimuthal angle, α_n^m and α_n^{-m} are the even and odd polynomial coefficients respectively and n and m are positive integers $(n \geq m)$. For even $(n-m)$, the radial Zernike polynomial $R_n^m(\rho)$, are defined as:

$$R_n^m(\rho) = \sum_{k=0}^{(n-m)/2} \frac{(-1)^k (n-k)!}{k!\left(\frac{(n+m)}{2}-k\right)!\left(\frac{(n-m)}{2}-k\right)!} \rho^{n-2k}, \tag{23}$$

and are identical to zero for odd $(n-m)$. The polynomial coefficients are usually replaced with the more convenient Zernike coefficients Z_i. The first nine Zernike coefficients and their corresponding polynomial equations are listed in Table 1.

#	n	m	Polynomial equation	Aberration type
Z_0	0	0	1	None
Z_1	1	1	$\rho\cos(\theta)$	'x' tilt
Z_2			$\rho\sin(\theta)$	'y' tilt
Z_3	2	0	$2\rho^2-1$	Defocus
Z_4	2	2	$\rho^2\cos(2\theta)$	Astigmatism along 'x' and 'y'
Z_5			$\rho^2\sin(2\theta)$	Astigmatism along $\pm 45^0$ from the 'x' axis.
Z_6	3	1	$(3\rho^2-2)\rho\cos(\theta)$	'x' coma
Z_7			$(3\rho^2-2)\rho\sin(\theta)$	'y' coma
Z_8	4	0	$6\rho^4-6\rho^2+1$	Primary spherical

Table 1: The first nine Zernike coefficient.

2.1.5 Digital imaging

Modern cameras use photodetector arrays to capture digital images. The CMOS sensors provide high performance with a low-price tag, making it the leading sensor technology today. Pixel size has also become smaller and the smallest one today (commercially) is around 1μm.

There are two leading techniques to capture color images: (a) Three-CCDs are used in conjunction with color-separation beam splitter prisms that split the light into three sensors such that the three of them provide a full resolution RGB image; (b) A color filter array is placed in registration over the focal plane array where each pixel is covered with one of the R, G, or B color filters. The quantum efficiency curves provide the filter's response for the entire visible wavelength (Fig. 9). The second option is the most popular technique since only one sensor is required.

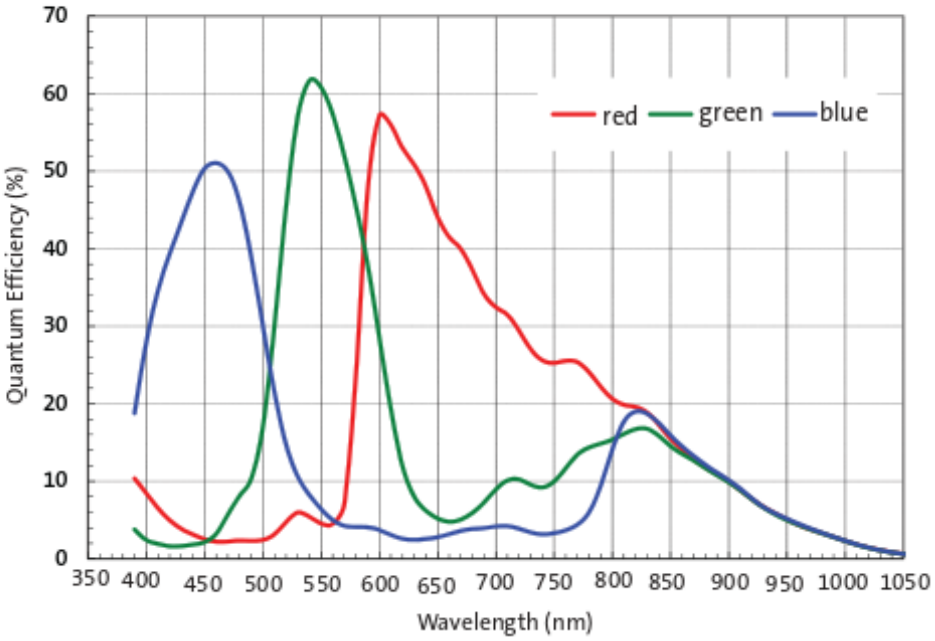


Fig. 9: Quantum efficiency curve for color array filter

The three different color channels of a standard sensor are organized in the form of a Bayer Matrix. A Bayer Matrix mosaic, shown in Fig. 10, is a color filter array that assigns the RGB color filters on a square grid of photo sensors. A particular arrangement of color filters is used in most single-chip digital image sensors used in digital cameras, camcorders, and scanners to create a color image. The filter pattern is 50% green, 25% red and 25% blue, hence it is called GRGB or in another permutation RGGB. Sub-pixels should be decoded in order to fill the matrix of each color (R, G and B). The process of decoding colors (Demosaicing) is done by the Image Signal Processor (ISP) in digital cameras.

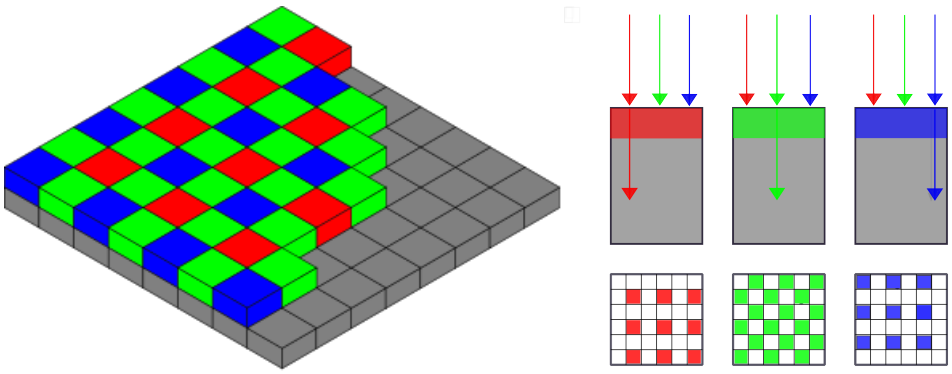


Fig. 10: Bayer matrix color array.

2.2 Computational photography

2.2.1 Light field Imaging

Light field imaging is one of the best example how CP can overcome conventional imaging limitations, allowing post-facto changes of the image DOF, focus point and even viewing angle. Adelson and Bergen [10] proposed the 7-dimensional Plenoptic function $P(x, y, z, \theta, \phi, \lambda, t)$ to describe the light field positions in space for all directions, color and time. In practice, the wavelength and time are usually omitted, and the ray space is represented as a 5D point. The two-plane light field introduced by Levoy and Hanrahan [11] and Gortler et al. [12], has been used in most studies. This 4D light field representation, describes a ray passing through two planes, using the 2D coordinates of its two intersections.

The acquisition of light field can be done by several methods. Levoy and Hanrahan [11] used a video camera mounted on a computer-controlled gantry to capture large arrays of images, each in a different viewing position. Wilburn et al. [13] constructed an array of 100 video cameras to capture the 4D light field simultaneously. Adelson and Wang [14] presented single lens Plenoptic camera where the light rays gathered through the main lens are recorded separately using a lenticular array placed on the sensor plane. Ng et al. [15] presented a hand-held light field camera, where a micro-lens array was placed on the sensor plane to separate the light rays gathered by the camera's main lens, as illustrated in Fig. 11.

The work of Ng paved the way for the first commercial light field camera introduce by Lytro, Inc [16], where the first generation utilized 100k microlens array on an 11MP sensor to produce a final resolution of 1.3MP (after computational rendering). The second-generation camera, Lytro ILLUM, used a 40MP sensor to produce a 4MP image. Both cameras are presented in Fig. 12.

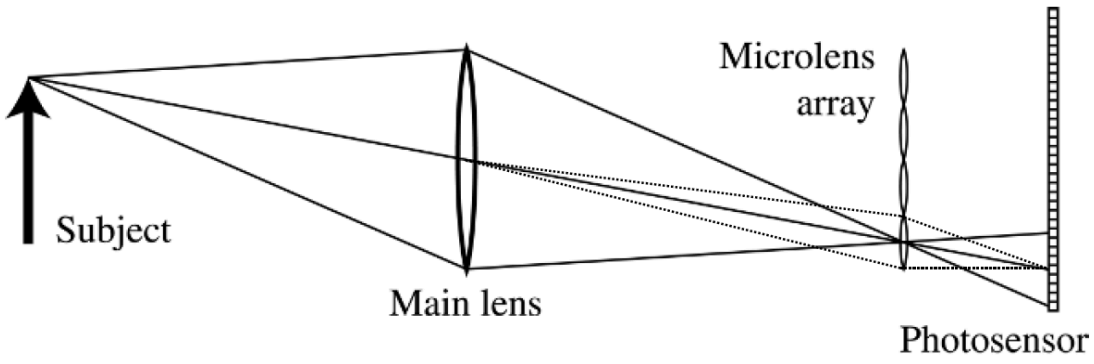


Fig. 11: Schematic diagram of the light field camera used in [15]. *The main lens focuses the subject onto the microlens array which separates the converging rays into an image on the photosensor behind it.*

Fig. 12: First (left) and second (right) generation light filed cameras from Lytro, Inc.

Some of the post processing features, available in light field cameras, are represented in Fig. 13. As illustrated in Fig. 11, each microlens is responsible for forming a sub-image on a small section of the main sensor. This can also be represented as rays, emerging from several parts from the focal plane as shown in Fig. 13. A full aperture imaging is achieved by summing all pixels behind each microlens (Fig. 13, top). Reducing the aperture size can be done by summing only the central portion of the rays of each microlens (Fig. 13, center). Image refocusing is done by summing rays from several adjacent microlens (Fig. 13, bottom).

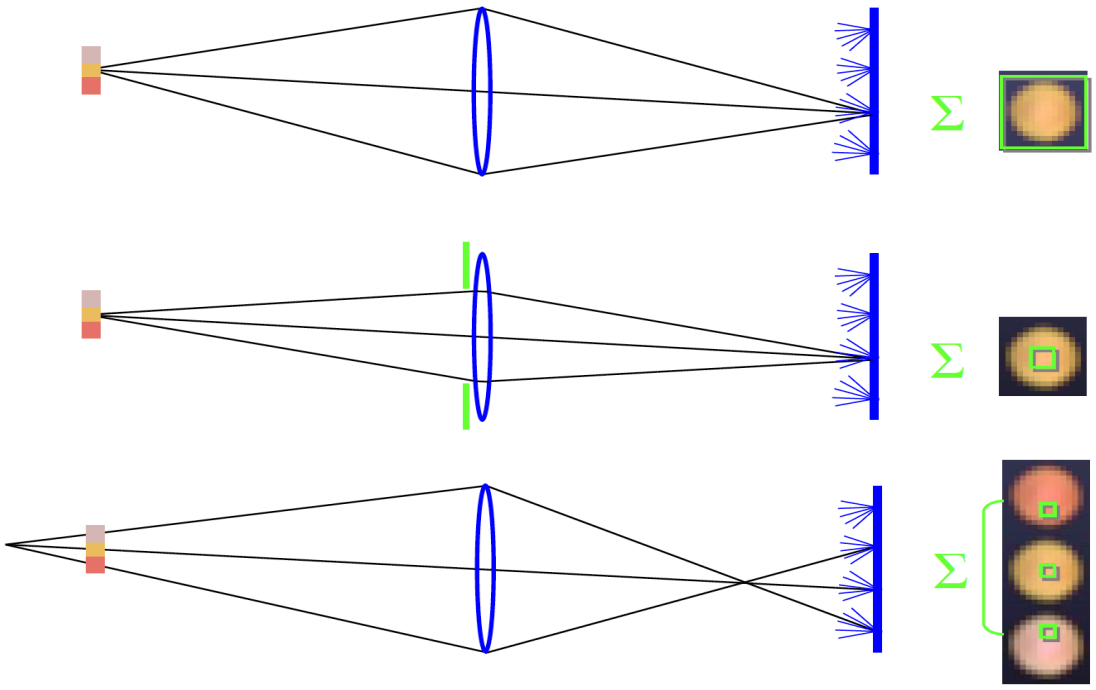


Fig. 13: Illustration of few post processing feature available in light field cameras. (Top) full aperture imaging; (Center) small aperture with large DOF; (Bottom) image refocusing.

Light field cameras can also produce depth maps by creating a focal stack image that can be utilized for producing depth map (see Section 2.3.3). Several studies recently utilized the 4D light field data for accurate depth map estimation [17], [18].

As mentioned in this section, light field cameras provide many post processing features which are not available in conventional imaging. However, light field cameras suffer from low resolution, bulkiness, and noise and require a unique design, which makes it harder to integrate them into existing systems, especially small-scale ones.

2.2.2 Computational sensors

The field of computational sensors deals with methods for manipulating image sampling, to overcome conventional sampling limitation such as capturing moving object, high speed video and lensless imaging.

In conventional imaging, opening the shutter for a specific exposure time is required for capturing a properly exposed single image. Moving objects during the exposure time cause motion blur, which destroys high-frequency spatial details. One can reduce the motion blur by setting fast exposure time, but the loss of light will increase the noise of the captured image.

A coded exposure camera modulates the exposure time by opening and closing the shutter within the exposure time using a carefully chosen binary pseudo-random code. The chosen exposure code produces an invertible motion PSF which allows using simple deconvolution methods to restore the blurred image. Raskar et al. [19] suggested such scheme using ferroelectric liquid crystal shutter set the exposure on/off, to enable severe motion blur restoration. This method relied on used intervention to crop the object of interest, assume linear 1D motion and reduce the light by 50%. A follow up paper [20] improved this technique by optimizing the coded exposure scheme for unknown motion direction and magnitude and increase SNR by allowing the 'on' time be greater than 50%. Tai et al. [21] presented a new approach for estimating spatially variant motion PSF but user assistant was still required of this scheme. McCloskey [22] utilized similar scheme using coded flash illumination instead of coded exposure. This method provided several advantages to coded aperture, most notable was increasing the signal (since shutter was open the entire time of image acquisition) but required external, camera synchronized flash device with a limited range such that this method can only be used for close distance objects.

Unlike still cameras, capturing video required high bandwidth which introduce a fundamental tradeoff between spatial and temporal resolution. High speed video cameras required high frame rate, which usually decrease spatial resolution or increase camera cost due to special hardware requirements. Studies such as Hitomi et al. [23] Reddy et al. [24] suggested high speed video cameras, which were based on compressive sensing models, that implemented using a liquid crystal on silicon modulator, as a per-pixel coded exposure modulator. Both papers presented similar results as in [23], 18 frames per video

frame were recovered which provided up to 1000 fps (with exposure time of 18ms per frame), with spatial resolution of 1280×1024, while in [24], 8 frame per video frame with spatial resolution of 1024×768 were recovered, providing up to 250 fps form a 25 fps video camera. The notable limitations of those methods include 50% loss of light (due to the modulator), and required a precise alignment between the modulator and the camera.

Another notable type of compressive sensing based cameras are Lensless cameras [25]–[28] that eliminate the optics completely, thus leading to new types of thin cameras. Lensless cameras capture scenes using sensor, usually equipped with a coded mask, and required a preprocess calibration to allow post-processing image restoration. Those cameras are still in their proof-of-concept stage due to several limitations such as low resolution and high computational time.

2.2.3 Coded aperture imaging

As discussed in Sections 2.1.2 and 2.1.3, the PSF is determined by the pupil shape and size, as well as by the phase exhibited in that plane. Coded aperture techniques try to manipulate the PSF of conventional corrected (with no aberrations) imaging system, by introducing diffractive elements in the pupil plane [29]–[36]. Other studies utilize special uncorrected optical designs [37]–[40] to produce similar results. Such designs usually reduce the quality of the image obtainable for in-focus conditions, but offer other advantages such as EDOF and depth estimation abilities. Coded aperture techniques can be divided into two groups: the first one tries to produce a constant PSF for large DOF while the second group focuses on creating a unique PSF design for each depth.

2.2.3.1 Constant PSF methods

In what follows, techniques for creating imaging systems with constant PSF throughout the desired DOF are reviewed. Constant PSF means also corresponding constant MTF curves; nevertheless, those will be low, resulting in low contrast images. However, since the PSF is almost constant throughout the entire DOF, the resulting image contrast can be restored digitally, using a non-blind de-convolution algorithm [41] taking advantage of the uniform PSF provided by the imaging system. The main drawback of those methods is the extensive computational power required for utilizing the deconvolution algorithm. Another inherent issue is the noise amplification caused by such deconvolution operation. One should note however, that since the PSF is almost constant for all depths under consideration, the ability to extract depth information is very poor.

Imaging methods that use a wave-front coding mask (either phase, or amplitude, or both) became more attractive in the last two decades. One of the first and most prominent studies in this field was carried by [29], where a cubic phase mask was designed to generate a constant PSF for large DOF. A different approach is based on radially symmetric binary optical phase masks [33] composed of one or several rings providing a predetermined phase-shift (usually, π value) and with an optional amplitude ring. Unlike the difficulties encountered in the fabrication of phase profiles of 3rd order polynomial

degree structures, such as the cubic mask, the binary phase masks are easy and inexpensive to manufacture on a mass production scale.

One should note that the cubic as well as the binary phase masks provide the exact desired phase shift for a single wavelength only, whereas the phase shift for other wavelengths changes accordingly. A polychromatic phase mask [34] can overcome this limitation by optimizing the mask parameters to achieve similar phase shift for all wavelength. A different approach relies on a pair of masks with inverse profiles and different dispersion properties [42]. This "doublet-like" mask design is more complex than a simple mask but can provide the desired phase shift for all wavelengths, if so desired.

Using an uncorrected lens with optical aberration [40] or purposely designing one with specific spherical aberration [39], one can also produce constant PSF at the expense of reducing the image quality for in-focus scenarios. The advantage of such an approach is that there is no need for additional diffractive elements in the lens assembly. Moreover, the design of the lens is much simpler since there is no need for many corrected elements.

Focal sweep is a technique whereby one captures several images of the same scene while changing the focus point. This technique can be used for EDOF and depth estimation, but it requires acquisition of several images. To produce an EDOF image from a single frame, Kuthirummal et al. [43] moves the focus point while the picture was captured and uses a computational stage to recover the EDOF image. This approach was presented later in [44] using a radial diffuser which acted as a passive focal sweep element. Similar concept was implemented using uncorrected lens as a type of spectral focal sweep [37] such that the amplified chromatic aberration produces a constant PSF, an average for all wavelength.

2.2.3.2 Depth sensitive PSF

The constant PSF approach produces "flat" images without depth-related changes, making the restoration process much simpler as the PSF is known in advance. However, for depth estimation purposes, the PSF should be depth-dependent, while at the same time retaining enough information to restore a reliable EDOF image in its entirety.

A study by Levin et al. [31] used an amplitude-coded aperture with a conventional camera to produce an all-in-focus image allowing depth estimation after proper digital processing. A related method presented by Zhou et al. [45] uses amplitude-coded aperture pairs for depth from defocus and deblurring. Although Zhou et al. [45] improved the depth estimation presented by Levin et al. [31], the obvious downside is that this process requires taking two images while switching apertures between two frames and keeping the camera still as well. The main drawback of both solutions based on amplitude masks is the reduction of light efficiency by 50% on the average. A view of these masks is shown in Fig. 14.

Another related study utilized a color-dependent ring mask [46], whereby the aperture size, and as a result, the DOF, is color-dependent. This spectrally-varying DOF has been used for EDOF and depth estimation whereby the foreground and background can be

segmented for refocusing purposes. Similar to the solutions based on amplitude masks mentioned in the previous paragraph, the light efficiency of this case is reduced by 60%, making it unsuitable for low light conditions.

A different approach, proposed by DxO labs [38], utilizes a lens with deliberately high chromatic aberration to achieve color separation. However, increasing longitudinal chromatic aberrations while reducing other types of aberrations (lateral chromatic aberrations, spherical aberrations …) requires special lens design.

Fig. 14: Amplitude masks example: left - Levin et al. [31]*; right - Zhou et al.* [45]

Milgrom et al. [35] proposed the use of a special RGB phase mask that exhibits significantly different response in the three major color channels R, G and B. It has been shown that each channel provides best performance for different depth regions, so that the three channels jointly enable coverage of an extended DOF. Its major advantage is light efficiency above 95%. This method was specifically designed for use of EDOF in barcode imagers. A follow-up on this approach is investigated in this dissertation for EDOF imaging of natural scenes [36], [47] as well as for estimating a depth map from a single image. Details are provided in Section 3.

2.3 Depth imaging

2.3.1 Stereo

One of the earliest passive depth sensing techniques generally involve stereo camera systems [48]–[50]. The camera uses stereo triangulation to generate depth information. The process requires the calibration of two or more cameras that simultaneously capture 2-D images of a scene, each one from a slightly different angle. An algorithm, known as the correspondence problem, is then used to locate corresponding pixels in the images. The stereo camera approach has several disadvantages. The disparity problem arises when one image captures a feature that is not contained in the other image due to occlusion of

objects. Another common issue is that uniform areas with few objects, or areas containing many objects, can greatly reduce accuracy [51].

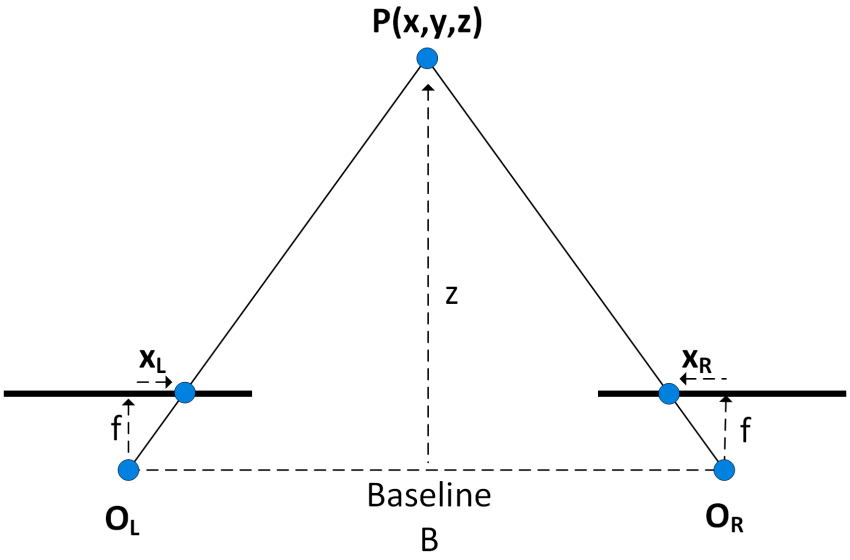


Fig. 15: Stereo camera illustration. The left and right sensors are centered at O_L and O_L respectively.

The basic concept of stereoscopy is illustrated in Fig. 15. A point object $P(x,y,z)$ will project a point on each of the two left and right sensors located in x_L, x_R respectively. Camera calibration step [52] is required to compensate for small variation in manufacture or assembly. The distance between the centers of the two left and right sensors is called Baseline, which affects the depth measurement accuracy i.e. a larger baseline provides better angular resolution. Using the known camera parameters such as focal lengths, pixel sized, and baseline, the equations of the projection lines through these image points is calculated to evaluate the distance z:

$$z = \frac{f \cdot B}{(x_L - x_R)} \tag{24}$$

The depth measurement accuracy [53] is dependent on several parameters: noise, which reduce the ability of finding the correct correspondence point; pixel size, which directly correlate with the disparity term $(x_L - x_R)$ evaluation accuracy; focal length and baseline (Eq. (24)). Another important limitation is related to the DOF of the camera, determined by the finite aperture size. Finding the correspondence required some texture, which cannot be detected in blurry areas outside the DOF of the camera. For this reason, the aperture size in stereo cameras is usually small to produce large DOF, however this also

increases noise. In addition, such a dual camera system significantly increases the form factor, cost and power consumption.

2.3.2 Active illumination methods

Active depth sensing methods composed from a single camera and a light emitting source. The classical active depth sensing methods is the time of flight method [54], which have been used in many LIDAR (a backronym for Light Detection and Ranging) applications, in robotics and the autonomous vehicles.

Time of flight depth sensors collect data using transmitting a signal, generally a light wave, and measuring it as it reflects from the object. The time of flight and the speed of the signal are then used to calculate an accurate depth measurement. As its accuracy is inversely related to the distance traveled, time of flight sensor required to distinguish between femtoseconds, making it more useful for outdoor depth sensing [55]. However, time of flight depth sensing methods usually produce very dense range images and require little or no image processing.

Time of flight devices use single laser stripe scanned progressively over the surface of an object, required the object to be static during scanning process. To overcome this limitation, coded structured light depth sensing methods involve the projection of a light pattern such as multi-stripe and sinusoidal fringe [56]. Each point in the pattern is encoded with information that identifies its coordinates [57]. Once each point of light is reflected and captured by a sensor, the depth map is calculated using triangulation methods (see Fig. 16) which is similar to the one presented in stereo [58]. However, as opposed to passive stereo, the correspondence problem is solved by the encoding of the projected light.

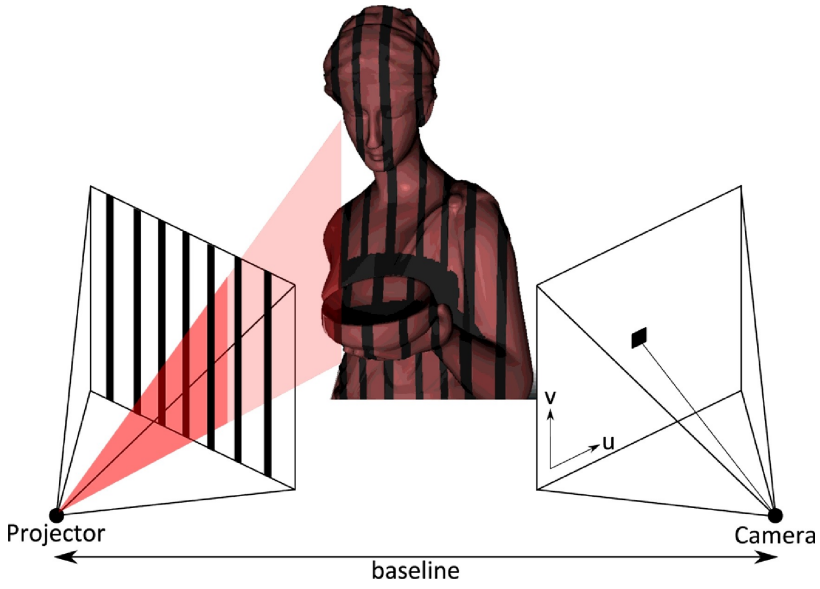


Fig. 16: Structured light system illustration. (source [59] *)*

The well know Kinect sensor utilizes structural light projection alongside an RGB sensor, thus providing consumers an affordable device that can collect both depth and color images [59]. The Kinect has been used to create RGBD data sets such as NYUD [60], [61] which are used in many computer vision depth learning methods.

2.3.3 Depth from focus or defocus

Depth from Focus (DFF) refers to method for depth estimation from an input consisting of several images of the same scene, captured with different focus point, commonly known as focal stack [62]–[65]. This can be achieved by keeping the camera still, while moving an object along the optical axis (using a rail) or by keeping the object still while changing the camera focus point (usually by varying the distance between the sensor and the pupil plane). Depth can be estimated by identifying an image from the focal stack which exhibits highest sharpness. This process is repeated for each pixel (or a small patch). As the number of frames is limited, usually the object will not be in perfect-focus for a specific image, or part of it. Nevertheless, one can interpolate the location of the ideal focus point based on the responses from adjacent images. By stitching together the sharpest in-focus pixels across the focal stack, DFF can also generate an all-in-focus image [64], [65].

The main drawback of DFF is the need of acquiring several images while the scene remains still. This scenario is plausible when using DFF for static scenes with controlled environment, such as industrial inspection, but in practice, frames in the focal stack are not perfectly aligned. Suwajanakorn et al. [64]] introduced a method to compute depth maps with mobile phone cameras, by aligning a focal stack of 25 frames to overcome scene parallax issue (see Fig. 17). The computational time for producing a 640×360 depth map and all-in-focus RGB image was around 20 minutes. Kim et al. [65] extended this process for video, but only one depth map produced for each focal stack (30-100 frames). In both studies, the assumption was that the motion in the scene is minor and lightening condition remains the same.

Similar to the DFF method, Depth from Defocus (DFD) also captures several images with the same focus point but with different aperture settings [66], [67]. In-focus areas appear quite independent of the aperture setting, but OOF areas get increasingly blurry when the aperture is widened. This method also suffers from the same alignment issues as DFF.

DFD can also be achieved with a single image. Tai and Brown [68] use local contrast prior to measuring the defocus at each pixel and then apply Markov Random Field (MRF) propagation to refine the defocus map. Zhuo et al [69] estimated the amount of defocus blur at edge locations by blurring the input image, using a known Gaussian kernel, and calculating the ratio between the gradients of input and blurred images. By propagating the blur amount at edge locations to the entire image, a full defocus map can be obtained. Several papers presented similar approach with improved performance [70], [71]. The

reported computational time in [71] for producing an 800×600 depth map was around 3 minutes utilizing MATLAB implementation.

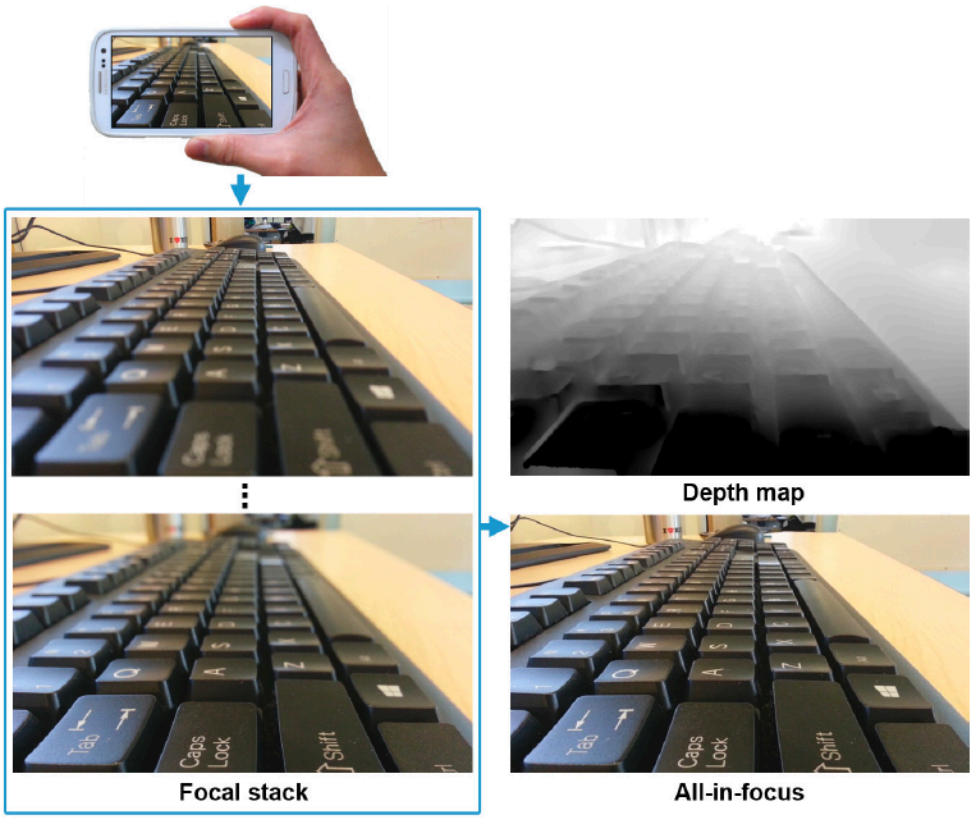


Fig. 17: Illustration of the proposed scheme in [64]. *The focal stack capture using a mobile phone used to compute all-in-focus image and a depth map estimation.*

2.3.4 Monocular based algorithms

Depth perception in human vision relies on both binocular and monocular cues [72]. As presented in Section 2.3.1, binocular cues are based on human stereo vision, as disparity between two different view-points of the same scene allows triangulating the distance to an object. Monocular cues however, depend on a single view, and are based on visual features observed in a static view of a scene, or motion-based cues, which are based on motion parallax, where nearby objects appear to move faster than farther objects.

Visual features observed in a static view can be described by eight types of cues [72]:

Occlusion: occurs when one object partially hides another one in a view. The occluded object appears farther to the observer, which provides information about relative depth.

Perspective Convergence: When parallel lines extend from an observer, they appear as if they are converging as distance increases.

Relative size: Given two objects of equal size, the one farther to the observer appears to be smaller. This cue depends on the observer's knowledge about the objects actual size.

Familiar size: Given two objects of different known sizes, one can judge distance based on prior knowledge about the size of the objects. For example, if the objects appear in the image to have the same size, the observer will assume that the smaller one is closer.

Relative height: This cue is related to the position of objects with respect to the horizon line in the visual field. Objects that are near the horizon usually appear at distance.

Texture gradients: As distance increases, patterns of a textured surface get finer and appear smoother.

Atmospheric Perspective: Details of distant object are degraded by atmospheric conditions like haze, fog, and smoke. As distance increases, details are less visible with respect to those which are closer.

Shadows: The cast shadow of objects can provide information about the 3D object shape, and its relative location in the scene.

Many studies in computer vision have utilized those monocular cues to predict depth map from a single image. Saxena et al. [73] estimated an absolute depth map, using features such as texture energy, texture gradients, and haze, calculated from square image patches and their neighbors at multiple size scales. Features for relative depth were also computed, based on the difference between neighboring patches' histograms of the absolute depth features. They then modelled the depth estimation problem as a Markov Random Field (MRF), using both Gaussian and Laplacian distributions for a posteriori distribution of the depth.

Eigen et al. [74] employed an architecture of two deep neural networks to the depth estimation problem, one of which makes a coarse global prediction, and the other one locally refines it. The deep network requires massive amounts of training data, so it is further augmented by applying scaling, rotation, translation, color variation, and horizontal flips to existing data. Liu et al. [75] train a deep neural network architecture, based on learning the unary and pairwise potentials of a Continuous Random Field (CRF) model, using a deep CNN framework.

Chen et al. [76] follow up on research by Zoran et al. [77], on estimate metric depth, using depth relations between pairs of points in an image. By training the network on a large dataset, they achieve metric depth prediction performance compared with algorithms trained on dense metric depth maps such as Eigen et al. [74].

2.4 Sparse representation

2.4.1 Background

Ideas of sparse, redundant, and otherwise parsimonious representations have attracted considerable attention in the field of signal processing and beyond [78]–[81]. As opposed to the conventional priors on signals such as "band limitedness" set be by classical Nyquist-Shannon sampling theory, these approaches assert that certain signals can be represented using only a few "atoms" in a redundant dictionary. When incorporated into the analog-to-digital conversion setup, these ideas suggest that appropriately designed linear measurements can sense the signal directly in a domain in which it has a low-dimensional representation, leading to compressed sensing (CS) techniques. CS was successfully used for efficient data acquisition protocols in several fields, most notably in medical imaging. CS is used to speed up MRI while preserving diagnostic quality [82], [83], in CT imaging [84] to reduce radiation exposure, and in medical ultrasound to reduce the data transfer rate and allow wireless probes [85], [86].

Flavors of sparse and redundant representations [87]–[94] have been proved as a powerful tool for image processing, compression, and analysis. It is now well-established that small patches from a natural image can be represented as a linear combination of only a few atoms (predefined patches) in an appropriately constructed over-complete (redundant) dictionary. This constitutes a powerful prior that has been successfully employed to regularize numerous otherwise ill-posed image processing and restoration tasks such as denoising [87], [95], [96], image inpainting [97], [98], super-resolution [88], [91], [99], demosaicing [100] and more.

2.4.2 Sparse representation of natural images

Sparse representation of natural image signals has been proven in recent years to be a powerful tool for image representation. Natural images share common features over small patches and therefore by using an overcomplete dictionary one can represent those patches as a linear combination of a few predefined patches from our dictionary.

Consider a signal column vector $\mathbf{x} \in \mathbb{R}^m$ and dictionary $\mathbf{D} \in \mathbb{R}^{m \times N}$ composed of $N > m$ columns atoms signals. The signal $\mathbf{x}$ has sparse approximation over $\mathbf{D}$ if one can find a vector $\mathbf{z} \in \mathbb{R}^N$, having only a few non-zero coefficients, such that $\mathbf{Dz}$ will give a close approximation of the signal $\mathbf{x}$. This sparse approximation problem can be described as:

$$\min_{\mathbf{z}} \left\| \mathbf{x} - \mathbf{D}\mathbf{z} \right\|_2^2 + \lambda \left\| \mathbf{z} \right\|_0 \tag{25}$$

where $\left\| \mathbf{z} \right\|_0$ counts the number of non-zeros in the vector $\mathbf{z}$, and λ controls the relative importance of the ℓ_0 regularization term over the ℓ_2 data fitting term.

Taking into consideration an additive Gaussian noise and a representation error, the problem in Eq. (25) can be formulated as:

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \qquad \text{s.t. } \|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2^2 < \varepsilon \tag{26}$$

where $\varepsilon > 0$ is the allowed representation error. In this form, the solution to Eq. (26) favors sparsity of $\mathbf{z}$ over data fitting. An alternative form of the above problem exchanges the data fitting term with the constraint,

$$\min_{\mathbf{z}} \|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2^2 \qquad \text{s.t. } \|\mathbf{z}\|_0 < S \tag{27}$$

where S is the target sparsity.

While the ℓ_0 norm minimization problem is known to be computationally intractable, approximation and relaxation methods exist. In particular, theoretical results in [101], [102] suggest that under certain conditions on the dictionary (high mutual incoherence), the ℓ_0 regularization term can be relaxed into the convex ℓ_1:

$$\min_{\mathbf{z}} \|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1 \tag{28}$$

while keeping the relaxed problem equivalent to the original one. The above LASSO problem [101] can be solved using numerous algorithms such as coordinate descent [103]–[105], Least Angle Regression (LARS) [106] and the Iterative Shrinkage Thresholding Algorithm (ISTA) [107]–[113].

2.4.3 Iterative-shrinkage algorithms

Minimization problem (28) is a convex program that can be solved using ISTA [107]–[113]. Given a gradient descent step size μ, ISTA first performs a gradient descent step on the smooth (ℓ_2) part of the objective:

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \mu \mathbf{D}^T \left(\mathbf{x} - \mathbf{D}\mathbf{z}_t\right) \tag{29}$$

The algorithm monotonically decreases the value of the objective function if the inverse of the gradient step size $1/\mu$, is larger than the largest eigenvalue of $\mathbf{D}^T\mathbf{D}$. Next, the element-wise soft-thresholding operator implementing the proximity map of the non-smooth ℓ_1 term is applied:

$$S_\lambda(\theta) = \text{sign}(\theta) \cdot \max\left(|\theta| - \lambda, 0\right) \tag{30}$$

The ISTA algorithm scheme is presented in Fig. 18.

The simplicity of ISTA make it attractive for solving large-scale problems, however, it is also known to converge slowly. Beck and Teboulle [112] presented a fast version of ISTA (dubbed FISTA) which preserves the computational simplicity of ISTA but with a global

rate of convergence which was proved to be faster than ISTA (in fact, the fastest possible for first-order methods).

Inputs:

- signal $\mathbf{x} \in \mathbb{R}^{m \times 1}$ and dictionary $\mathbf{D} \in \mathbb{R}^{m \times N}$
- regularization parameter λ and gradient descent step size μ
- number of iterations T

Output:

- Approximation vector $\mathbf{z} \in \mathbb{R}^{N \times 1}$ such as $\mathbf{x} \approx \mathbf{D}\mathbf{z}$

Initialization:

- Initial solution $\mathbf{z}_0 = 0$

Main iteration loop: (set $t=1$)

- While ($t \le T$)
  - Perform gradient descent step: $\mathbf{z}_t = \mathbf{z}_{t-1} + \mu \mathbf{D}^T \left(\mathbf{x} - \mathbf{D}\mathbf{z}_{t-1} \right)$
  - Apply soft thresholding operator: $\mathbf{z}_t = S_\lambda \left(\mathbf{z}_t \right)$
  - $t = t + 1$
- End while
- Set the solution to: $\mathbf{z} = \mathbf{z}_T$

Fig. 18: The ISTA algorithm

2.4.4 Orthogonal matching pursuit

In addition to the convex relaxation techniques, another family of approaches to tackle the intractability of the ℓ_0 norm minimization problem is greedy approximation, of which Matching Pursuit (MP) and its variant, the Orthogonal Matching Pursuit (OMP) [114]–[116] are prominent members.

The MP algorithm starts by finding the best fitting atom which presents the largest inner product with the signal in (25). This atom index is then added to the support I of the representation vector $\mathbf{z}$. Next, one finds the second atom which fits the residual and after adding it to the support I, the coefficient vector $\mathbf{z}$ is recalculated such that it will minimize the data term $\left\| \mathbf{x} - \mathbf{D}_I \mathbf{z} \right\|_2^2$, where the dictionary $\mathbf{D}_I \in \mathbb{R}^{m \times N}$ is equal to $\mathbf{D}$ for column vectors with indices I and zero elsewhere. The process terminates when either the size of the support I reaches the sparsity threshold S, or when the data fitting term becomes smaller than the error threshold ε. Note that the algorithm is greedy since atom indices added to the support at previous iterations cannot be removed. Unlike MP that greedily

approximates both the support and the related coefficients (i.e., the values of $\mathbf{z}$ are never re-calculated), the OMP variant recalculates $\mathbf{z}_k$ at every iteration by solving a simple least-squares problem with the current support. The OMP scheme is described in Fig. 19.

Inputs:

- signal $\mathbf{x} \in \mathbb{R}^{m\times 1}$ and dictionary $\mathbf{D} \in \mathbb{R}^{m\times N}$
- error threshold ε and sparsity threshold S

Output:

- Approximation vector $\mathbf{z} \in \mathbb{R}^{N\times 1}$ such as $\mathbf{x} \approx \mathbf{D}\mathbf{z}$

Initialization:

- Initial solution $\mathbf{z}_0 = 0$
- Initial residual $\mathbf{r}_0 = \mathbf{x}$
- Initial support $I_0 = \varnothing$;

Main iteration loop: (set k=1)

- While ($k \le S$)
  1. Estimate best fit atom index as:
     $i = \arg\max_i \left\langle \mathbf{d}_i^T, \mathbf{r}_{k-1} \right\rangle \quad ;\forall i \notin I_{k-1}$
  2. Update index vector: $I = I \bigcup i$
  3. Update solution: $\mathbf{z}_k = \min_{\mathbf{z}} \left\| \mathbf{x} - \mathbf{D}_I \mathbf{z} \right\|$
  4. Compute residual: $\mathbf{r} = \mathbf{x} - \mathbf{D}\mathbf{z}_k$
  5. Stop if $\left\| \mathbf{r}_k \right\|_2^2 < \varepsilon$ otherwise $k = k + 1$
- End while
- Set the solution to: $\mathbf{z} = \mathbf{z}_k$

Fig. 19: Orthogonal Matching Pursuit algorithm.

2.4.5 Dictionary learning

Natural images share common features over small patches. The dictionary $\mathbf{D}$ can be produced from either predesigned transforms [117], [118] such as DCT (Discrete Cosine Transform), Wavelet [119], [120], contourlets [121], [122], bandlets [123], [124], curvelets [125], [126] or from a set of sampled data from training images [95], [127]. The latter has been selected as the favored dictionary for image deblurring in many studies and will also be used in the scope of this dissertation.

The dictionary learning process is based on finding a dictionary $\mathbf{D}$ that minimizes the objective function in Eq. (25) on training data. One of the most popular methods for dictionary training is the *k*-SVD method [127] which is an iterative process, alternating

between finding the best vector coefficients using OMP and then updating $\mathbf{D}$ according to the current coefficients vector using Singular Value Decomposition (SVD).

Giving a set of sample signals $\{\mathbf{x}_i\}_{i=1}^{M}$ which can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{m \times M}$, and the corresponding sparse coefficients matrix $\mathbf{Z} \in \mathbb{R}^{N \times M}$, the *k*-SVD process attempts to factorize $\mathbf{X}$ into the product $\mathbf{DZ}$, wherein the right factor $\mathbf{Z}$ is sparse. This is achieved by minimizing:

$$\min_{\mathbf{D},\mathbf{Z}} \|\mathbf{X} - \mathbf{DZ}\|_2^2 \qquad \text{s.t. } \forall i,\ \|\mathbf{z}_i\|_0 < S \ . \tag{31}$$

The first step in *k*-SVD is to find a sparse coefficient matrix $\mathbf{Z}$. In the second stage, each atom is optimized one at a time. To isolate the *k*-th atom, $\mathbf{d}_k$, the data term from (31) is written as:

$$\begin{aligned} \|\mathbf{X} - \mathbf{DZ}\|_2^2 &= \left\| \mathbf{X} - \sum_{j=1}^{N} \mathbf{d}_j \mathbf{z}_T^j \right\|_2^2 \\ &= \left\| \left(\mathbf{X} - \sum_{j \neq k} \mathbf{d}_j \mathbf{z}_T^j \right) - \mathbf{d}_k \mathbf{z}_T^k \right\|_2^2 \end{aligned} \tag{32}$$

In this form, the matrix multiplication $\mathbf{DZ}$ was replaced with the summation of *N* rank-1 matrices. Notice that the row vector $\mathbf{z}_T^k$ in the *k*-th row in matrix $\mathbf{Z}$, should not be confused with the *k*-th column vector $\mathbf{z}_k$.

The error matrix $\mathbf{E}_k$ is defined as the data error when removing the *k*-th atom from the dictionary:

$$\mathbf{E}_k = \mathbf{X} - \sum_{j \neq k} \mathbf{d}_j \mathbf{z}_T^j \ . \tag{33}$$

One can use SVD to update the atom $\mathbf{d}_k$ and corresponding coefficients row $\mathbf{z}_T^k$ to minimize (32), but this will result in a non-sparse $\mathbf{z}_T^k$, as such, the representation matrix $\mathbf{Z}$ will also not remain sparse. To avoid this issue, we define the indices group ω_k that points to the examples of $\mathbf{X}$ that use the atom $\mathbf{d}_k$, i.e. nonzero entries of $\mathbf{z}_T^k$:

$$\omega_k = \left\{ i \,|\, 1 \le i \le M, \mathbf{z}_T^k(i) \neq 0 \right\} \tag{34}$$

Now we can define the restricted matrix $\mathbf{\Omega}_k \in \mathbb{R}^{M \times M_k}$ (M_k is equal to length of ω_k), with ones on the $(\omega_k(i), i)$ entries and zeros elsewhere. This matrix shrinks the vector $\mathbf{z}_T^k$ such that the vector $\mathbf{z}_R^k = \mathbf{z}_T^k \mathbf{\Omega}_k$ contains only nonzero entries. The restricted error matrix

$\mathbf{E}_k^R = \mathbf{E}_k \mathbf{\Omega}_k$ can also be represented as the only the columns in $\mathbf{E}_k$ that correspond to examples that are using the atom $\mathbf{d}_k$. Using the above notation, the minimization of the $\mathbf{E}_k^R$ keep the support of $\mathbf{z}_T^k$ intact.

Using SVD, the restricted error matrix $\mathbf{E}_k^R$ can be decomposed to $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$ and update the atom $\mathbf{d}_k$ as $\mathbf{u}_1$ (the first column of $\mathbf{U}$) and $\mathbf{z}_T^k$ as the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$. The full scheme of the *k*-SVD algorithm is presented in Fig. 20.

Inputs:

- Examples group $\{\mathbf{x}_i\}_{i=1}^{M}$ and initial dictionary $\mathbf{D} \in \mathbb{R}^{m \times N}$
- error threshold ε_d and sparsity threshold S
- maximum iteration limit *K*

Output:

- Dictionary $\mathbf{D} \in \mathbb{R}^{m \times N}$ which best represents the data examples (Eq. (31))

Initialization:

- Initial Dictionary with random N samples form the examples group
- Initial solution $\mathbf{z}_0 = 0$
- Initial residual $\mathbf{r}_0 = \mathbf{x}$
- Initial solution index vector $I_0 = \varnothing$;

Main iteration loop: (set k=1)

- While ($k \le K$)

Sparse approximation stage:

Apply OMP to find the sparse coefficient matrix $\mathbf{Z}$

- *For* $i = 1:M$

  $\mathbf{z}_i = \arg\min_{\mathbf{z}} \|\mathbf{x}_i - \mathbf{D}_{k-1}\mathbf{z}\|_2^2 \quad \text{s.t. } \|\mathbf{z}\|_0 < S$

- End

Dictionary update stage:

- *For* $j = 1:N$
- Calculate the error matrix $\mathbf{E}_k = \mathbf{X} - \sum_{j \ne k} \mathbf{d}_j \mathbf{z}_T^j$
- Find the indices group ω_k with points to examples that use the atom $\mathbf{d}_k$

  $\omega_k = \{i \mid 1 \le i \le M, \mathbf{z}_T^k(i) \ne 0\}$
- Calculate the restricted matrix $\mathbf{\Omega}_k$
- Calculate the restricted error matrix $\mathbf{E}_k^R = \mathbf{E}_k \mathbf{\Omega}_k$
- Apply SVD $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$
- Update atom $\mathbf{d}_k = \mathbf{u}_1$
- Update coefficients row vector as $\mathbf{z}_T^k = \mathbf{\Delta}(1,1) \cdot \mathbf{v}_1$
- End

Stopping criteria:

- Stop if $\|\mathbf{X} - \mathbf{D}_k\mathbf{Z}\|_2^2 < \varepsilon_d$ otherwise $k = k + 1$
- End while
- Set the solution to: $\mathbf{D} = \mathbf{D}_k$

Fig. 20: k-SVD algorithm.

2.5 Basic concepts of convolutional neural networks

Convolutional Neural Networks (CNNs) are multi-layer feed-forward networks, inspired by Hubel and Wiesel's study of neurobiological signal processing in cat's visual cortex [128]. Based on first technical realization of the brain visual recognition from Fukushima [129], LeCun et al. [130] created one of the first CNNs that was trained to recognize handwritten characters (Fig. 21.). Following this study, several deeper CNN structures were created to solve classification tasks based on two-dimensional input images [131]–[133]. Deep neural networks have (CNNs in particular) created revolution by achieving state-of-the-art results in many tasks across different domains of science and engineering such as speech recognition [134], sentence classification [135], gene ontology annotation predictions [136], and even for advertising [137].

Using convolutional layers, the networks can learn to recognize local features, such as edges or corners, by restricting the receptive fields of hidden layers (layers between input and output layers which their value is not given by the data) to local connectivity and to add shift invariance by enforcing spatially shared weights. Furthermore, spatial subsampling in the form of pooling layers reduces sensitivity to shifts and distortion.

2.5.1 CNN architecture

The input and output of each layer are sets of arrays called feature maps. Those three-dimensional feature maps can be connected with different types of layers.

Convolutional layers consist of multiple filters that are defined by their weights. The layer defines the number of filters and their kernel size, the stride in which they are applied and the amount of padding to handle image borders. The convolved output of a filter is called a feature map and a convolutional layer with *n* filters creates *n* feature maps, which are the input for the next layer as illustrated in Fig. 21.

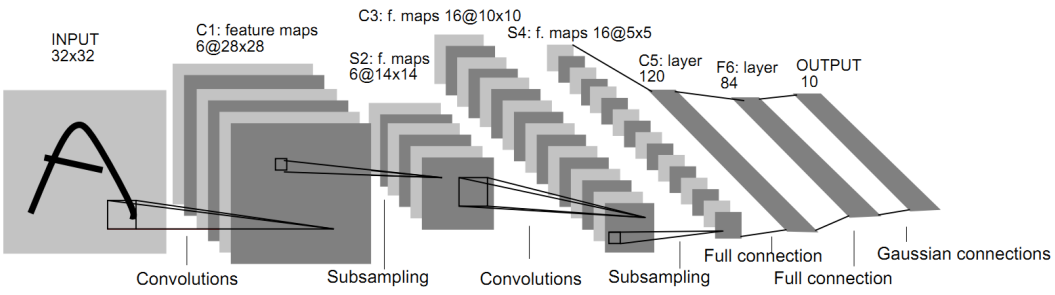


Fig. 21: Architecture of LeNet-5 [130] *CNN for handwriting recognition.*

Pooling layers reduce feature map resolution and thereby the sensitivity to shift and distortions, as exact feature location is discarded, and only relative and approximate

location information remains. Typical polling method called Max-pooling, outputs the maximum value within a rectangular neighborhood of the activation map.

Another way of reducing the data volume size is adjusting the stride parameter of the convolution operation. The stride parameter controls whether the convolution output is calculated for a neighborhood centered on every pixel of the input image (stride 1) or for every n-th pixel (stride n). Research has shown that pooling layers can often be discarded without loss in accuracy by using convolutional layers with larger stride value [138]. The stride operation is equivalent to using a fix grid for pooling.

Batch Normalization layer [139] quickly became very popular mostly because it helps to converge faster. It adds a normalization step (shifting inputs to zero-mean and unit variance) to make the inputs of each trainable layers comparable across features. It allows using a high learning rate while keeping the network learning, which reduce the training steps. In addition, batch normalization allows to use saturating nonlinearities such as *tanh* and *sigmoid*, by preventing the network from getting stuck in the saturation mode (e.g. gradient equal to 0).

Rectified Linear Unit (ReLU). After each conv layer, the convention is to apply a nonlinear layer (or activation layer) immediately afterward. This was found to greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions due to its linear non-saturating form [140]. It also helps to alleviate the vanishing gradient problem, which is the issue where the lower layers of the network train very slowly because the gradient decreases exponentially through the layers. The ReLU layer applies the function $f(x) = \max(0, x)$ to all the values in the input volume. In basic terms, this layer just changes all the negative activations to 0.

Fully Connected layer takes an input volume (whatever the output is of the conv or ReLU or pool layer preceding it) and outputs an N dimensional vector where N is the number of classes the CNN is designed to predict. A fully connected layer takes the output of the previous layer (which represent the activation maps of high level features) and determines which features most correlate to a particular class and has particular weights such that when you compute the products between the weights and the previous layer, you get the correct probabilities for the different classes.

Deconvolutional layer. For classification purposes, the size of the feature maps becomes smaller in deeper layers of the network, resulting a prediction vector for the probability of an image to be classify as a certain class. A pixelwise prediction, such that classification for each pixel can be calculated, can be obtained by a deconvolutional layer [141]. Long et al. [142] used the deconvolutional layer and utilized bilinear upscaling to initialize the deconvolutional weights thus producing larger output map as illustrated in Fig. 22.

Fully Convolutional Network (FCN). Introduced by Matan et al. [143][30], the basic idea of a fully convolutional network is the size independence of the input image. This

can be thought of as if the fully connected layers of conventional CNNs are converted to convolutional layers with a specific filter size. The fully connected layers can be interpreted as convolutional layer with a filter size of the last feature map. For this reason, the two types of layer can easily be transformed by just copying the weights. The final output of an FCN is a feature map, which can produce different output feature map sizes in dependence of the input size.

Long et al. [142] introduced a fully convolutional network structure with deconvolutional layers, which was trained end-to-end for semantic segmentation. This combination of the layers facilitates a two-dimensional output of the neural network for each class, where the output has the same size as the input (see Fig. 22) and, as a result, a pixelwise classification is produced.

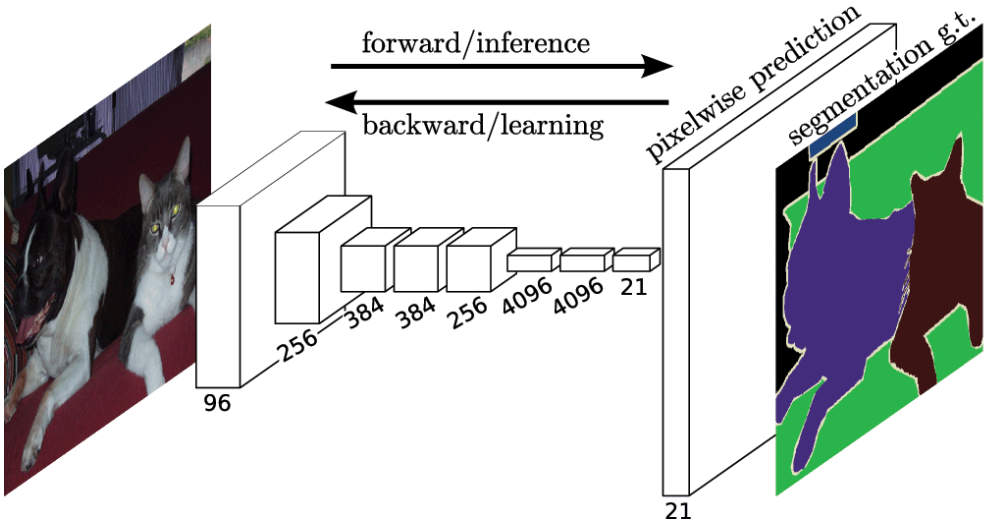


Fig. 22: Fully Convolutional Network for pixelwise semantic segmentation [142]

2.5.2 Training

The training process of the CNN is the most important part since it provides the weights for each layer. For training, both input image and ground truth data are available. All the network parameters are generally initialized with zero mean Gaussian random variables.

An image fed to CNN during the first forward pass will result with a prediction vector that will most likely provide equal prediction for all classes. To quantify the capacity of the network to approximate the ground truth labels for all training inputs, a loss function is defined such as Mean Square Error (MSE) or Cross Entropy.

MSE loss is defined as:

$$MSE(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i}\left|\mathbf{x}_i - \mathbf{y}_i\right|^2 \tag{35}$$

with an n-prediction $\mathbf{x}_i$ vector and a binary vector $\mathbf{y}_i$ of zeros except a one in the corresponding class dimension.

The Cross-Entropy loss is defined as:

$$CrossE\left(\mathbf{x},\mathbf{y}\right) = -\sum_i \mathbf{y}_i \cdot \log\left(\frac{\exp\left(\mathbf{x}_i\right)}{\sum_j \exp\left(\mathbf{x}_j\right)}\right) \tag{36}$$

While MSE give too much emphasis on the incorrect outputs, cross-entropy considers the closeness of a prediction and is a more granular way to compute error.

After computing the prediction and its associated loss for each example, we sum up all the loss to compute the final error. A backpropagation algorithm is then used to update the weights. To find which weights contribute mostly to error of the network, the partial derivatives $\partial E / \partial w$ of the cost function E for all weights w, is calculated. Once all the derivatives are computed, the weights can be updated using a chosen optimization technique such as Stochastic Gradient Descent (SGD) [144]. This back and forward process (forward pass, backward pass and optimization) constitute a single training cycle. This process is repeated for a specific number of times for each set to training images commonly known as a batch. Once finished, hopefully a low enough local minimum is found such that the network should provide good predictions [145].

3. RGB phase mask

3.1 Binary phase mask

Radially symmetric binary optical phase masks have been proposed for overcoming limitations set by OOF imaging [33]. A phase mask, incorporated in the optical train is meant to compensate for the OOF phase error by adding a constant phase shift near the aperture edge. Careful design of the phase level along a radial ring, results in an increased DOF. The drawback is that one gets a reduced contrast level when the image is in-focus. For computer vision applications, as for instance for barcode reading, those type of masks offers all-optical solution for EDOF.

To provide increased DOF for both positive as well as negative values of Ψ, those masks consisted of one or several rings, each exhibiting a phase shift of π radians for a center wavelength (as illustrated in Fig. 23). However, such phase mask provides the exact desired phase shift for a single wavelength only, while the phase shift for other wavelengths changes accordingly. Milgrom at el. [34] addressed this wavelength dependent issue by carefully designing the mask parameters as a function of the visible wavelength range such that the mask will provide EDOF for color cameras.

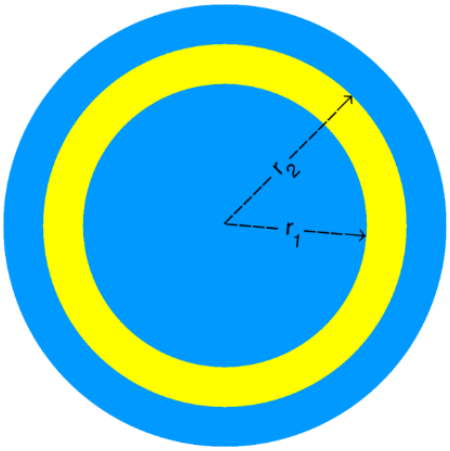


Fig. 23: Single ring phase mask

The phase shift is achieved by creating layers of depth *d* into a glass plate with a refractive index $n(\lambda)$. For a wavelength λ_1, the phase difference φ_1 created by the depth gap *d* can be expressed as:

$$\varphi_1 = \frac{2\pi d}{\lambda_1}\left[n(\lambda_1)-1\right] \tag{37}$$

To achieve a phase shift of π, the ring depth should be:

$$d = \frac{\lambda_1 \varphi_1}{2\pi\left[n(\lambda_1)-1\right]} = \frac{\lambda_1(2m+1)}{2\left[n(\lambda_1)-1\right]}; \qquad m \in \mathbb{Z} \tag{38}$$

The ring depth d was designed for a wavelength λ_1. Considering a low dispersion glass where $n(\lambda)$ is almost constant in the range of the visible light, the phase shift for a different wavelength λ_2, can be expressed as:

$$\varphi_2 = \frac{\lambda_1 \varphi_1}{\lambda_2} \frac{\left[n(\lambda_2)-1\right]}{\left[n(\lambda_1)-1\right]} \approx \frac{\lambda_1}{\lambda_2} \cdot \varphi_1 \tag{39}$$

The phase mask adds a constant phase ring to the imaging system aperture in order to compensate the phase error caused by OOF conditions. Both the defocus parameter Ψ (Eq. (20) as well as the cutoff frequency f_c depend on the wavelength. For consistency, both parameters will be referred with respect to the Blue wavelength $\lambda_1 = 450nm$. Thus, the defocus parameter Ψ_i for wavelength λ_i will be expressed as:

$$\Psi_i = \Psi \frac{\lambda_1}{\lambda_i} \tag{40}$$

Accordingly, the wavelength dependent cutoff frequency $f_c(\lambda_i)$ (Eq. (16)) will be expressed as:

$$f_c(\lambda_i) = f_c \frac{\lambda_1}{\lambda_i} \tag{41}$$

After setting the extent of the desired Ψ range that one would like to handle and the acceptable minimum contrast of the resulting image, one selects the number of phase rings to implement in the mask, and then determines the appropriate rings' radii by solving the following optimization process:

$$\max_{\boldsymbol{r}} \min_{\Psi \in DOF} \left[\nu : MTF(\nu, \boldsymbol{r}, \Psi, \lambda) = C_d\right] \tag{42}$$

where $\boldsymbol{r}$ is the mask radii vector, Ψ is the defocus measure, ν is the spatial frequency, and C_d is the desired acceptable minimum contrast value. The result of the optimization process expressed in Eq. (42) provides the ring radii that will maximize the minimum (worst) spatial cut-off frequency set at the minimum acceptable contrast level, along the entire DOF under consideration.

One should note that the cut-off frequency is determined by the acceptable minimum contrast value, rather than by the frequency with zero contrast, as commonly used [34]. This distinction provides a more realistic requirement, although it reduces the resulting DOF extent. The optimization goal is to extend the cut-off frequency, while at same time

maintaining an acceptable minimum contrast value; post-processing techniques could enhance existing contrast in the optical image, provided it is above a minimum level, restricted by the acceptable noise level. Image data with contrast below that level is considered lost, and cannot be restored 'a-posteriori'. One should note that different considerations in the design of the optical system are possible. Those will result in other optimization goal, as for instance maximizing the 'area under' the MTF.

3.2 Mask design for depth-sensitive PSF

Milgrom et al [35] proposed the use of a special RGB phase mask that exhibits significantly different response in the three major color channels R, G and B. It has been shown that each channel provides best performance for different depth regions, so that the three channels jointly provide an extended DOF. The mask considered in [35] provided a phase shift of 3π for a blue wavelength and consequently 2π phase shift for a red wavelength. Therefore, such mask did not affect the red channel imaging performance, obtained by virtue of the full-size aperture.

Unlike barcode objects, natural images exhibit mostly low spatial frequencies. Moreover, the purpose of the mask was to increase the diversity between the three main color channels R, G and B. Increasing the phase shift to several π, will increase the sensitivity to changes in wavelength (Eq. (39)), which should increase the diversity between colors. Unlike Milgrom et al [35] that used a 3π mask, we designed a 4π mask which provides 3π phase shift to the R channel. This means that the R channel will perform in a similar fashion as the B channel responded to the 3π mask. Using the 3π mask, it was expected that the B channel should provide acceptable contrast images for $\Psi = 6$ region. However, since Ψ depends on the wavelength (Eq. (40)), the R channel now exhibits higher values of Ψ, thus providing information for a higher value of the DOF. The MTF curves shown in Fig. 24, exhibit the response of the 4π mask that demonstrates an increased DOF as compared to that obtainable with a 3π mask. Note that the B channel response for $\Psi = 6$ using the 3π mask is similar to that of the R channel response for $\Psi = 8$ using the 4π mask.

It is instructive to examine the contrast level of a single spatial frequency (say $f_c / 4$) for different Ψ values. A comparison for three cases is presented in Fig. 25: 3π mask (left) and 4π mask (right), both with solid lines. The dashed lines in both plots present the curves for clear aperture. The 4π mask exhibits larger DOF than that of the 3π mask and provides wider separation of the three-color channels. This enhancement effect is crucial for the post processing stage as described in the next sections. It is instructive to provide the actual distances corresponding to the Ψ values used in Fig. 25. For instance, for an iPhone 6 rear camera (having a 1.8mm aperture), focused at a nominal distance located $1.4m$ away, the Ψ values range from (-4) to 8, corresponding respectively to object locations extending from infinity up to $50cm$ from the camera.

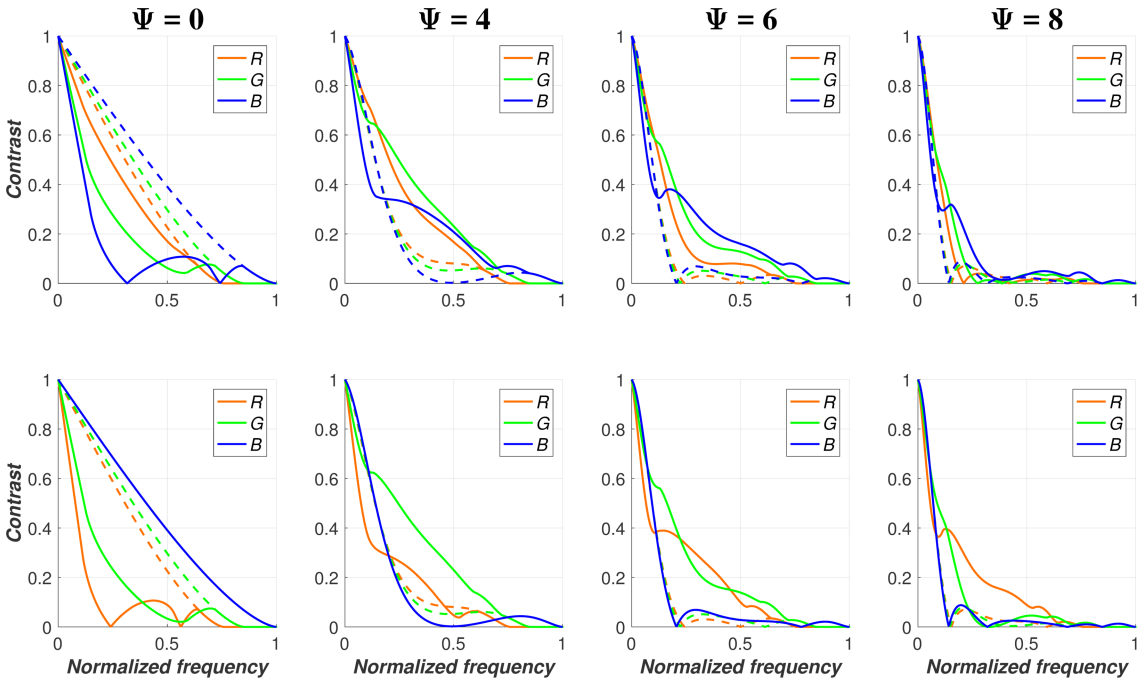


Fig. 24: Comparison between the MTF of aperture equipped with a phase mask of 3π (top – solid lines) and 4π mask (bottom – solid lines), for $\Psi = 0,4,6,8$ (left to right). The dashed lines in all the plots present the curves for clear aperture case.

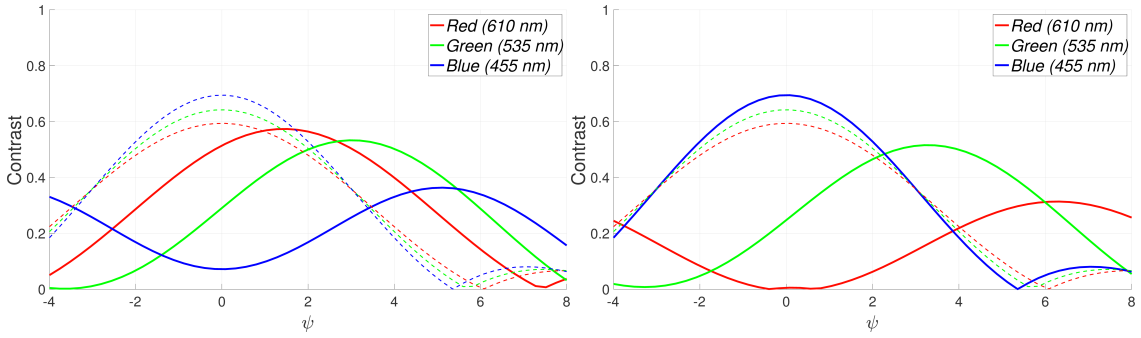


Fig. 25: Comparison between simulated MTFs of a single spatial frequency ($f_c / 4$) as a function of OOF factor Ψ. Solid line: aperture equipped with a phase mask of 3π (left) and 4π (right). The dashed lines in both plots presents the curves for clear aperture.

3.3 Mask optimization

An increased DOF was achieved by using a binary phase mask with a single step ring of 3π, as shown in [27], while the 4π mask, described in the previous section, offered superior extension of the DOF, as demonstrated in [36]. The purpose of those masks was to increase the response diversity of the three main color channels R, G and B, such that each will perform best for different depth regions. Finally, the three channels jointly provide an extended DOF.

Although this phase mask performed well in case of image deblurring, it was designed to be used in conjunction with a corrected lens and a specific range of Ψ 's. In this section, we investigate a new mask consisting of three step rings, each with a unique depth, thus a different phase, that provides more flexibility in designing an imaging system that can also be used in conjunction with an uncorrected lens.

As shown in [36], the chromatic diversity is the key element in finding the correct blurring model in a blind deblurring process; it also provides valuable information for the depth restoration process. The diversity between objects from different depths will be determined by the differences between the corresponding blurring models as expressed by the corresponding PSF's of the respective chromatic channels. A well designed chromatic separation optical system, where each color channel will have a Diffraction Limited (DLIM) performance for a different focal distance , will require a special lens design [38] and even then, for small scale cameras (as for smartphones), the lens performance must be reduced to meet the small form factor requirements.

The search for mask parameters (rings radii and their respective phase) is based on an analytical calculation of the derivative of the PSF with respect to Ψ . A similar calculation was also presented in [9] to optimize a cubic phase mask for a single wavelength such that the PSF will remain constant within the required DOF. For our purpose, we need to modify the search criteria such that the diversity between color channels will be maximized, while at same time maintaining the contrast above a minimum level.

Using the PSF expression from Eq. (12), the generalized PSF h_Ψ (Eq. (12)) can be expressed as a function of the generalized pupil P_Ψ (Eq. (21)):

$$h_\Psi\left(\mathbf{u};\lambda\right) = \mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}\cdot\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}} \tag{43}$$

where $\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}$ and $\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}}$ are the Fourier transform and the conjugate Fourier transform of P_Ψ respectively; $\mathbf{u}=\left(u,v\right)$ and $\mathbf{x}=\left(x,y\right)$are the image plane and pupil plane coordinates respectively.

Using the chain rule, the PSF derivative with respect to Ψ is:

$$\begin{aligned}\frac{d\left[h_\Psi\left(\mathbf{u};\lambda\right)\right]}{d\Psi} &= \frac{d}{d\Psi}\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}\cdot\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}} \\ &+ \mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}\frac{d}{d\Psi}\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}}\end{aligned} \tag{44}$$

As Ψ does not depend on the Fourier transform integration variables, one can interchange the order of Fourier transform with the derivative, such that:

$$\begin{aligned} \frac{d}{d\Psi}\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\} &= \frac{j}{R^2}\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\cdot r^2\right\} \\ \frac{d}{d\Psi}\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}} &= -\frac{j}{R^2}\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\cdot r^2\right\}} \end{aligned} \tag{45}$$

where $r^2=\left(x^2+y^2\right)$ and R is the pupil radius. Using Eq. (45), the PSF derivative form Eq. (44) is now reduced to:

$$\frac{d\left[h_\Psi\left(\mathbf{u};\lambda\right)\right]}{d\Psi} = \frac{2j}{R^2}\mathrm{Im}\left\{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\cdot r^2\right\}\cdot\overline{\mathbb{F}_{2D}\left\{P_\Psi(\mathbf{x};\lambda)\right\}}\right\} \tag{46}$$

Next, we define the PSF Derivative Energy (PSFDE) $E\left(\lambda,\Phi,\Psi\right)$ as the square of the derivative 2'nd norm by:

$$E\left(\lambda,\Phi,\Psi\right)=\left\|\frac{d\left[h_\Psi\left(\mathbf{u};\lambda;\Phi\right)\right]}{d\Psi}\right\|^2=\sum_s\sum_t\left(\frac{d\left[h_\Psi\left(s,t;\lambda;\Phi\right)\right]}{d\Psi}\right)^2 \tag{47}$$

where Φ is the phase mask parameters (radii and phases) and $\left(s,t\right)$ are the sensor plane coordinates. The PSFDE quantifies the PSF changes as a function of Ψ, but only for a single wavelength. For color images, we define the "Joint PSFDE", or $E_J\left(\Phi,\Psi\right)$, that quantifies the PSF changes for all three-color channels:

$$E_J\left(\Phi,\Psi\right)=\sum_{\lambda=\lambda_R,\lambda_G,\lambda_B}E\left(\lambda,\Phi,\Psi\right) \tag{48}$$

For a DLIM system $E_J\left(\Phi,\Psi\right)$ will be zero at focus (Ψ=0), but will exhibit high peaks around $\Psi=3$ (Fig. 26 – left) For an uncorrected lens with spherical aberrations (Fig. 26 – right), the PSF will deteriorate and thus the difference between high and low values of $E_J\left(\Phi,\Psi\right)$ will be much smaller in comparison to DLIM systems.

For best coverage for a specific Ψ range, $E_J\left(\Phi,\Psi\right)$ should be as high as possible for all Ψ values, thus insuring high diversity while keeping at least one channel sharp enough for the deblurring process.

We can thus define a merit function M:

$$M\left(\Phi\right)=\max_{\Phi'}\left[\min_{\Psi}\left\{E_J\left(\Phi',\Psi\right)\right\}\right] \tag{49}$$

The merit function $M\left(\Phi\right)$ provides the maximum value of the minimal $E_J\left(\Phi,\Psi\right)$ over the range of Ψ. For a DLIM system $M\left(\Phi\right)$ is zero, since the minimal value of $E_J\left(\Phi,\Psi\right)$ is zero for $\Psi=0$; for an uncorrected lens, $M\left(\Phi\right)$ will still be low (Fig. 26).

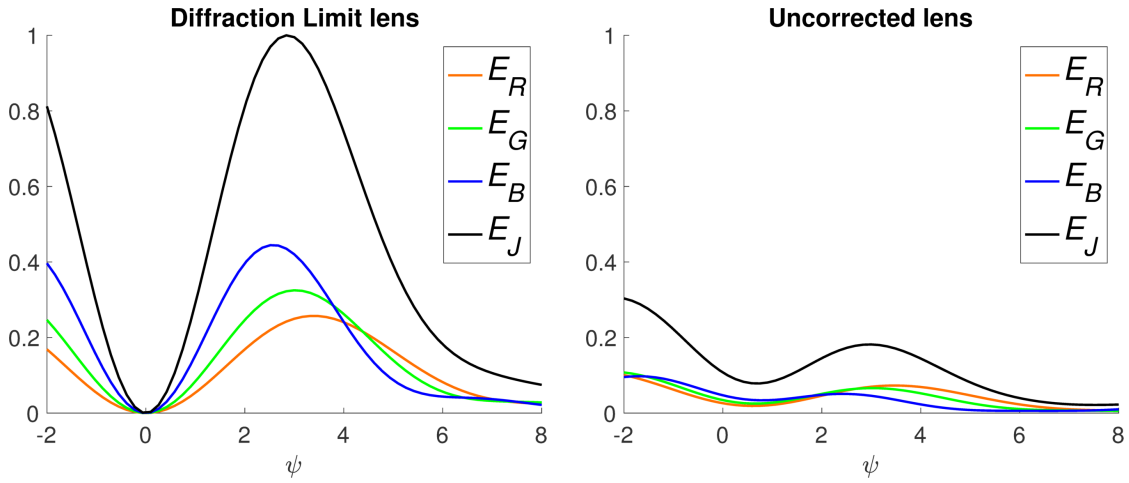


Fig. 26: The Joint PSFDE: DLIM system (left); Lens with spherical aberrations (right). The Zernike coefficient Z_8 was set to 1.8 (see Table 1)

As mentioned at the beginning of this section, we seek a perfectly designed lens that provides perfect DLIM performance for each color channel separately, while each color is focused at a different distance. This will be used as a test case. As shown in Fig. 27(a) such theoretical lens will provide proper color separation within the Ψ range, which was set to $\Psi=[-2,8]$. Our phase mask performance, shown in Fig. 27(b), produces similar result as that of a perfectly designed lens (theoretical lens) with chromatic aberration only. When using an uncorrected lens with spherical aberrations, equivalent results have been achieved using different mask parameters as shown in Fig. 27(c).

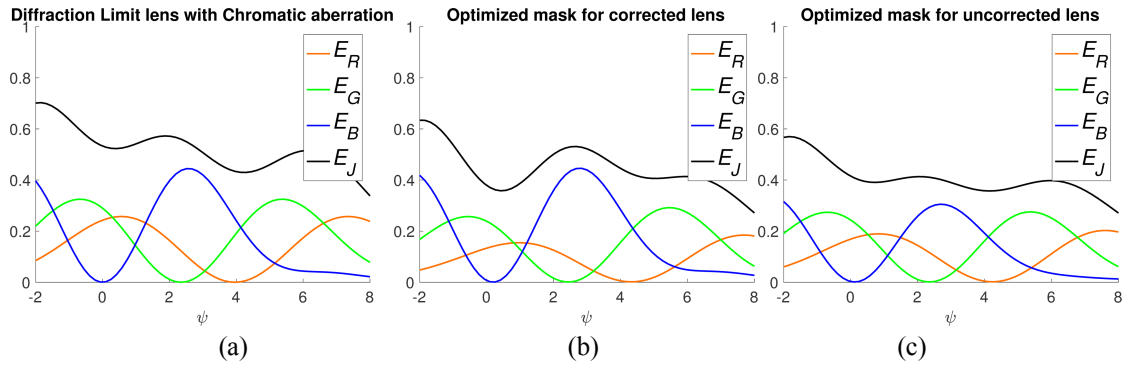


Fig. 27: The Joint PSFDE: (a) DLIM with chromatic aberrations; (b) optimized mask for a corrected lens; (c) optimized mask for uncorrected lens.

For the DLIM corrected lens the mask radii vector was set to $\mathbf{r}=(0,0.39,0.84,0.93,1)$ with respect to the maximal aperture radius, and with phase value of $\phi=(0,6.3,12.4,12.3)[rad]$ (each value refers to the phase of the ring between adjacent radii values. For the DLIM uncorrected lens, the mask radii vector was set to be the same

as for the corrected lens with a phase-mask, but the phase values changed to $\phi = (0, 8.2, 14, 12.6)[rad]$. A 3D illustration of the second mask is presented in Fig. 28.

The phase mask parameters search was limited to a maximum phase shift of 5π radians such that the relative phase between wavelength inside the visible range (Eq. (39)) will not exceed 1.2π. Since the sensor color filters transfer colors near the peak wavelength, this phase constraint insures that the PSF will not change significantly between the three main color channels.

In practice, to make the fabrication process simpler and more reliable, a two-ring limitation was set in the searching process. An optimized three-rings mask surpass the two-ring mask only by a small margin. Such a design may be considered in the future for other optical system setups.

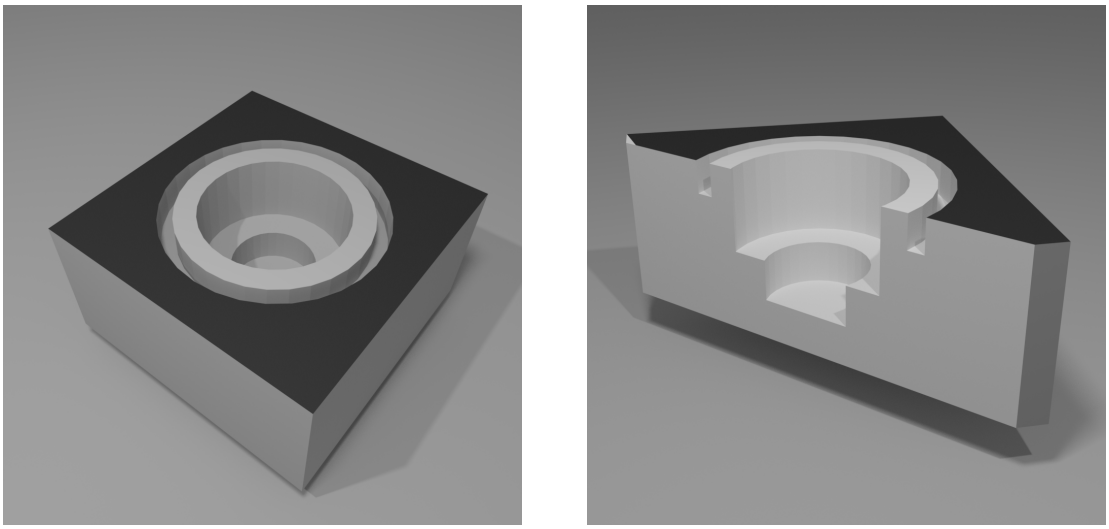


Fig. 28: 3D mask illustration. The phase difference was created by etched sections on a glass surface. The right image shows a cross-section of the mask.

The search for the mask parameters can be computationally exhausting since for each mask one needs to calculate the derivative (which involves a Fourier transform operator) for each color and for each Ψ within the range. This search can be simplified by analytical calculations as described in detail in Appendix A.

3.4 Spherical aberration compensation mask

In the previous section, a method for mask optimization was presented. When facing an imaging system using an uncorrected lens, this procedure provides a mask exhibiting similar results to the mask that has been used when dealing with DLIM lens.

One may consider designing a mask for an uncorrected lens as the combination of two masks in tandem; the first mask has the task of reducing the lens aberrations and the second one creates the necessary color separation. This observation led to another application of the phase mask as a way to correct aberration. This can also eventually reduce the complexity, thickness and even cost of conventional lens manufacturing.

The search for the optimal mask for reducing aberration is based on the same concepts presented in the previous section. The Joint PSFDE for DLIM lens, as shown in Fig. 27(a), exhibits zero response for $\Psi = 0$ (in focus) and high response at about $\Psi = \pm 3$. We define a new merit function:

$$M_{aberr}(\Phi) = \max_{\Phi'}\left[\max_{\Psi}\left\{E_J(\Phi',\Psi)\right\} - E_J(\Phi',\Psi=0)\right] \tag{50}$$

This merit function finds the best mask that will provide high Joint PSFDE for one Ψ within the range, while ensuring that the Joint PSFDE value for $\Psi = 0$ will remain low. The MTF for an in-focus system suffering from spherical aberration is presented in Fig. 29. The dash line in both plots shows the DLIM response. Imaging with a phase mask in the lens assembly exhibits near DLIM performance, while the uncorrected system exhibits poor results. The Zernike coefficient Z_8 (Table 1) was set to 1.8 (the estimated value for the spherical aberration found in one of our experimental lens).

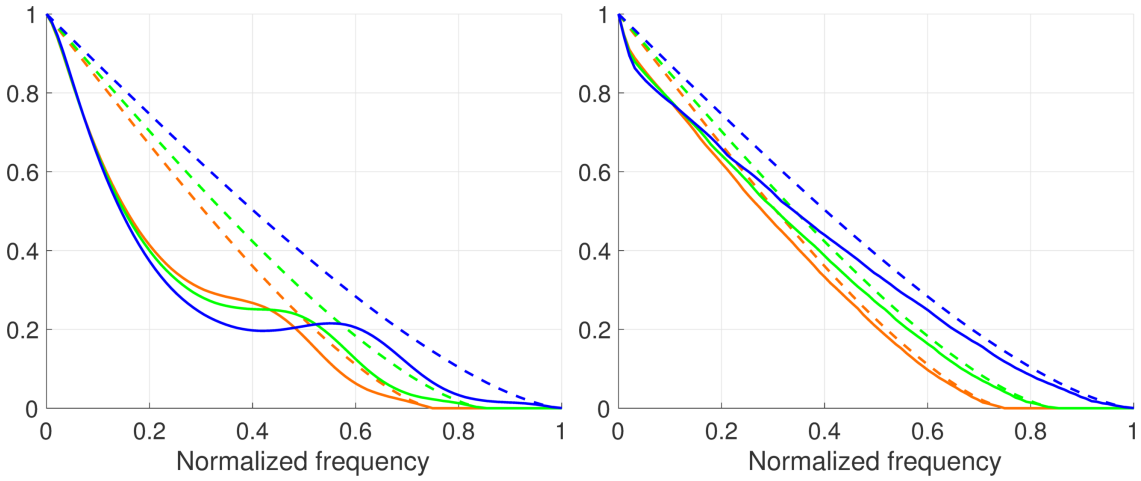


Fig. 29: MTF for in-focus system with aberration (left: solid line) and same system equipped with a correcting phase mask (right: solid line). The dash lines in both plots show the DLIM response.

To demonstrate the effect of the mask on an image, a simulation study of a test image using a DLIM lens, uncorrected lens, and uncorrected lens equipped with a phase mask, was carried. The results presented in Fig. 30 confirm the mask ability to compensate for optical aberrations.

Fig. 30: Imaging with DLIM lens (left – PSNR 26.3dB), with uncorrected lens (center – PSNR 23.7dB) and with a phase mask (right – PSNR 25.4dB).

3.5 Mask fabrication

Most of the imaging system designs used in this work require a diffractive optical element to manipulate the wave-front of the field in the exit pupil. The process was carried at Tel-Aviv University Nano Center using Lithography techniques and chemical wet etching.

This fabrication technique is very similar to the technology used to produce VLSI circuits [146]. In this method, to fabricate N rings phase mask one required $N+1$ binary lithography masks. Each binary mask defines the regions of the substrate that will be exposed during the Photo-Lithography, i.e. each pixel in the mask is either transparent or opaque.

After developing the first layer, the photoresist is removed to expose the designated areas. Hydrofluoric acid is then used to etch the glass such that a depth difference is created between the exposed area and the rest of the substrate, and thus, creating the desired phase difference. This process is repeated for each binary mask until the desired etched depth for the entire area is achieved.

A 130μm thick Soda Lime glass substrate was used for fabrication. This thin glass allowed incorporating the phase mask with a commercial lens with minimal change to the original lens design.

3.6 Chapter summary

In this section the use of a specially designed phase mask was presented. This simple optical element can be easily fabricated, and its small dimensions allow implementation in existing commercial cameras. The search for the optimized mask parameters has proven to be a powerful tool in the attempt of designing the optical stage of our phase coded computational camera. In the next chapters, the valuable information exhibited by the output image, will be exploited for image deblurring and depth estimation.

4. Sparse model for image deblurring and depth estimation

In this chapter a method for extended depth of field imaging with depth estimation capabilities is presented [36], [47]. This method is based on image acquisition through a thin RGB binary phase plate, which was presented in the previous chapter, followed by fast automatic computational post-processing which utilized sparse representability of natural images. By placing a wavelength-dependent optical mask inside the pupil of a conventional camera lens, one acquires a unique response for each of the three main color channels, which adds valuable information that allows blind reconstruction of blurred images. Simulation as well as capture of real-life scene show how acquiring a one-shot image focused at a single plane, enables generating a de-blurred scene over an extended range in space, in addition to producing a depth estimation map. This process was also implemented on an FPGA module [147] which enabled real-time performance.

4.1 Outline

This chapter organized as follows:

Section 4.2 is dedicated to introducing the foundation for the non-blind procedure which is used in Section 4.3 for spatially varying blind deblurring. Section 4.4 presents real-life experimental results acquired with a prototype camera equipped with our RGB phase mask. An FPGA-based real-time implementation of EDOF image reconstruction is demonstrated in Section 4.5. Section 4.6 is dedicated to describing the depth evaluation method which adds refocusing abilities to our system. Finally, Section 4.7 summarizes the chapter.

4.2 Sparse model for non-blind image deblurring

4.2.1 Image deblurring using a sparse synthesis pair

Sparse representation has proven to be a strong prior for non-blind image deblurring (refer, e.g., to [87], [88], [92]–[94]) where the blurring kernel is known, as well as for blind ones [89], [90]. The signal $\mathbf{x}$ is said to admit a sparse representation (or, more accurately, an approximation) over a dictionary $\mathbf{D}$ if one can find a vector $\mathbf{z} \in \mathbb{R}^N$ with only a few non-zero coefficients, such that $\mathbf{x} \approx \mathbf{D}\mathbf{z}$. The sparse representation pursuit problem can be cast as the ℓ_0 pseudo-norm minimization,

$$\mathbf{z} = \arg\min_{\mathbf{z}_i} \left\|\mathbf{x} - \mathbf{D}\mathbf{z}\right\|_2^2 \qquad \text{s.t. } \left\|\mathbf{z}\right\|_0 < S \tag{51}$$

where $\left\|\mathbf{z}\right\|_0$ counts the number of non-zeros in the vector $\mathbf{z}$ with the sparsity limited to S.

While the sparse representation pursuit (Eq. (51)) is computationally intractable, it can be efficiently approximated by several known techniques such as OMP [114]–[116]. The dictionary $\mathbf{D}$ can be constructed axiomatically based on image transforms such as DCT

or wavelet, or learned from a training set sampled from representative images. Here, we adopt the latter approach to construct a structured dictionary for encoding overlapping 8×8 patches represented as 64-dimensional vectors.

Assuming we have a sharp image *x* that has been blurred by a known kernel *h,* the blurred image *y* can be described as:

$$y = h * x + \eta \tag{52}$$

where η is additive Gaussian noise and '$*$' denotes the convolution operator. In the scope of this work we assumed that $\eta < 1$. Consider a patch from the blurred image represented in a raster scan order as a column vector $\mathbf{y}_i \in \mathbb{R}^{64}$, and a blurred dictionary $\mathbf{D}_b \in \mathbb{R}^{64\times N}$ composed of *N* 64-dimensional atoms generated from the sharp atoms in dictionary $\mathbf{D}_s \in \mathbb{R}^{64\times N}$, such that $\mathbf{D}_b = \mathbf{H}\mathbf{D}_s$, where the matrix $\mathbf{H} \in \mathbb{R}^{64\times 64}$ represents a blurring operator corresponding to the blurring kernel *h*.

Using OMP the sharp patch $\mathbf{x}_i \in \mathbb{R}^{64}$ can be restored by greedily solving the optimization problem

$$\mathbf{z}_i = \arg\min_{\mathbf{z}_i} \left\|\mathbf{y}_i - \mathbf{H}\mathbf{D}_s\mathbf{z}_i\right\|_2^2 \qquad \text{s.t. } \left\|\mathbf{z}_i\right\|_0 < S \ , \tag{53}$$

where $\mathbf{z}_i \in \mathbb{R}^N$ is the is the sparse vector corresponding to the *i*-th patch $\mathbf{y}_i$. Solution of (53) produces the sparse code coefficients $\mathbf{z}_i$ of the blurred patch $\mathbf{y}_i$ as a linear combination of atoms from the blurred dictionary $\mathbf{D}_b$. The sharp patch $\mathbf{x}_i \in \mathbb{R}^{64}$ can be recovered by $\mathbf{x}_i \approx \mathbf{D}_s\mathbf{z}_i$. This process implies that for all *i*, $\mathbf{y}_i \approx \mathbf{H}\mathbf{D}_s\mathbf{z}_i \approx \mathbf{D}_b\mathbf{z}_i$.

To reduce transition artifacts, patches are taken from the blurred image with a step size of one pixel to produce maximum overlap.

4.2.2 Dictionary selection

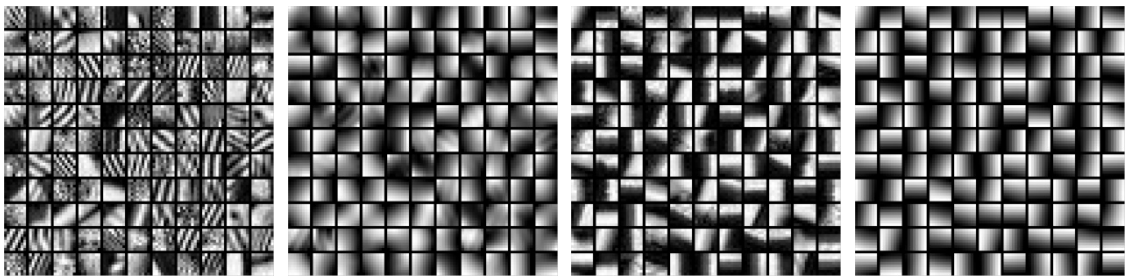
Blurring the dictionary atoms directly in the non-blind image deblurring setting asserts that for all *i*, $\mathbf{D}_s\mathbf{z}_i$ is a good estimation of the sharp patch $\mathbf{s}_i$. The relation between the clean and the blurred dictionaries allows restoring the original image from the blurry one. However, in real settings, only the blurry patch $\mathbf{y}_i$ is known and there is no guarantee that the coefficients vector $\mathbf{z}_i$ will yield a good approximation the sharp patch $\mathbf{x}_i$.

The dictionary learning process is the key element in the restoration process and many studies have focused on different learning processes [127], [148]–[150].Our approach for this problem is based on selecting a set of sampled data from training images [95] and using *k*-SVD [127] taking the imaging optics into account. The blur kernel transfers in

most cases only the low frequency in a specific patch while suppressing most of the high frequencies as shown in Fig. 31. By choosing only the low frequency patches for our dictionary, the corresponding blurred patches will still resemble the sharp ones and there will be a much better chance of choosing the correct sparse code coefficients.

The selection process is employed as follows: (1) a temporary dictionary is selected randomly from a set of sampled images; (2) the dictionary is learned to fit a sampled data using the *k*-SVD process; (3) only a few low frequency edges-like atoms are selected from the temporary dictionary; (4) the process is repeated until the required number of atoms is selected. For fast implementation of the OMP process [87], [100] each atom was normalized. Our results show that even if one uses a fixed small dictionary (128 atoms) it works as well as a large one and can also be used for other arbitrary natural scenes without repeating the training process again each time.

The restoration process using random patches and low-frequency patches is shown in Fig. 32. The random dictionary is composed of patches which contain high frequency noise and, as a result, the restoration process enhances the noise in the image as shown in Fig. 32. Our specially selected dictionary (which is composed out of low frequencies patches) restored the blur image without enhancing the noise and with fewer artifacts.



(a) (b) (c) (d)

Fig. 31: Dictionaries comparison: (a)-(b) – randomly selected dictionary before and after imaging; our "low frequency" dictionary before (c) and after (d) imaging.

(a) (b) (c) (d)

Fig. 32: Non-blind restoration example: (a) – original image, (b) blurred image, (c) restoration using random patches, (d) – restoration using our dictionary.

4.2.3 RGB dictionary

Assuming approximately constant depth and, hence, the same known Ψ for all pixels in the vicinity of a spatial location ξ in the sensor plane, the image formation model can be described as the following convolution:

$$\begin{aligned} y^R(\xi) &= \left(h_\Psi^R * x^R\right)(\xi) + \eta^R(\xi) \\ y^G(\xi) &= \left(h_\Psi^G * x^G\right)(\xi) + \eta^G(\xi) \\ y^B(\xi) &= \left(h_\Psi^B * x^B\right)(\xi) + \eta^B(\xi) \end{aligned} \tag{54}$$

where x^i, i=R,G,B, denote the color channels of the ideal image, y^i denote the corresponding OOF image in the sensor plane, h_Ψ^i is the color depended PSF kernel and η^i is the additive white Gaussian noise at least at reasonable illumination intensities. In vector notation combining the three channels, the expression simplifies to

$$\mathbf{y}(\xi) = \left(\mathbf{h}_\Psi * \mathbf{x}\right)(\xi) + \boldsymbol{\eta}(\xi). \tag{55}$$

Similar to the gray level dictionary the was presented in the previous section, the RGB dictionary is composed from sampled color images using an $8\times8\times3$ patch size (8×8 for each color channel). Each atom in the RGB sharp dictionary $\mathbf{D}\in\mathbb{R}^{192\times N}$ is formed by concatenating the R, G and B vectors. Using the previous formulation, a sparse vector $\mathbf{z}_i$ corresponding to a blurred patch $\mathbf{y}_i\in\mathbb{R}^{192}$ is estimated by the following optimization problem:

$$\mathbf{z}_i = \arg\min_{\mathbf{z}_i} \left\|\mathbf{y}_i - \mathbf{H}_\Psi \mathbf{D}\mathbf{z}_i\right\|_2^2 \qquad \text{s.t. } \left\|\mathbf{z}_i\right\|_0 < S \tag{56}$$

where the matrix $\mathbf{H}_\Psi\in\mathbb{R}^{192\times192}$ represents a blurring operator corresponding to the blurring kernel h_Ψ.

Unlike the gray level case, color patches have three bias components (the signal mean value, or DC), each one is responsible for the way the color appears and therefor it's value must be reserved. However, the OMP process does not guarantee that the restored patch will maintain the same bias as the original one. Expending the dictionary by adding similar atoms but with different biases can reduce the bias restoration error, but this will increase computational efforts. In [96] the authors address this issue by defining a new inner product of two column vectors $\mathbf{x},\mathbf{y}\in\mathbb{R}^{192}$ to use in the OMP step as:

$$\left\langle \mathbf{y},\mathbf{x}\right\rangle = \mathbf{y}^T\mathbf{x} + \gamma\mathbf{y}_m^T\mathbf{x}_m, \tag{57}$$

where the first term represents the classic inner product and the second one represents the inner product of the mean color vector $\mathbf{x}_m,\mathbf{y}_m\in\mathbb{R}^3$ which contain the mean value of each

color channel. The parameter γ controls the importance of the bias correction. This essentially force selecting atoms with the same color bias.

This scheme may be suitable for denoising where the bias may change drastically due to severe noise, but in optical deblurring the bias usually remains the same since the 'DC' value is not affected by the blurring process. In addition, the suggested scheme will encourage bias restoration instead of texture restoration. This may cause a problem when using our phase mask; for instance, when the Red channel exhibits high contrast level, but the input has low bias level while the Blue channel exhibits low contrast with high bias input. This scenario will encourage atoms selection based on bias and not on details which not only reduce the restoration performance but more importantly, will limits our ability to distinguish between atoms from one focus scenario to another, which is the basis for our blind restoration process (as will be presented in the next section).

For our deblurring purposes, the mean value was removed from each color separately and the entire atom was normalized afterwards. At the representation stage of the blurred image (Eq. (53)) the DC information was removed from each color channel patch to insure that only the texture is being represented without the influence of strong DC data. The removed DC is added to the recovered patch after it has been restored using the dictionary switching method. This allows us to restore the correct bias using a relatively small dictionary (for our simulations, 128 atoms were sufficient). An illustration of a colored dictionary we used is presented in Fig. 33. Notice that the colors appearing in Fig. 33 do not reflect the actual bias (since it was zero); random bias values were added for illustration purposes.

Fig. 33: Example of a color dictionary.

4.2.4 Non-blind color image deblurring using a phase mask

Before attempting to restore a color image blindly, it is instructive to assess the performance and limitations of such restoration for a non-blind case. The blurring kernel depends on the optical cut-off frequency f_c (Eq. (16)) as well as the out-of-focus parameter Ψ. As described in Section 3, for the in-focus condition the phase mask deteriorates the image in comparison to the clear aperture with no mask, as easily

observed in Fig. 25. Therefore, one must show that the blurred image captured with the phase mask can be restored correctly when the image is in-focus.

Fig. 34 (top row) shows the restoration results for in-focus images taking with and without phase mask. The results show that even though the blurry input image taken with the phase mask is worse than the blurry input taken without the mask, the resulting restored images in both cases is sharp.

The advantage of the phase mask comes into play when one deals with a blurry image due to a strong out-of-focus factor. Fig. 34 (bottom row) shows the results of the restoration process for such case. Using the phase mask in the imaging process creates a better input for the restoration stage with much more information to work. The simulation results in Fig. 34 clearly show that for a strong out-of-focus condition image restoration will not work unless one uses a phase mask.

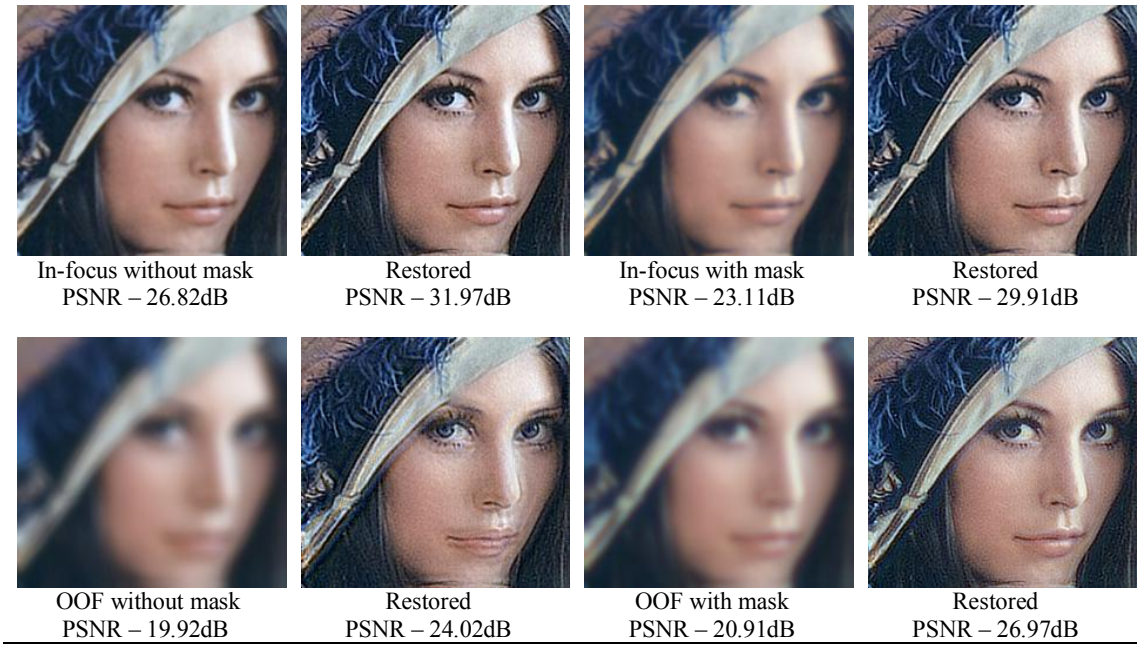


Fig. 34: Restoration results for in-focus blurring and restoration (top row) and strong OOF condition (bottom row), with clear aperture (a-b) and with phase mask (c-d)

4.3 Blind image deblurring using a phase mask

4.3.1 Blind image deblurring model via stacked dictionary

In the practical case of blind image deblurring, the blur kernel is unknown; moreover, it varies with the object depth location. Many studies dealt with this problem, with limited success. For instance, using different iterative processes [89], [90], [92]–[94], [151]–[153], one tries to estimate the blurring kernel so that image restoration can be thereafter achieved. Reconstruction processes usually require high computational complexity,

which limits their use for many real-time applications. Our approach based on using an optical phase mask allows restoring the image without needing any iterative process to estimate the blurring kernel. Furthermore, most of blind deblurring algorithms are constructed with the explicit assumption that the input image is localized at a single depth position, such that the blur can be described as a convolution with a spatially constant kernel. In real life scenes, however, the objects are at different distances, and the blur changes abruptly when crossing the object boundaries. Many natural scenes can be approximated by a "2.5D world" assumption asserting that a scene comprises a plurality of objects at different depths, yet each object is approximately planar and perpendicular to the optical axis. Under this assumption, the blur can be modeled as the convolution with a piece-wise constant blur kernel defined by the defocus parameter Ψ of that particular object. We also assume that the blurring kernel is affected only by the defocus parameter ($\Psi_{\min}$ to $\Psi_{\max}$), ignoring other sources of blur, e.g., due to camera or scene motion.

To cover most DOF, we propose to construct *k* sub-dictionaries using different blurring kernels for different defocus parameters and then concatenate them into a single "Multi-Focus Dictionary" $\mathbf{D}_{\Psi}$:

$$\mathbf{D}_{\Psi} = \left(\mathbf{H}_{\Psi_1}\mathbf{D},...,\mathbf{H}_{\Psi_k}\mathbf{D}\right) = \left(\mathbf{D}_{\Psi_1},...,\mathbf{D}_{\Psi_k}\right) \tag{58}$$

For this application, we used $\Psi_{\min} = 0$ and $\Psi_{\max} = 8$. Assuming that a step of $\Delta\Psi = 1$ is discriminative enough, one thus needs to construct nine sub-dictionaries.

Similar to the non-blind search in Eq. (56), the blind problem can be address as finding the sparse vector $\mathbf{z}_i \in \mathbb{R}^{k\cdot 192}$ representation using the concatenate dictionary $\mathbf{D}_{\Psi}$:

$$\mathbf{z}_i = \min_{\mathbf{z}} \left\|\mathbf{y}_i - \mathbf{D}_{\Psi}\mathbf{z}_i\right\|_2^2 \qquad \text{s.t. } \left\|\mathbf{z}_i\right\|_0 < S \tag{59}$$

The reconstruction of the sharp vector $\mathbf{x} \in \mathbb{R}^{192}$ preformed using a k-times concatenated version of the sharp dictionary as $\mathbf{x}_i = \left(\mathbf{D},...,\mathbf{D}\right)\mathbf{z}_i$. This process is based on a strong assumption that elements from the correct sub-dictionary will be selected in the pursuit process but in general, there is no way to ensure that this will be the case.

The OMP process chooses elements from the dictionary that best match the input patch, based on largest inner product, treating each RGB input as a single input vector. For a specific value, when using a corrected lens with clear aperture, the PSF kernel is very similar for all color channel. Alternatively, when using our RGB mask, the PSF kernel is different for each color channel such that for an input patch (or vector) the contrast level varies strongly for each color channel. The response is unique for each OOF scenario and therefore the blurred input vector will most likely associate with blurred vectors from that experience the same blurring process. The diversity exhibited by the color information when using a phase mask allows applying the non-blind deblurring technique described

in the previous section directly. The advantage of the DC removal process from the input patches (which was also discussed in the previous section) is now clear as strong bias in one color channel will not affect the atoms selection process.

Comparing our algorithm with an open access state-of-art algorithm, such as the one provided by Krishnan et al. [151], our process produced better results when applied to natural images out of the Kodak Dataset [154]. We also run the process on texture images from the Colored Brodatz Texture Database [155] and observed similar performance, hinting that the MFD dictionary will work on almost any natural scene. The example in Fig. 35 shows how an OOF image taken with a conventional clear aperture [Fig. 35(b)] cannot be restored using Krishnan algorithm [151] [Fig. 35(c)]. When using a phase mask the image is visually better [Fig. 35(d)] than that of a clear pupil, but the restoration process applying Krishnan [151] still introduces strong artifacts [Fig. 35(e)]. However, applying our process on the image taken with the phase mask one gets an improved sharp image [Fig. 35(f)].

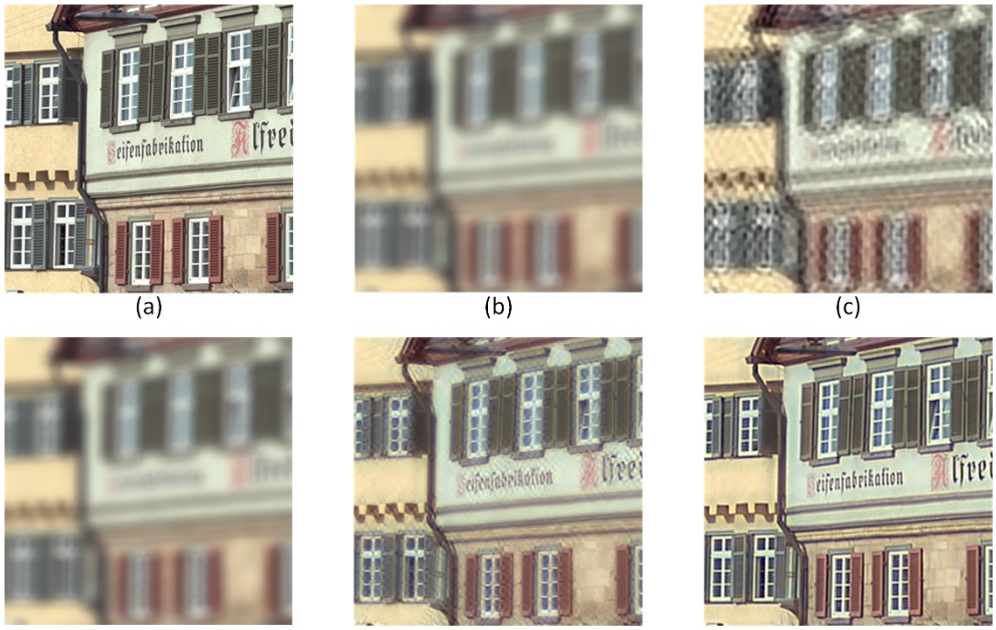


Fig. 35: Example of simulated image blurring and restoration. (a) Original image from the KODAK dataset ; (b) Out of focus with clear aperture (PSNR – 15.16dB); (c) Deblurring of (b) using Krishnan [151] *(PSNR – 14.82dB); (d) Out of focus with phase mask (PSNR – 16.37dB); (e) Deblurring of (d) using Krishnan* [151] *(PSNR – 19.22dB); (f) Deblurring of (d) using our new process (PSNR – 23.04dB).*

4.3.2 Spatially varying blind blurring

As indicated in the previous section, for natural depth scenes one cannot assume that the image is blurred by a single blurring kernel. Our process analyzes small patches and not the whole image; therefore, the restoration process is applied to every region inside the

image frame independent on the restoration process of other regions in the frame. This feature is what allows implementing our process directly in one step on images with spatially varying blur.

To demonstrate the process let us assume a 2.5D scene with four objects, each located at a different distance from the camera as shown in Fig. 36. The top image shows the simulation of capturing a scene using a conventional camera focused on the background. As expected, other objects in the scene are blurred according to their distance from the focus point. The bottom image presents the result obtained with an imaging system comprising a phase mask followed by our post process blind restoration scheme. One can notice that all objects were restored without noticeable artifacts. Although there is a strong red content in the front objects and blue one in the background objects, a reversed scenario has also been examined producing similar results, thus showing the robustness of our system.

To reduce running time, we examined the minimal number of sub-dictionaries that will still provide good results in comparison to the results shown in Fig. 36. To cover the full range of OOF factors inside the scene one will need to use at least three sub-dictionaries (corresponding to $\Psi = 0, 4, 8$) without losing any noticeable quality.

Fig. 36: Simulated 2.5D scene with four objects each located at a different distance from the camera corresponding to $\Psi = 0$ (background buildings) to $\Psi = 8$ (the woman on the right) – Conventional imaging (top). Imaging with our system using a phase mask and blind post-processing (bottom).

4.4 Experimental stage

4.4.1 Demosaicing implementation

In the previous sections, the sparse representation (Eq. (59)) was based on an imaging model (Eq. (55)) which assumes the full RGB information is available from the blurred image i.e. the image was demosaiced prior to the deblurring process. This assumption holds, since camera manufacturers usually provide the demosaiced version of the image. However, the proprietary methods used for producing those demosaiced imaged are not available. Since the deblurring performance depends on the imaging model, adding a hidden black box to the model might damage the restoration quality. Moreover, several studies [96] had successfully implemented the demosaicing stage as part of the representation process, and thus enable faster process which also required less memory.

The color information is sampled using a color filter array (CFA) where each pixel captures information of only one of the main color channels (RGB). Combing the three channels using vector notation and adding the CFA model, the raw mosaiced image $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{B} \cdot \left(\boldsymbol{h}_{\Psi} * \mathbf{x} \right) + \boldsymbol{\eta}, \tag{60}$$

where $\mathbf{x}$ is the three-color ideal sampled image and $\mathbf{B}$ is the Bayer sampling matrix such that each pixel in $\mathbf{y}$ contains information of only one of the three-color channels. Note that the action of $\mathbf{B}$ is not shift-invariant.

Implementing this model within the sparse representation scheme (59), produce the following single-step demosaicing-deblurring optimization

$$\mathbf{z}_i = \arg\min_{\mathbf{z}_i} \left\| \mathbf{y}_i - \mathbf{B}\mathbf{D}_{\Psi}\mathbf{z}_i \right\|_2^2 \qquad \text{s.t. } \left\| \mathbf{z}_i \right\|_0 < S. \tag{61}$$

For a 64-dimension input vector $\mathbf{y}_i$ (a 8×8 mosaiced patch), $\mathbf{B}$ is the 64×192 matrix performing the action of CFA on the 192-dimension atoms from the dictionary $\mathbf{D}_{\Psi}$. Notice that the restoration step remains the same as $\mathbf{x}_i = \left(\mathbf{D}, ..., \mathbf{D} \right) \mathbf{z}_i$ perform both the deblurring and the demosaicing process.

As mentioned in the beginning of this chapter, patches are taken from the blurred image with a step size of one pixel to reduce transition artifacts. In typical signal with a 2×2 CFA arrangement, $\mathbf{B}$ is only invariant to even shifts. To produce maximal overlap in the model, the input raw image needs to be divided into four different images corresponding to 0- and 1-pixel shifts along each axis. Assuming the CFA is arranged in the '*rggb'* format, the other three will be arranged as *'gbrg'*, *'grgb'* and *'bggr'* respectively. After each image is deblurred and demosaiced separately using two-pixel step size, the four images are combined into a fully overlapping image. Alternately, one can also use only the first image while reducing a bit the smoothness of the transition between patches.

4.4.2 Setup and results

A proof-of-concept experimental system aiming to test the approach, described in this chapter, was carried out. The first-generation system consisted of a CCD camera (Allied Vision *G-146)* with a pixel size of *4.65μm* and a *16mm* lens (*Computar M1614-MP2*) into which the phase mask was inserted. The mask was manufactured on a *1.5mm* thick Soda-Lime substrate onto which a single ring, 4π phase pattern was etched in the center to provide the necessary phase shift. A view of the scene setup, onto which we marked the value corresponding to the object position is shown in Fig. 37.

A comparison between a conventional camera and our method is presented in Fig. 38. Both images were captured in the exact same lighting conditions and exposure time. The left image in Fig. 38 shows the captured scene with a conventional lens (clear aperture) that was focused on the background poster. The right image in Fig. 38 shows the results of capturing the same scene using an aperture with a phase mask (see Fig. 37) followed by our post processing algorithm. One can notice that using the proposed method one restores an image with all objects in focus. The zoomed sections in Fig. 38 allow the viewer to observe the restored image quantitatively with the use of the resolution target object. Notice that the conventional captured Rosetta image exhibits low contrast or even contrast reversal, as opposed to the sharp restoration results one gets using our system.

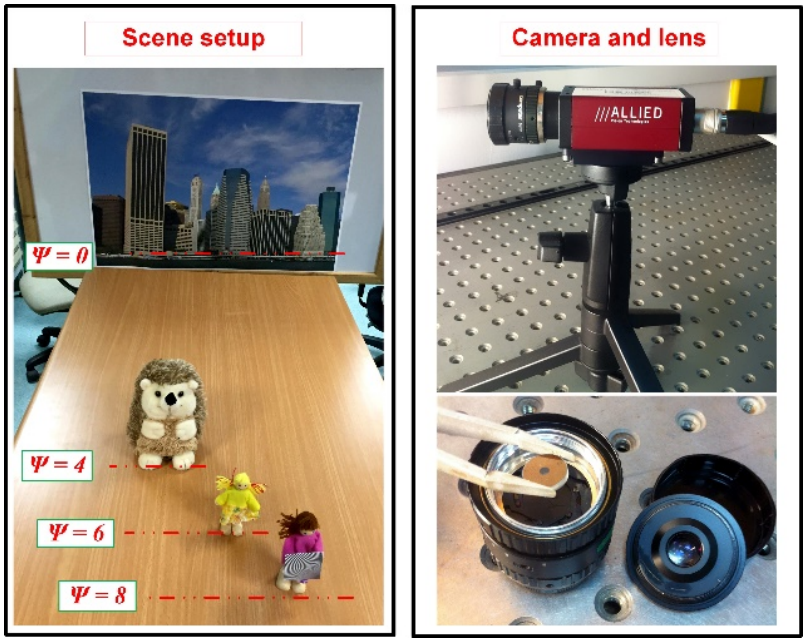


Fig. 37: Experimental set-up: Left - Scene line-up including the relative OOF factor for each plane. Right - View of camera and mask insertion in the lens assembly.

The computational run time was about 2 minutes for a 1.3MP image using a MATLAB implementation running on an Intel i7-2620M laptop with 8GB of RAM. In [156], a fixed-complexity alternative to iterative pursuit methods was presented. It achieved real-time performance on various image processing applications and inverse problems. An FPGA

prototype applying a similar methodology to our present imaging system is presented in the next section.

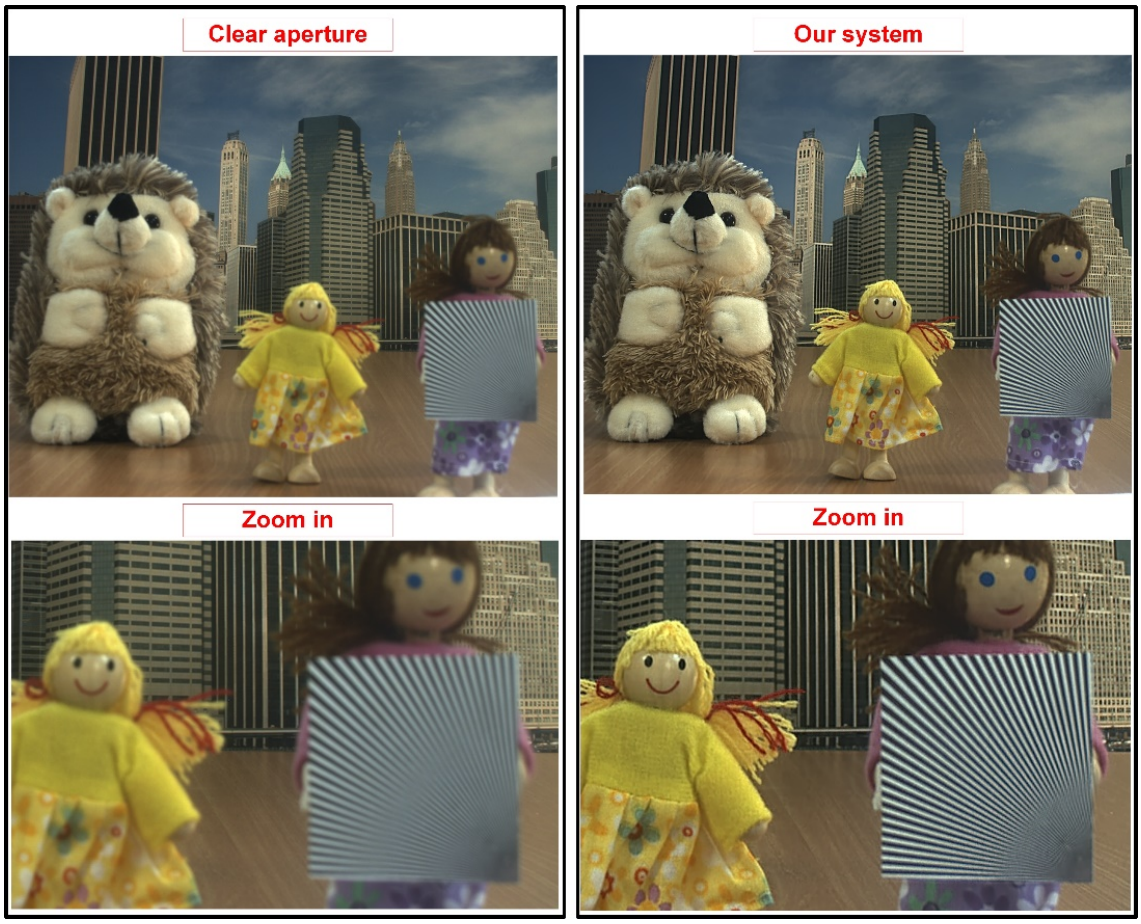


Fig. 38: Experimental results – Imaging with a conventional clear aperture (left) and imaging with a phase mask and our post processing restoration (right).

Increasing camera resolution while reducing pixel size poses several challenges concerning noise and reduced DOF. In the second version of our imaging system a CMOS sensor with a 1.67μm pixel pitch was used. A pixel pitch about three times smaller than our initial prototype allowed increasing the resolution from 1.4MP to 10.5MP. The mask thickness was also reduced to 130μm allowing a simpler assembly inside a 16mm lens (LM16JCM-V by Kowa). The lens was chosen due to its unique two-parts design, allowing the mask insertion without the need to disassemble the lens (Fig. 39). We also used an optimized two-rings mask, designed by the optimization process detailed in Section 3.3.

As presented in Fig. 40, using our system (central column), we achieved larger DOF than that obtained with a conventional camera with the same aperture size (left column). When comparing to a conventional camera with a higher F# (right column) our system still

achieved a higher DOF while reducing the noise to a minimum. Both demosaicing as well as noise reduction were performed alongside the DOF restoration process.

Fig. 39: The 130 µm mask (left) and the 'Kowa LM16JCM-V' two-part 16mm lens.

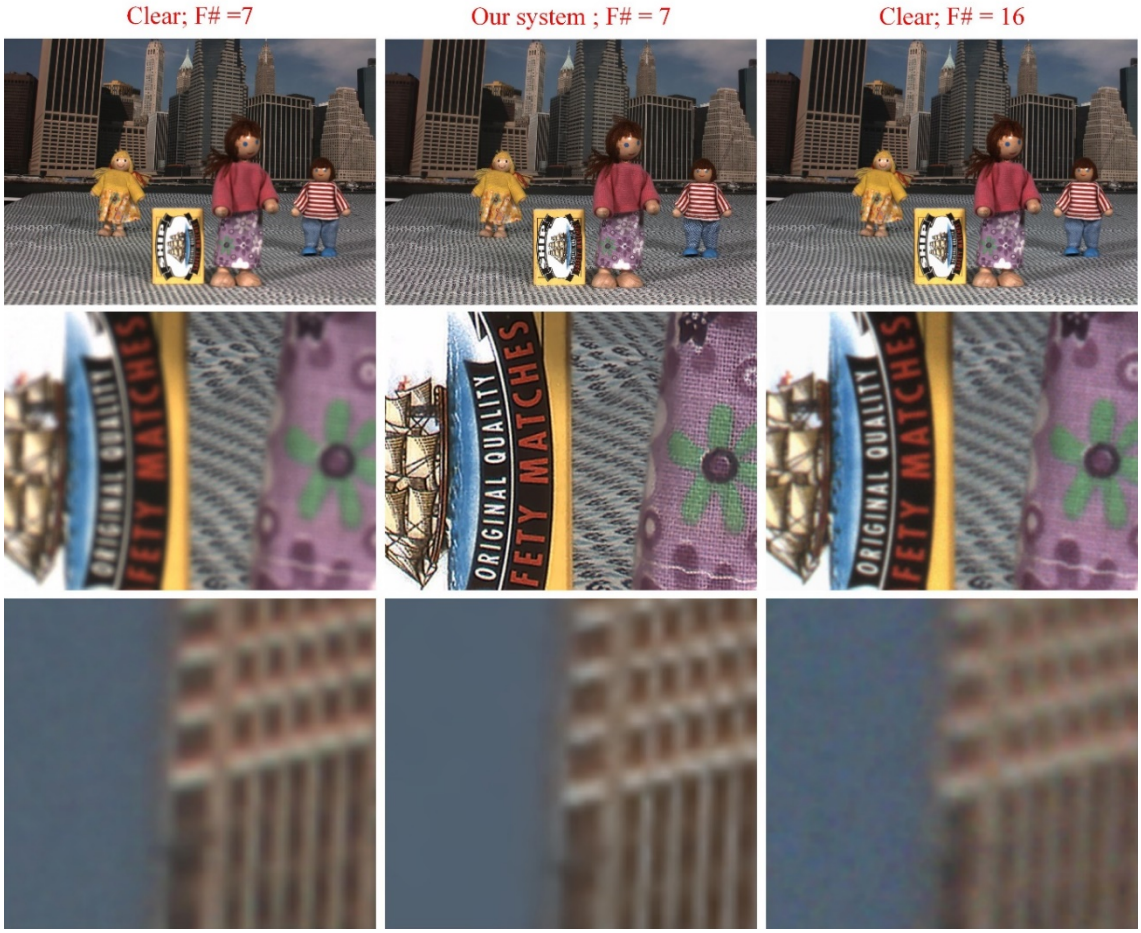


Fig. 40: DOF vs. Noise – left to right: Clear aperture with F#=7; Our system with F#=7; Clear aperture with F#=16; notice the sharpness and noise reduction in the center column.

4.5 FPGA implementation for real-time EDOF system

In the previous section, a method for EDOF imaging was presented. Using a phase mask, the captured image exhibits a unique color response which was then use for blind restoration using post-processing. The computational stage was based on a non-blind method that handle several blurring kernels within a single image, while the purely computational methods for blind image restoration handles only one blurring kernel at a time. Even though our restoration process offered a competitive restoration time to other methods, it has not reached its full potential as a real-time process.

In this section, a proof-of-concept end-to-end system for fast computational EDOF imaging. This system is based on a fast, non-iterative reconstruction algorithm, operating with constant latency in fixed-point arithmetic's and achieving real-time performance in a prototype FPGA implementation. The output of the system, on simulated and real-life scenes, is qualitatively and quantitatively better than the result of clear-aperture imaging followed by state-of-the-art blind deblurring.

4.5.1 Fast image reconstruction

As was explained in Section 2.4.3, the pursuit problem (Eq. (61)) can be posed as a convex optimization problem by choosing an ℓ_1 regularization term which controlled by the parameter μ, resulting the following optimization problem:

$$\mathbf{z}_i = \arg\min_{\mathbf{z}_i} \left\| \mathbf{y}_i - \mathbf{B}\mathbf{D}_\Psi \mathbf{z}_i \right\|_2^2 + \mu \left\| \mathbf{z}_i \right\|_1 \tag{62}$$

This problem can be solved using proximal algorithms such as the iterative shrinkage thresholding algorithm (ISTA) or its accelerated version (FISTA) [112] as detailed in Section 2.4.3. However, these iterative solvers typically require hundreds of iterations to converge, resulting in prohibitive complexity and unpredictable input-dependent latency, which is unacceptable in real-time applications.

To overcome this limitation, we follow the approach advocated by [113], in which a small number, *T*, of ISTA iterations is unrolled into a feed-forward neural network that subsequently undergoes supervised training on typical inputs, as explained in the sequel. A pseudo-code of ISTA is given in Fig. 41, where $\mathbf{Q} = \frac{1}{L}\mathbf{B}\mathbf{D}_\Psi$, $\mathbf{S} = \mathbf{I} - \frac{1}{L}\mathbf{Q}^T\mathbf{Q}$, and $\sigma_\theta(x) = \max(|x| - \theta, 0)\text{sign}(x)$ is a two-sided shrinkage function with the threshold $\theta = \frac{2\mu}{L}$ applied element-wise. *L* denotes a scalar larger than the largest eigenvalue of $\mathbf{Q}^T\mathbf{Q}$.

Inputs:

- signal $\mathbf{x}$, shrinkage threshold θ and matrices $\mathbf{S}, \mathbf{Q}$
- number of iterations T

Output:

- Approximation vector $\mathbf{z}$ such as $\mathbf{x} \approx \mathbf{D}\mathbf{z}$

Initialization:

- Initial solution $z_1 = 0$; and $b_1 = Q^T x$

Main iteration loop: *for* $t = 1, 2, ..., T$

- *for* $t = 1, 2, ..., T$
  - $\mathbf{z}_{t+1} = \sigma_\theta(\mathbf{b}_t)$
  - $\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{S}(\mathbf{z}_{t+1} - \mathbf{z}_t)$
- End while
- Set the solution to: $\mathbf{z} = \mathbf{z}_T$

Fig. 41: ISTA algorithm for blind image restoration

As can be easily interpreted from the ISTA algorithm, the network comprises three types of layers: An initialization layer (denoted as I), which finds the representation of the input signal in the dictionary; several $(T-2)$ recurrent middle layers (M) performing the gradient step followed by element-wise shrinkage; and a final layer (F) which translates the resulting dictionary coefficients to the reconstructed signal. All these types of layers can be realized from the single multi-purpose calculator stage shown in Fig. 42 (right) that is amenable for hardware implementation. To get the "I" configuration, we set $\mathbf{b}_{in} = 0$, $\mathbf{A} = -\mathbf{Q}^T$, $c = 0$ and $\theta = 0$. The ``M" configuration layer is fed by the output of the previous calculator, and the matrix $\mathbf{A}$ is set to $\mathbf{S}$. The output is further fed to either another ``M" layer or to the ``F" layer. Finally, the ``F" configuration of the calculator is a reduction into multiplication by the matrix $\mathbf{D}$. This is achieved by setting $\mathbf{b}_{in} = 0$, and $\mathbf{A} = \mathbf{D}$.

Supervised training of the network is done by initializing the parameters as detailed above, and then adapting them using a stochastic gradient procedure minimizing the reconstruction error $\mathcal{F}$ of the entire network. We use the following empirical loss:

$$\mathcal{F} = \frac{1}{N} \sum_{n=1}^{N} f(x_n^*, \hat{x}_n) \tag{63}$$

which for a large enough training set, N, approximates the expected value of f with respect to the distribution of the ground truth signals x_n^*. Here, $\hat{x}_n$ denotes the output of the network, and the loss objective f minimized during the training process is the standard sum of squared differences,

$$f = \frac{1}{2} \| \; x_n^* - \hat{x}_n \|_2^2 \; . \tag{64}$$

Similarly to [113], the output of the network and the derivatives of the loss with respect to the network parameters are calculated using the standard forward and back propagation approach. Practice shows that the training process allows to reduce the number of layers by about two orders of magnitude while achieving a comparable reconstruction quality.

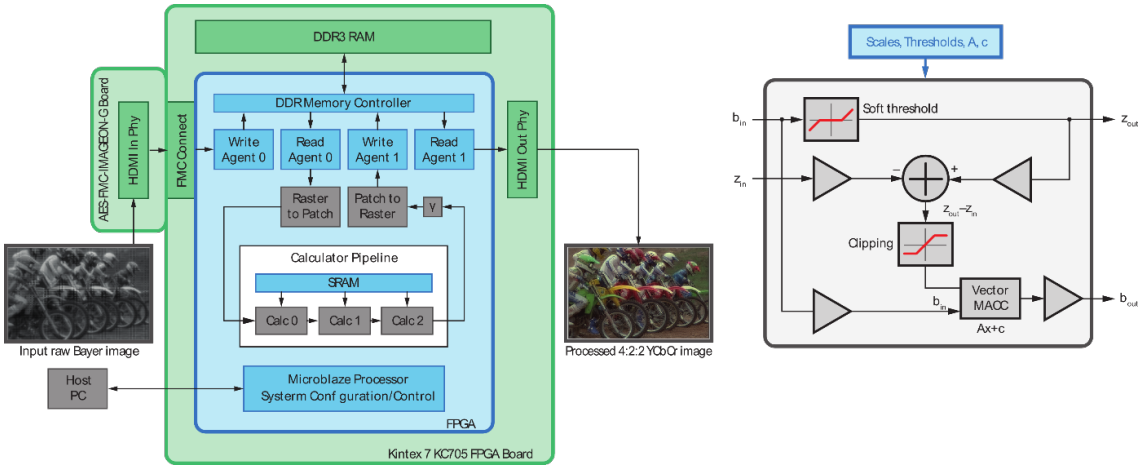


Fig. 42: ***Schematic description of the FPGA reconstruction system****. The raw Bayer image from the sensor at 12bit/pixel is passed, through the HDMI input interface daughter board, to the Kintex 7 FPGA chip. The image is buffered in the external DRAM, from where it is fed as a stream of possibly overlapping 8x8 patches to the calculator pipeline comprising of up to eight stages (see detail on the right), implementing the neural network architecture. The output patches in 4:2:2 YCbCr format are average-pooled and buffered in raster order in the DRAM, from where the image is sent over to the HDMI output interface on the FPGA board. The parameters of the calculator stages and other register values controlling the data flow are stored in the static memory on the chip, into which they are loaded by the host application on system startup.*

4.5.2 FPGA image reconstruction system

To demonstrate that the proposed image reconstruction process is efficient and is amenable to hardware implementation, we built a prototype FPGA system. An FPGA is a programmable chip containing configurable logic blocks and routing resources, therefore offering a fair amount of flexibility previously only possible with software, along with a hardware-like computational speeds and reliability. While being distinct in many aspects from application-specific integrated chips (ASICs), modern FPGAs are the closest approximation of an ASIC one can get without incurring the costs of custom chip manufacturing.

A schematic description of our system is depicted in Fig. 42 (left). We used the Xilinx Kintex 7 chip on the KC705 development board chosen mainly because of the availability

of video interfaces. As the output, we used the onboard HDMI output phy, while for the input, we added an external HDMI phy board connected to the main board through an FCM connector. The input frames are received by the board through the HDMI interface in raw Bayer format, 16 bits per pixel with the most significant bytes packaged as the Y channel, and the least significant byte packaged as the 4:2:2 color channels. The input is relayed to Write Agent 0 on the FPGA chip that buffers it in the external dynamic memory. The content of the buffer is brought into the chip by Read Agent 0, which reorders the raster scan order into a stream of 8×8 patches with configurable amount of overlap. The patches are then fed into a configurable calculator pipeline implementing the reconstruction algorithm detailed in the previous section. The pipeline comprises three configurable stages, one of which is configured as the initial stage (I), another as a middle stage (M), and yet another as the final stage (F), yielding the flow structure of the form $I \to (T-2)\times M \to F$.

The output of the calculator pipeline is produced in 4:2:2 YCbCr format comprising 64 luma values at 16 bits per pixel, and additional $32+32=64$ chroma values at 8 bits per pixel. The luma component undergoes gamma conversion implemented as a lookup table, reducing it to 8 bits per pixel. The patches are average-pooled (in case of overlap), reordered into raster scan order, and buffered into the dynamic memory by Write Agent 1. Finally, Read Agent 1 conveys the content of the output buffer to HDMI output.

A schematic block diagram of a calculator stage is depicted in Figure Fig. 42(right). Calculations are performed on vectors in fixed point arithmetic's with 16 precision bits except the multiply-and-accumulate (MACC) block that uses 48-bit arithmetic's internally. To keep a reasonable dynamic range, the data are scaled between various operations by scale factors that were carefully selected to minimize precision loss on a large set of patches from a collection of natural images. Compared to its floating-point counterpart, the fixed-point implementation produced negligible quality degradation in all our experiments.

Each calculator performs element-wise soft thresholding and the multiplication of the input data by a matrix of size 64×192 (initial stage, converting the input 64-dimensional Bayer patch into a set of 192 coefficients), 192×192 (middle stage, performing operators on the coefficients), or 192×128 (final stage, converting the coefficients into a 4:2:2 YCbCr patch with 64 luma dimensions and additional 64 chroma dimensions). This is implemented by using MACC blocks of respective sizes. The parameters of each calculator stage, including threshold values and matrix coefficients, are stored in a local static memory on the FPGA chip.

Since MACC operations are fully pipelined, they require one clock cycle. The total number of clock cycles it takes a single patch to pass though the chain is given by $64+192\times(T-2)$, where $(T-2)$ denotes the number of middle stages. There are additional overheads of approximately 100 cycles per network layer. Due to high resource utilization, we were able to use clock frequency of 125MHz only. This results in overall

throughput of about 16, 1920×1080 frames per second without patch overlap and a 4-layer network.

4.5.3 Results

In this experiment, we evaluated the performance of our algorithm on synthetic data from the KODAK dataset [154]. Each image was convolved with the same PSF corresponding to $\Psi = 8$ and mosaiced to simulate the input to the system. The reconstruction neural network was trained using 2×10^6 patches taken from the KODAK training set. The networks with $T = 8$ layers and $T = 4$ were converted to fixed-point arithmetic's as described in the previous section.

The algorithm was compared to the OMP with a k=192 atom dictionary as described in Section 4.3. As a reference, we compared our algorithms to the blind deblurring algorithm from [151] following MATLAB default demosaicing algorithm.

Image reconstruction quality in terms of average PSNR and SSIM [157], and execution times are presented in Table 2. It is evident that the highest PSNR is achieved by the FPGA implementation with $T = 8$, while restricting to $T = 4$ layers performed twice as fast yielding almost the same average PSNR and SSIM scores. Interestingly, the neural network achieves better reconstruction quality compared to the greedy OMP algorithm, which we attribute to the effect of supervised training. Both sparse prior-based algorithms outperform the blind deconvolution algorithm [151] by over 1 dB PSNR and about 17% higher SSIM. Comparing the execution times of the algorithms on a standard CPU shows that the OMP algorithm is about 30 times faster than [151], while the 4-layrs FPGA implementations was about 90 times faster than OMP about 2700 times faster than [151] with superior reconstruction quality.

As indicated in the previous section, for natural depth scenes one cannot assume that the image is blurred by a single blurring kernel. Our reconstruction process analyzes small patches rather than the entire image; therefore, the process is applied to every region inside the image independently of the other regions, allowing our algorithm to treat the input as if it had a single blur kernel.

To demonstrate the process, we used the same scene, presented in Fig. 38 (Section 4.3.2). A similar restoration process to the one described in Section 4.4 was carried except this time a patch stride of 2 pixels was set (instead of stride of 1 pixels – full overlap). A comparison of the OMP and FPGA results is presented in Fig. 43 top and bottom raw respectively. Two zoom-in images are also presented to the right of each output image. The 2-stride setting reduced the running time by 4 compared to the 1-stride process but it also introduces some artifacts the are noticeable in the OMP output. The FPGA however, provided sharp demosaicing and deblurring results without noticeable artifacts. As mentioned earlier, the FPGA performed much faster than the OMP process which make it ideal for real-time applications.

	PSNR [dB]	SSIM	Time [sec]
Raw input	22.63	0.59	-
Blind deblurring [151]	24.57	0.68	501.17
OMP	25.8	0.80	17.09
FPGA (T=4)	25.86	0.81	0.19
FPGA (T=8)	26.16	0.82	0.36

Table 2: Comparison of average PSNR, SSIM and run time on KODAK images [154]. The values presented in the table are the averages over all test images in the KODAK dataset after they have been blurred using $\Psi = 8$ and reconstructed with the different algorithms. All patch based algorithms were run using a patch stride of 2 pixels. Executing times were measured on an Intel Xeon CPU. Our fixed-point implementation was executed on a 100MHz Xilinx Kintex 7 FPGA.

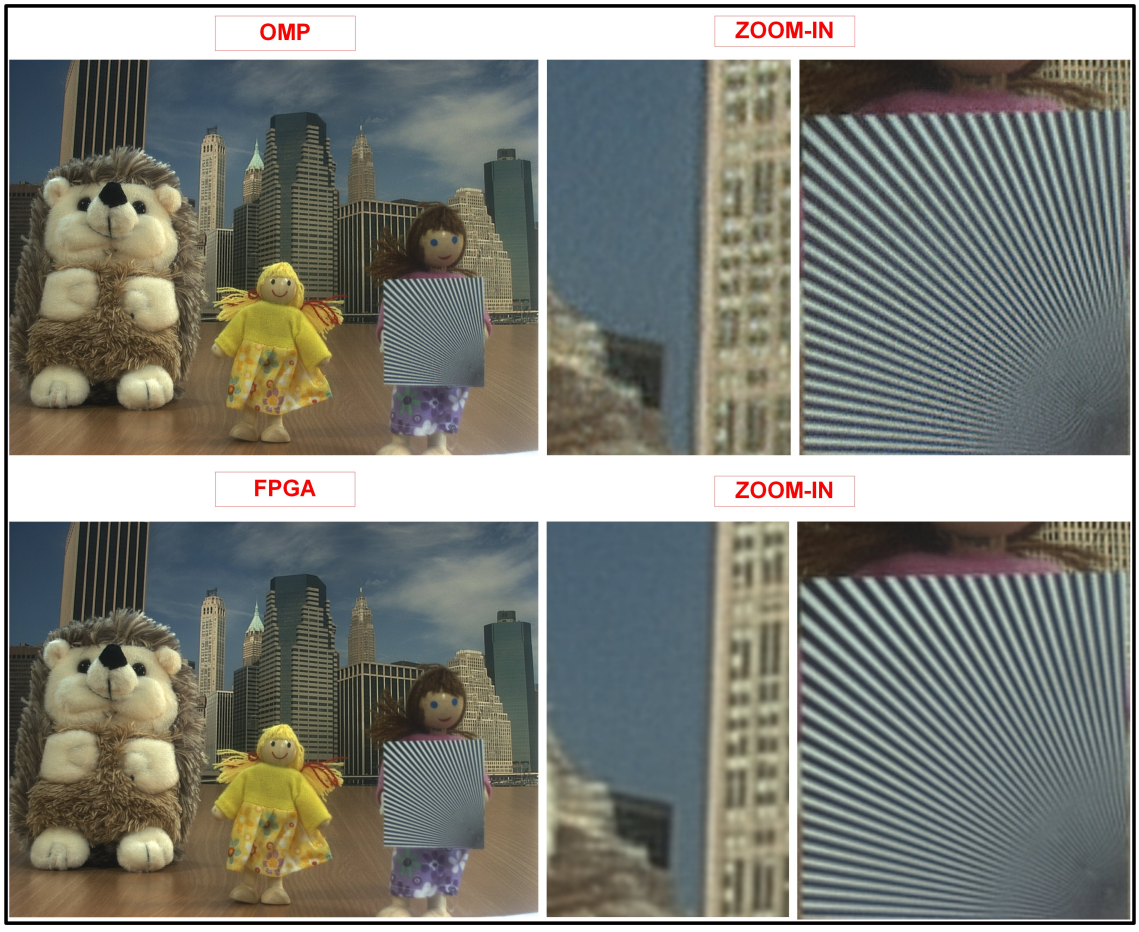


Fig. 43: Comparison between OMP (top row) and the FPGA implementation (bottom row). The right most columns show magnified fragments.

4.6 Depth estimation and image refocusing

4.6.1 Scoring model for Ψ labeling map

The blind restoration process described in the previous section implies an application for depth map estimation associated with the most responsive sub-dictionary in each patch. This application is limited by the system ability to find the correct sub-dictionary for each patch. While minor errors in the dictionary selection process, such as selecting elements from adjacent dictionary, will have almost no effect on the deblurring process, it will however affect the depth estimation process.

An obvious extension for this model will be adding a group sparsity prior [150], [158], [159] using a mix of ℓ_2 norm to encourage sparse coefficients from the same group (i.e. same sub-dictionary) to be zero or nonzero simultaneously. Assuming a perfect group sparsity model is implemented on our system, the depth resolution will still be limited to the number of groups (sub-dictionaries) in our model. Increasing the number of groups will reduce the separation performance of the group model while demanding more computational efforts.

To overcome the above limitation, we suggest an optimized phase mask (Section 3.3), and a new scoring model based on the OMP process to estimate a continuous depth map using a few sub-dictionaries in the process.

Using a single blurred sub-dictionary $\mathbf{D}_{\Psi m}$ $(m = 1, 2, \ldots, k)$ to represent a blurred patch $\mathbf{y}_i$ the sparse coefficient vector z_{im} is calculated using a modified version of (61) as

$$\mathbf{z}_{im} = \arg\min_{\mathbf{z}_i} \|\mathbf{z}_i\|_0 \qquad \text{s.t. } \left\|\mathbf{y}_i - \mathbf{B}\mathbf{D}_{\Psi_m}\mathbf{z}_i\right\|_2^2 < \varepsilon . \tag{65}$$

In other words, for each patch we calculate the sparsest vector $\mathbf{z}_{im}$ for each sub-dictionary separately such that the data term will be smaller than ε. In practice $\|\mathbf{z}_i\|_0$ is also limited to *32* nonzero elements. Following the OMP process, each patch will be given a performance score for each sub-dictionary, comprising both the sparsity and the reconstruction error

$$\mathbf{s}_{im} = \|\mathbf{z}_{im}\|_0 \cdot \left\|\mathbf{y}_i - \mathbf{B}\mathbf{D}_{\Psi_m}\mathbf{z}_{im}\right\|_2^2 , \tag{66}$$

where $\mathbf{s}_i \in \mathbb{R}^k$ is the scoring vector for each *i*-th patch. The first term $\|\mathbf{z}_{im}\|_0$ encourage sparsity while the second one represents the data fitting error.

Next, using an average scoring of overlap patches we construct a three-dimensional normalized scoring map $\mathbf{S} \in \mathbb{R}^{r \times c \times k}$, where r and c are, respectively, the row and column dimensions of the input raw image ***y***. Notice that the better the fit of a sub-dictionary to the data the lower the score $\mathbf{s}_{im}$, and vice versa. A Ψ label map can be estimated by

setting the Ψ label for each pixel as the minimum value over the 3'rd dimension of the $\mathbf{S}$ matrix. However, the distinction between low and high scoring of a patch will be significant in rich texture areas but, for low texture areas (or smooth areas with noise), scoring will be low for all sub-dictionaries and thus, the label map will be noisy.

To address this issue, we compute a Confidence Map ($\mathbf{C} \in \mathbb{R}^{r \times c \times k}$), such that high value relates not only to the fit scoring but also to our confidence level of choosing one sub-dictionary over the others:

$$\mathbf{C}(i,j,q) = Max\left\{0, Mean^{(3)}\left[\mathbf{S}(i,j,\bullet)\right] - \mathbf{S}(i,j,q)\right\} \tag{67}$$

where $i = 1,...,r$, $j = 1,...,c$ and $q = 1,...,k$. The term $Mean^{(3)}\left[\mathbf{S}(i,j,\bullet)\right]$ returns the averaging value of the (i,j) image location over the 3'rd dimension of the $\mathbf{S}$ matrix. For rich texture patches, the $\mathbf{S}$ matrix mean value will be high, due to high scoring of the unfitted sub-dictionaries, which will set the confidence level of the sub-layers with higher than average scoring to zero, while setting high confidence level in the low scoring layers (which related to high fit level). For low texture patches, the $\mathbf{S}$ matrix mean value will be very low in all layers, thus setting the confidence level of those area to a minimum.

Each layer in the $\mathbf{C}$ is filtered (using Gaussian kernel) to reduce data noise. After a final filtering stage using morphological operations, the discrete Ψ'*s* labels map was set as the max value over the 3'rd dimension of the $\mathbf{C}$ followed by morphological filtering stage to smooth the map and to fill the zero confidence area with their nearest neighbor label (Fig. 44 (b)). To produce a continuous map, one can use the $\mathbf{C}$ to perform a weighted averaging between the different layers to estimate the sub-Ψ's labeling as shown in Fig. 44(c). The image in Fig. 44(a) was captured using our two-rings phase mask design (as described in Section 3.3).

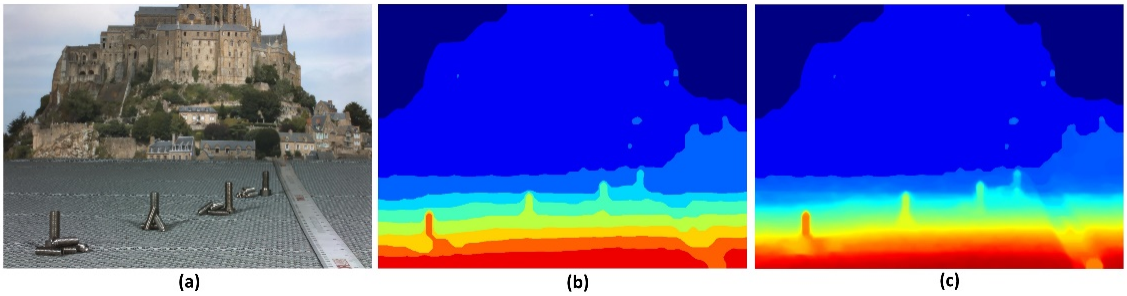


Fig. 44: Depth map. (a) input image captured with our two-rings phase mask design; (b) Ψ's map segmentation; (c) continuous Ψ's map.

The scoring method can be set as an integral part of the EDOF system, replacing the previous method of using all sub-dictionaries at once with the single dictionary step described in this section. This sparse presentation of this new method is more accurate

and thus, the restoration process produces better results. The overhead run time is about 20% as the iterative process required *k*-times steps but each is calculated using a *k*-times smaller dictionary.

4.6.2 Depth estimation results

The new algorithm method which presented in the previous section produced Ψ 's labels map which can be easily transformed into a metrical depth map using Eq. (20):

$$z_o = \left(\frac{\lambda \Psi}{\pi R^2} + \frac{1}{z_n} \right)^{-1} \tag{68}$$

where z_n is the nominal object position (focus point) and z_o is the actual object position. To set the focus point we used a "clear phase mask", which was simply no-rings phase mask, to compensate for mask thickness. After the focus point was set to the designated distance, the clear mask was replaced with the RGB mask. An example of depth map segmentation is presented in Fig. 46. Pseudo-colors were used for identifying various distances, as clearly observed on the "floor" portion of the figure.

Fig. 45: Depth segmentation visualization of a scene captured with our system. For illustration purposes, the colored map is fused onto the actual image (see Fig. 40).

To test the system performance and depth accuracy we captured a scene with a background poster and a textured surface. The distances from the camera to a few points over the table were manually measured and a full map was calculated using the camera properties. We used our process to estimate the distance from the camera and compared the results with the actual, manually measured distance. The image shown in Fig. 46 shows the depth estimation obtained with our approach in comparison to the measured depth of the scene. For a discrete depth map, one can detect depth over a range of 60cm to 150cm with a 4cm error. The continuous map can reduce the estimation error by at least half.

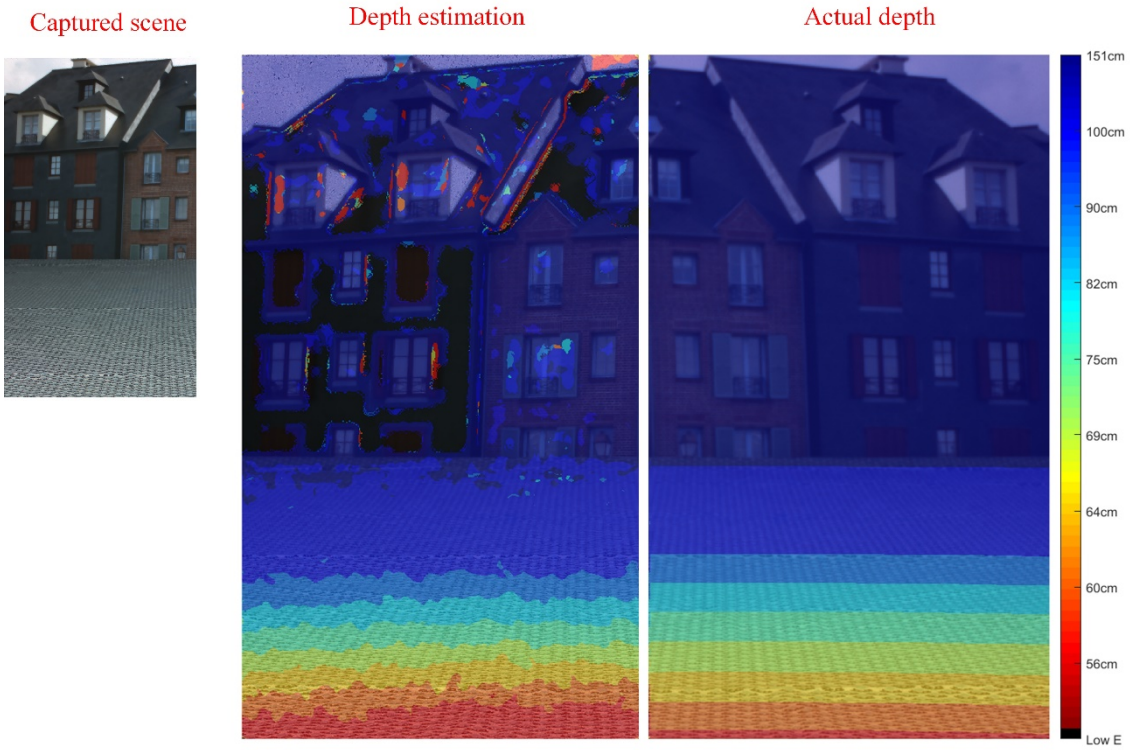


Fig. 46: Depth estimation comparison to an actual depth measurement results.

4.6.3 Image refocusing

Image refocusing became one of the most interesting application today in the competitive world of smartphones cameras. Our system provides this ability by utilizing both the al-in-focus image output and depth map estimation. Changing the focus point is done by setting the designated distance as the new focus point and gradually blurring adjacent depth segments while increasing the blurring effect we move away from the focus point. To create a DLSR-like effect with a shallow DOF we can increase the blurring effect by choosing gaussian kernel with higher variance. This effect is demonstrated in Fig. 47 as the focus point changed from the foreground Fig. 47(a), mid-field Fig. 47(b) and background Fig. 47(c).

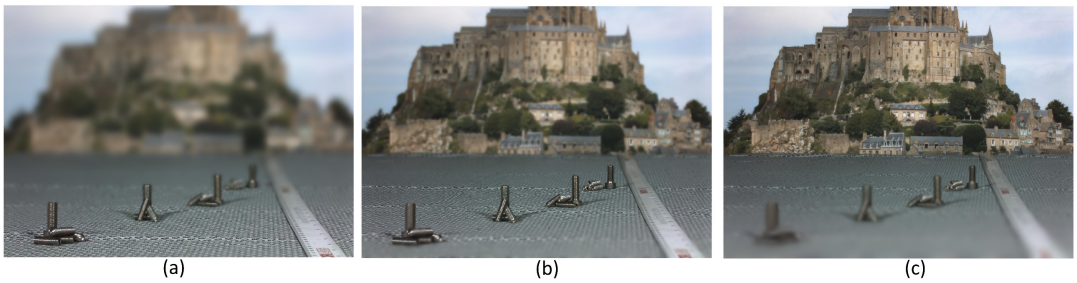


Fig. 47: Imaged refocusing: using both depth map and all-in-focus image one can produce a DSLR like image with shallowed DOF. Focus point can be changed computationally to the foreground (a) center field (b) and background (c).

4.7 Chapter summary

In this section, we have presented the foundation for extended depth-of field system, based on a modified conventional optical imaging system equipped with a special thin binary phase mask, followed by an electronic post-processing stage. The phase-coded aperture computational EDOF imaging system aims at solving one of the biggest challenges in today's miniature digital cameras, namely, acquisition of images with both high spatial resolution and large depth of field in demanding lighting conditions. The processing is based on simple dictionary based image deblurring algorithm. Our method was tested successfully on real life natural depth scenes with no need of prior knowledge about the scene composition or user intervention. The experimental results provide added validity of our method.

Our proposed solution can be easily incorporated into existing imaging systems since it requires the addition of a thin mask (which can eventually be etched or fabricated onto one of the existing optical surfaces), and a simple real-time hardware computational unit, which was demonstrated in an FPGA prototype. As we demonstrated through extensive experiments, our system outperforms existing techniques for post processing an image taken by a standard camera.

In the last part of this chapter, a method for estimating continuous metrical depth map from a single image capture with our phase mask camera, was introduced. The scoring process used in this method produce accurate depth map which can also be used for improving the deblurring procedure. Integration between all-in-focus and depth map provide artistic, yet highly commercial, refocusing abilities which allow changing the focus point post facto as well as emulating a DSLR like shallow DOF.

The next chapter presents a neural network based method which not only improves the depth estimation accuracy, but also speeds up the process by three orders of magnitude.

5. Depth Estimation from a Single Image using Deep Learned Phase Coded Mask

Depth estimation from a single image is a well-known challenge in computer vision. With the advent of deep learning, several approaches for monocular depth estimation have been proposed, all of which have inherent limitations due to the scarce depth cues that exist in a single image. Moreover, these methods are very demanding computationally, which makes them inadequate for systems with limited processing power. In this chapter, a phase-coded aperture camera for depth estimation is proposed. The camera is equipped with an optical phase mask that provides unambiguous depth-related color characteristics for the captured image. These are used for estimating the scene depth map using a fully-convolutional neural network. The phase-coded aperture structure is learned together with the network weights using back-propagation. The strong depth cues (encoded in the image by the phase mask, designed together with the network weights) allow a much simpler neural network architecture for faster and more accurate depth estimation. Performance achieved on simulated images as well as on a real optical setup is superior to the state-of-the-art monocular depth estimation methods (both with respect to the depth accuracy and required processing power), and is competitive with more complex and expensive depth estimation methods such as light field cameras.

5.1 Introduction

While a single image lacks the depth cues that exist in a stereo image pair, there are still some depth cues, as detailed in Section 2.3.4, that enable depth estimation to some degree of accuracy. The ongoing deep learning revolution did not overlook this challenge, and some neural network-based approaches to monocular depth estimation exist in the literature [74], [75], [160]–[163].

Eigen et al. [74] introduced a deep neural network for depth estimation that relies on depth cues in the RGB image. They used a multi-scale architecture with coarse and fine depth estimation networks concatenated to achieve both dynamic range and resolution. Two later publications by Cao et al. [160] and Liu et al. [75] employed the novel fully-convolutional network (FCN) architecture (originally presented by Long et al. [142] for scene semantic segmentation) for monocular depth estimation. In [160] the authors used a residual network [164], and refined the results using a conditional random field (CRF) prior external to the network architecture. In [75] a simpler FCN model was proposed, but with the CRF operation integrated inside the network structure. This approach is further researched using deeper networks and more sophisticated architectures [161]–[163].

Common to all these approaches is the use of depth cues in the RGB image 'as-is', as well as having the training and testing on well-known public datasets such as the NYU depth [60], [61] and Make3D [73]. Since the availability of reliable depth cues in a regular RGB image is limited, these approaches require large architectures with significant regularization (Multiscale, ResNets, CRF) as well as separation of the models to indoor/outdoor scenes.

A modification of the image acquisition process itself allows using a simpler model, generic enough to encompass both indoor and outdoor scenes. To take advantage of optical cues as well, the PSF should be depth-dependent. Related methods use an amplitude coded mask [31], [165] or a color-dependent ring mask [46], [166] such that objects at different depths exhibit a distinctive spatial structure. The main drawback of these strategies is that the actual light efficiency is only 50% in [31], [165], 60% in [46] and 80% in [166], making them unsuitable for low light conditions. Moreover, those techniques (except [166]) are based on the same low DOF setup, having a focal length of 50mm, F#=1.8 lens (27.8mm aperture). Thus, they are also unsuitable for small-scale cameras since they are less sensitive to small changes in focus.

In this chapter, a novel deep learning framework for the joint design of a phase-coded aperture element and a corresponding FCN model for single-image depth estimation, is presented. A similar phase mask has been proposed in [35], [36] for extended DOF imaging (see Section 3); its major advantage is light efficiency above 95%. Our phase mask is designed to increase sensitivity to small focus changes, thus providing an accurate depth measurement for small-scale cameras (such as smartphone cameras).

The single ring RGB phase, presented in Section 3.1, demonstrated how color diversity can be utilized for EDOF imaging [36]. In Section 3.3, a multiple ring design was introduced to allow superior color diversity. In this chapter, the aperture coding mask was designed specifically for encoding strong depth cues with negligible light throughput loss. The coded image is fed to a FCN, designed to observe the color-coded depth cues in the image and thus, estimate the depth map. The phase mask structure is trained together with the FCN weights, allowing end-to-end system optimization. For training, we created the 'TAUAgent' dataset with pairs of high-resolution realistic animation images and their perfectly registered pixel-wise depth maps.

Since the depth cues in the coded image are much stronger than their counterparts in a clear aperture image, the proposed FCN is much simpler and smaller compared to other monocular depth estimation networks. The joint design and processing of the phase mask and the proposed FCN lead to an improved overall performance: better accuracy and faster run-time compared to the known monocular depth estimation methods are attained. Also, the achieved performance is competitive with more complex and higher cost depth estimation solutions such as light field cameras.

5.2 Outline

This chapter is organized as follows:

Section 5.3 presents the phase-coded aperture used for encoding depth cues in the image, and its optimization process. Section 5.4 describes the FCN architecture used for depth estimation and a demonstration of its performance on synthetic data. Experimental results on real images acquired using an optical setup with a manufactured optimal aperture coding mask are presented in Section 5.5. Our system is shown to exhibit superior performance in depth accuracy, reduced system complexity, as well as lower processing

power compared to competing methods. In Section 5.6, a 3D model reconstruction example is presented to illustrate the advantage of a metrical depth map system. Section 5.7 summarizes the chapter.

5.3 Mask design

In order to find the optimal phase mask parameters within a deep learning-based depth estimation, the imaging stage is modeled as the initial layer of a CNN model. The inputs to this coded aperture convolution layer are the all-in-focus image and its corresponding depth map. The coded aperture convolution layer parameters (or weights) are the radii and phase shifts ϕ_i of the mask's rings.

The layer forward model is composed of the coded aperture PSF calculation (for each depth in the relevant depth range) followed by imaging simulation using the clean input image and its corresponding depth map. The backward model uses the inputs from the next layer (backpropagated to the coded aperture convolutional layer) and the derivatives of the coded aperture PSF, h_Ψ (Eq. (12)), with respect to its weights, $\partial h_\Psi / \partial r_i$, $\partial h_\Psi / \partial \phi_i$, in order to calculate the gradient descent step on the phase mask parameters. One of the important hyper-parameters of such a layer is the depth range under consideration (in Ψ terms). The Ψ range setting, together with the lens parameters (focal length, F# and focus point) dictates the trade-off between the depth dynamic range and resolution. In this study, we set this range to $\Psi = [-4,10]$; its conversion to the metric depth range is presented in Eq. (68)

As mentioned above, the optimization of the phase mask parameters is done by integrating the coded aperture convolutional layer into the CNN model detailed in the sequel, followed by the end-to-end optimization of the entire model. To validate the coded aperture layer, we compared the case where the CNN (described in the following section) is trained end-to-end with the phase coded aperture layer to the case where the phase mask is held fixed to its initial value. The training of the phase mask improves the classification error by 5% to 10%.

For our setup, the optimization process yields a three rings mask such that the outer ring is deeper than the middle one as illustrated in Fig. 28 (Section 3.3). Such a design poses some fabrication challenges for the chemical etching process used at our facilities. To make the fabrication process simpler and more reliable, a two-ring limitation was set in the training process; this resulted in the normalized ring radii $\mathbf{r} = \{0.55, 0.8, 0.8, 1\}$ and phases $\phi = \{6.2, 12.3\}[rad]$. This optimized mask design was almost identical to the mask design using the method presented in Section 3.3. However, the data driven optimization, presented in this section, provides an important set of tools which can be utilized for optimizing other tasks in computer vision such as classification, denoising etc.

5.4 FCN for Depth Estimation

We now turn to describe the architecture of our fully convolutional network (FCN) for depth estimation, which relies on optical cues encoded in the image achieved via using the phase coded aperture incorporated in the lens as described in Section 4.4.2. These cues are used by our FCN model to estimate the scene depth. Our network configuration is inspired by the FCN structure introduced by Long et al. [142]. That work converts an ImageNet classification CNN to a semantic segmentation FCN by adding a deconvolution block to the ImageNet model, and then fine-tunes it for semantic segmentation (with several architecture variants for increased spatial resolution). For depth estimation using our phase coded aperture camera, a totally different 'inner net' should replace the ``ImageNet model''. The inner net should classify the different imaging conditions (i.e. Ψ values), and the deconvolution block will turn the initial pixel labeling into a full depth estimation map. We tested two different 'inner' network architectures: the first based on the DenseNet architecture [167], and the second based on a traditional feed-forward architecture. A full FCN based on both inner nets is presented, and the trade-off is discussed. The following sub-sections present the Ψ classification inner nets, and the FCN model based on them for depth estimation.

5.4.1 Ψ classification CNN

As described in Section 5.3, the phase coded aperture is designed along with the CNN such that it encodes depth-dependent cues in the image by manipulating the response between the RGB channels in each depth. Using these strong optical cues, the depth slices (i.e. Ψ values) can be classified using some CNN classification model.

For this task, we tested two different architectures; the first one based on the DenseNet architecture for CIFAR-10, and the second based on the traditional feed-forward architecture of repeated blocks of convolutions, batch normalization [139] and rectified linear units [131] (CONV-BN-ReLU, see Fig. 48). In view of the approach presented in [138], pooling layers are omitted in the second architecture, and stride of size 2 is used in the CONV layers for lateral dimension reduction. This approach allows much faster model evaluation (only 25% of the calculation in each CONV layer), with minor loss in performance.

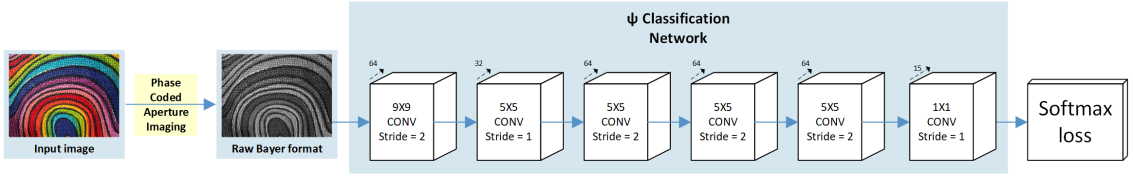


Fig. 48: Neural network architecture for the depth classification CNN (the 'inner' net in the FCN model in Fig. 49). Spatial dimension reduction is achieved by convolution stride instead of pooling layers. Every CONV block is followed by BN-ReLU layer (not shown in this figure).

To reduce the model size and speed up its evaluation even more, the input (in both architectures) to the first CONV layer of the net is the raw image (in mosaicked Bayer pattern). By setting the stride of the first CONV layer to 2, the filters response remains shift-invariant (since the Bayer pattern period is 2). This way the input size is decreased by a factor of 3, with minor loss in performance. This also omits the need for the demosaicing stage, allowing faster end-to-end performance (in cases where the RGB image is not needed as an output, and one is interested only in the depth map). One can see the direct processing of mosaicked images as a case where the CNN representation power 'contains' the demosaicing operation, and therefore it is not really needed as a preprocessing step.

Both inner classification net architectures are trained on the Describable Textures Dataset (DTD) [168]. About 40K texture patches (32x32 pixels each) are taken from the dataset. Each patch is 'replicated' in the dataset 15 times, where each replication corresponds to a different blur kernel (corresponding to the phase coded aperture in $\Psi = -4, -3, ..., 10$). The first layer of both architectures is the phase-coded aperture layer, whose inputs are the clean patch and its corresponding Ψ value. After the imaging stage is done, an Additive White Gaussian Noise (AWGN) with $\sigma = 3$ is added to each patch to make the network more robust to noise, which appear in images taken with a real-world camera. Data augmentation of four rotations is used to increase the dataset size and achieve rotation invariance. The dataset size is about 2.4M patches, where 80% of it is used for training and 20% is used for validation. both nets are trained to classify into 15 integer values of Ψ (between -4 and 10) using the softmax loss. These nets are used as an initialization for the depth estimation FCN, as presented in the following sub-section.

5.4.2 RGBD Dataset

The deep learning based methods for depth estimation from a single image mentioned in Section 5.1 [74], [75], [160]–[163] rely strongly on the input image details. Thus, most studies in this field assume an input image with a large DOF such that most of the acquired scene is in focus. This assumption is justified when the photos are taken by small aperture cameras as is the case in datasets such as NYU Depth [60], [61] and Make3D [73] that are commonly used for the training and testing of those depth estimation techniques. However, such optical configurations limit the resolution and increase the noise level, thus, they reduce the image quality. Moreover, the depth maps in those datasets are prone to errors due to depth sensor inaccuracies and calibrations issues (alignment and scaling) with the RGB sensor.

Our unique optical setup requires a dataset containing simulated phase coded aperture images and the corresponding depth maps. To simulate the imaging process properly, the input data should contain high resolution, all in-focus images with low noise, accompanied by accurate pixelwise depth maps. This kind of input may be generated almost only using 3D graphic simulation software. Thus, we use the popular MPI-Sintel depth images dataset [169], created by the Blender 3D graphics software. The Sintel

dataset contain 23 scenes with total of ~1k images. Yet, because it has been designed specifically for optical flow evaluation, the depth variation in each scene does not change significantly. Thus, we could only use about 100 unique images, which are not enough for training. The need for additional data has led us to create a new Sintel-like dataset (using Blender) called 'TAUAgent', which is based on the new open movie 'Agent 327'. This new animated dataset, which relies on the new render engine 'Cycles', contains 300 realistic images (indoor and outdoor), with resolution of 1024×512, and corresponding pixelwise depth maps. With rotations augmentation, our full dataset contains 840 scenes, where 70% are used for training and the rest for validation.

5.4.3 Depth estimation FCN

In similarity to the FCN model presented by Long et al. [142], the inner Ψ classification net is wrapped in a deconvolution framework, turning it to a FCN model (see Fig. 49). The desired output of our depth estimation FCN is a continuous depth estimation map. However, since training continuous models is prone to over-fitting and regression to the mean issues, we pursue this goal in two stages. In the first one, the FCN is trained for discrete depth estimation. On the second step, the discrete FCN model is used as an initialization for the continuous model training.

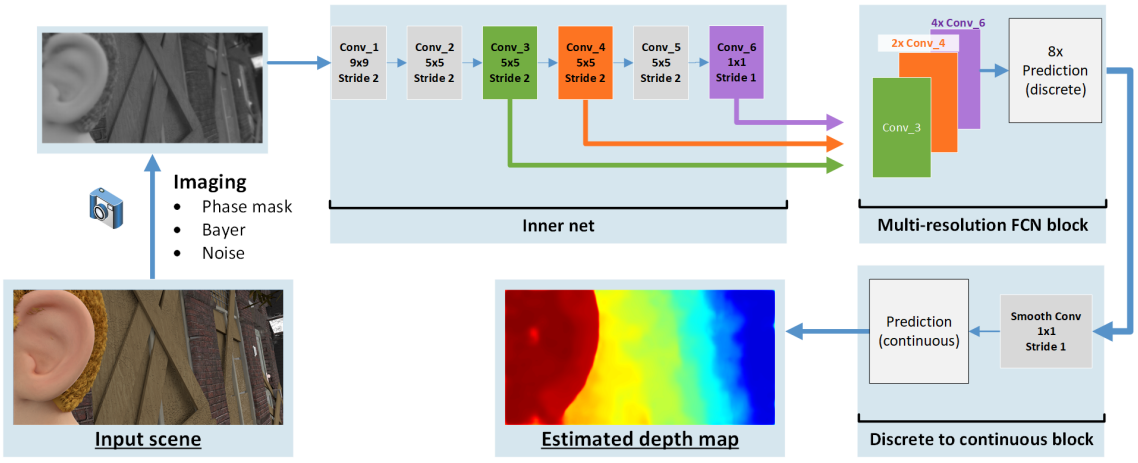


Fig. 49: Network architecture for the depth estimation FCN. The depth (Ψ) classification network (see Fig. 48) is wrapped in a deconvolution framework to provide depth estimation map equal to the input image size.

In order to train the discrete depth FCN, the Sintel and Agent datasets RGB images are blurred using the coded aperture imaging model, where each object is blurred using the relevant blur kernel according to its depth (indicated in the ground truth pixelwise depth map). The imaging is done in a quasi-continuous way, with Ψ step of 0.1 in the range of $\Psi=[-4,10]$. This imaging simulation can be done in the same way as the 'inner' net training, i.e. using the phase coded aperture layer as the first layer of the FCN model.

However, such step is very computationally demanding, and do not provide significant improvement (since the phase-coded aperture parameters tuning reached its optimum in the inner net training). Therefore, in the FCN training stage, the optical imaging simulation is done as a pre-processing step with the best phase mask achieved in the inner net training stage. In the discrete training step of the FCN, the ground-truth depth maps are discretized to $\Psi = -4,-3,...,10$ values. The Sintel/Agent images (after imaging simulation with the coded aperture blur kernels, RGB-to-Bayer transformation and AWGN addition), along with the discretized depth maps, are used as the input data for the discrete depth estimation FCN model training. The FCN is trained for reconstructing the discrete depth of the input image using softmax loss.

After training, both versions of the FCN model (based on the DenseNet architecture and the traditional feed-forward architecture) achieved roughly the same performance, but with a significant increase in inference time (x3), training time (x5) and memory requirements (x10) for the DenseNet model. When examining the performance, one can see that most of the errors are on smooth/low texture areas of the images, where our method (which relies on texture) is expected to be weaker. Yet, in areas with 'sufficient' texture, there are also encoded depth cues which enable good depth estimation even with relatively simple DNN architecture.

This similarity in performance between the DenseNet based model (which is one of the best CNN architectures known to date) to a simple feed-forward architecture is a clear example to the inherent power of optical image processing using coded aperture- a task driven design of the image acquisition stage can potentially save significant resources in the digital processing stage. Therefore, we decided to keep the simple feed-forward architecture as the chosen solution.

To evaluate the discrete depth estimation accuracy, we calculated a confusion matrix for our validation set (~250 images, see Fig. 50). After 1500 epochs, the net achieves accuracy of 68% (top-1 error). However, the clear majority of the errors are to adjacent Ψ values, and on 93% of the pixels the discrete depth estimation FCN recover the correct depth with a Ψ error of ± 1. As already mentioned above, most of the errors originate from smooth areas, where no texture exists and therefore no depth dependent color-cues were encoded. This performance is sufficient as an initialization point for the continuous depth estimation network.

The discrete depth estimation (segmentation) FCN model is upgraded to a continuous depth estimation (regression) model using some modifications. The linear prediction results serve as an input to a 1×1 CONV layer, initialized with linear regression coefficients from the Ψ predictions to a continuous Ψ values (Ψ values can be easily translated to depth value in meters, assuming known lens parameters and focus point).

The continuous network is fine-tuned in an end to end fashion, with lower learning rate (by a factor of 100) for the pre-trained discrete network layers. The same Sintel & Agent images are used as an input, but with the quasi-continuous depth maps (without

discretization) as ground truth, and L2 or L1 loss. After 200 epochs, the model converges to Mean Absolute Difference (MAD) of 0.6Ψ.

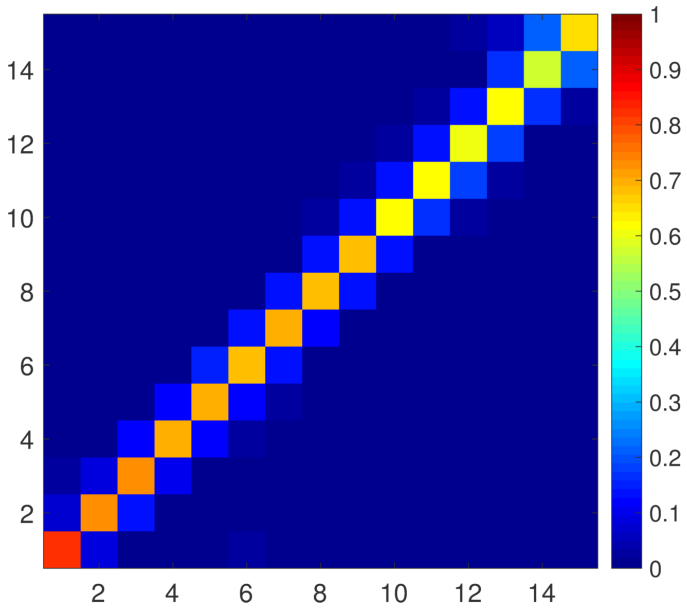


Fig. 50: Confusion matrix for the depth segmentation FCN validation set

.

5.4.4 Validation set results

As a basic sanity check, the validation set images can be inspected visually. As shown in Fig. 51, the depth cues encoded in the input image are hardly visible to the naked eye, however, the proposed FCN model achieves quite accurate depth estimation maps compared to the ground truth. Most of the errors are concentrated in smooth areas, as mentioned in previous section. The continuous depth estimation smooths the initial discrete depth recovery, achieving a more realistic result.

Notice the difference in the estimated maps when using the L1 loss (Fig. 51 (c)) and the L2 loss (Fig. 51 (d)). The L1 based model produces smoother output but reduces the ability to distinguish between fine details while the L2 model produces noisier output but provide sharper maps. This is illustrated in all scenes when the gap between the body and the hands of the characters is not visible as can be seen in Fig. 51 (c). Note that in this case the L2 model produces a sharper separation (Fig. 51 (d)).

The estimated maps in Fig. 51 (c-d) also presents a few limitations of our method. In the top row, the fence behind the bike wheel is not visible in our estimation since the fence wires are too thin. In the middle and bottom rows, the background details are not visible due to low dynamic range in these areas (the background is too far from the camera).

As mentioned above, our method estimates the blur kernel (Ψ value), using the optical cues encoded by the phase coded aperture. An important practical analysis is the translation of the Ψ estimation map to metric depth map. Using the lens parameters and

the focus point, transforming from Ψ to depth is straight-forward (see Eq. (68)). Using this transformation, the relative depth error can be analyzed. By setting a focus point, the $\Psi = [-4,10]$ domain is spread to some depth dynamic range. Close focus point dictates small dynamic range and high depth resolution, and vice versa. However, since the FCN model is designed for Ψ estimation, the model (and its Ψ's related MAD) remains the same. After translating to metric maps, the Mean Absolute Percentage Error (MAPE) is different for each focus point. Such analysis is presented in Fig. 52, where the aperture diameter is set to 2.3[mm] and the focus point changes from 0.1[m] to 2[m], resulting with a working distance of 9[cm] to 30[m]. One can see that the relative error is roughly linear with the focus point, and remains under 10% for relatively wide-focus point range.

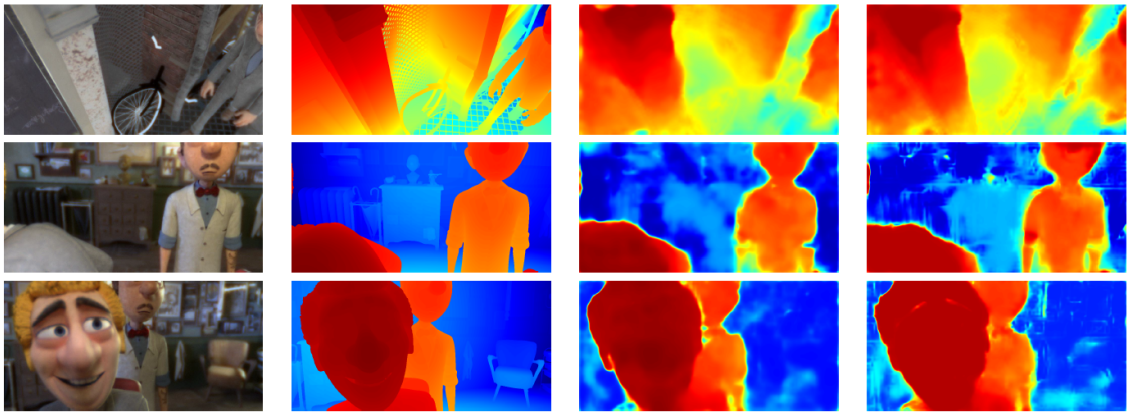


Fig. 51: Depth estimation results on simulated image from the 'Agent' dataset - (a) original input image (the actual input image used in our net was the raw version of the presented image), (b) Continuous ground truth (c-d) Continuous depth estimation achieved using the L1 loss (c) and the L2 loss (d).

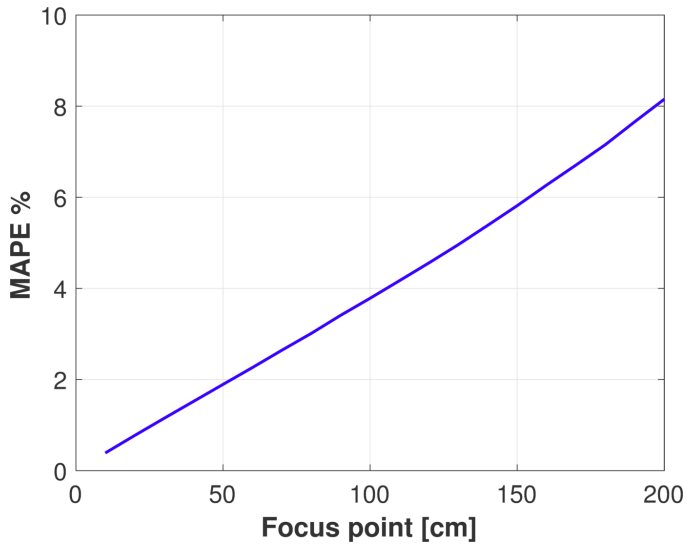


Fig. 52: MAPE as a function of the focus point using our continuous network.

5.5 Experimental results and comparison

To test the proposed depth estimation method, several experiments were carried. The experimental setup included an f=16mm, F#=7 lens (LM16JCM-V by Kowa) with our phase coded aperture incorporated in the aperture stop plane (see Section 4.4.2). The lens was mounted on a UI3590LE camera made by IDS Imaging. The lens was focused to $z_n = 110[cm]$, thus, the $\Psi = [-4,10]$ domain was spread between $0.5 - 2.2[m]$. Several scenes were captured using the phase coded aperture camera, and the corresponding depth maps were calculated using the proposed FCN model.

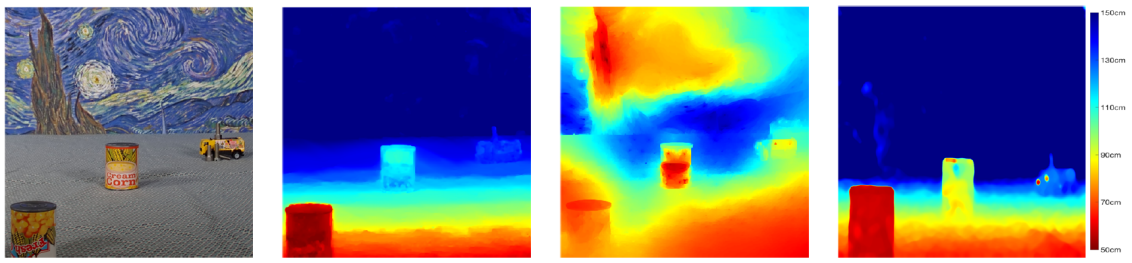
For comparison, two competing solutions were examined on the same scenes: Illum light field camera (by Lytro), and the monocular depth estimation net proposed by Liu et al. [75]. Since the method in [75] assumes an all in-focus image as an input, we used the Lytro camera all in-focus imaging option as the input to [75].

It is important to note that while our proposed method provide depth maps in absolute values (meters), the Lytro camera and [75] provide a relative depth map only (far/near values with respect to the scene). Another advantage of our technique is that it requires the incorporation of a very simple optical element to an existing lens, while light-field and other solutions like stereo require a much more complicated optical setup. In the stereo camera, two calibrated cameras are mounted on a rigid base with some distance between them. In the light field camera, special light field optics and detector are used. In both cases the complicated optical setup dictates large volume and high cost.

Fig. 53: Indoor scene (side view).

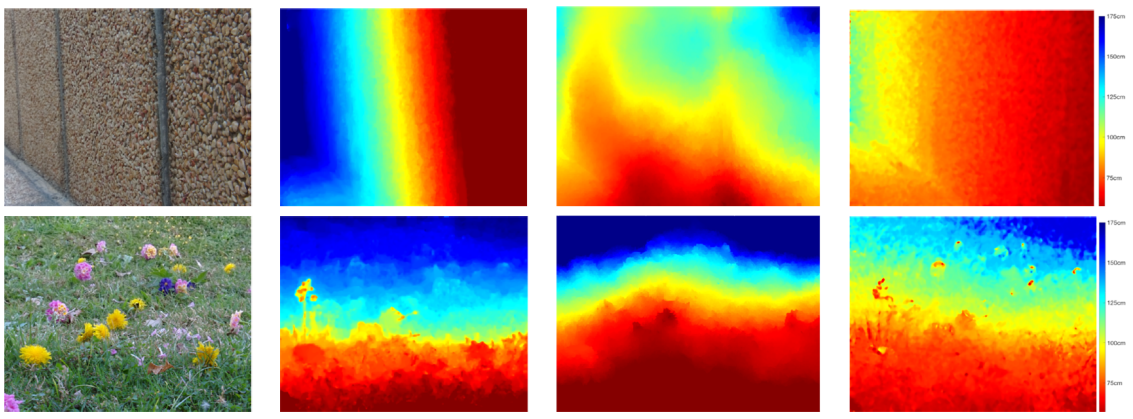
We examined all the solutions on both indoor and outdoor scenes. The indoor setup included objects that were laid on a table with a poster in the background (see Fig. 53 for a side view of the scene). The indoor example is presented in Fig. 54. Since the scene lacks global depth cues (especially in the background poster), the method from [75] fails to estimate a correct depth map. The Lytro camera estimates the gradual depth structure

of the scene with good identification of the objects, but in a relative scale only. Our method succeeds to identify both the gradual depth of the table and the fine details of the objects (note the screw located above the truck on the right). Note that some scene texture 'seeps' to our recovered depth map. Yet, it causes only a minor error in the depth estimate.



(a) scene (b) Light-Field (Lytro) (c) monocular (d) ours

Fig. 54: Indoor scene depth estimation. Left to right: (a) the scene and its depth map acquired using (b) Lytro Illum camera, (c) Liu et al. [75] *monocular depth estimation net, (d) our method. As each camera has a different field of view, the images were cropped to achieve roughly the same part of the scene. The depth scale on the right is relevant only for (d). Because the outputs of (b)&(c) provide only a relative depth map (and not absolute as in the case of (d)), their maps were brought manually to the same scale for visualization purposes. More examples appear in the supplementary material.*



(a) scene (b) Light-Field (Lytro) (c) monocular (d) ours

Fig. 55: Outdoor scenes depth estimation}. Depth estimation results for a granulated wall (upper) and grassy slope with flowers (lower) scenes. From left to right: (a) the scene and its depth map acquired using (b) Lytro Illum camera, (c) Liu et al. [75] *monocular depth estimation net, (d) our method. See caption of Fig. 54 for more details.*

Similar comparison is presented for two outdoor scenes in Fig. 55. On its first row, we have a scene with a granulated wall. In this example, the global depth cues are also weak, and therefore the monocular depth estimation fails to separate the close vicinity of the wall (right part of the image). Both the Lytro and our phase coded aperture camera

achieve good depth estimation of the scene. Note though that our camera has the advantage that it achieves an absolute scale and uses much simpler optics.

On the second row of Fig. 55, we have a grassy slope with flowers. In this case, the global depth cues are stronger. Thus, the monocular method [75] does better compared to the previous examples, but still achieves only a partial depth estimate. Lytro and our camera achieve good results.

Besides the depth map recovery performance and the simpler hardware, another important benefit of our proposed solution is the required processing power/run time. The fact that depth cues are encoded by the phase mask enables much simpler FCN architecture, and therefore much faster inference time. This can be considered as some of the processing was done by the optics (in the speed of light, with no processing resources needed).

For example, for a full-HD image as an input, our proposed network evaluates a full-HD depth map in $0.22[\text{sec}]$ (using Nvidia Titan X Pascal GPU). For the same sized input on the same GPU, the net presented in [75] evaluates a 3-times smaller depth map in $10[\text{sec}]$ (Timing was done by us using the same machine and the implementation of the network in [75] that is available at the authors' website). Of course, if a one-to-one input image to depth map ratio is not needed, the output size can be reduced, and our FCN will run even faster.

Another advantage of our method is that the depth estimation relies mostly on local cues in the image. This allows performing the computations in a distributed manner. The image can be split, and the depth map can be evaluated in parallel on different resources. The partial outputs can be recombined later with barely visible block artifacts.

5.6 3D modeling

As mentioned in the previous sections, our system is designed to handle Ψ range of $[-4,10]$ but the metric range depended on the focus point selection (see Section 5.4.4). This codependency allows one to use the same FCN model with different optical configurations.

To demonstrate this important advantage, we simulated an image (Fig. 56 (top)) captured with aperture of $3.45[mm]$ (1.5 the size of our original aperture used for training). The larger aperture provides better metrical accuracy in exchange of reducing the dynamic range. The focus point was set to $48[cm]$, providing a working range of $39[cm]$ to $53[cm]$. We then produced an estimated depth map, which was translated into point cloud data using the camera parameters (sensor size and lens focal length) from Blender. The 3D face reconstruction, shown in Fig. 56(bottom), also validates our metrical depth estimation capabilities as we were able to create this 3D model in real time.

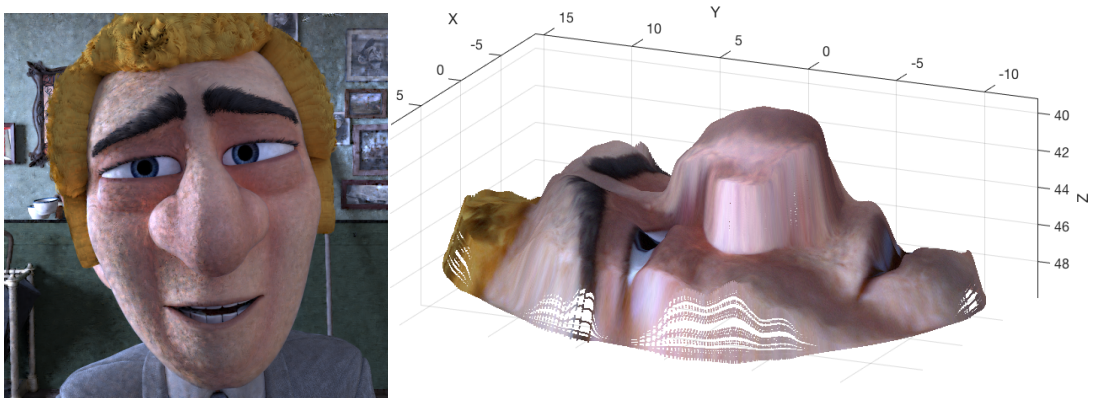


Fig. 56: 3D face reconstruction. Input image (left) and point cloud output (right).

5.7 Chapter summary

In this chapter, a method for real-time depth estimation from a single image using a phase coded aperture camera, was presented. The phase mask is designed together with the FCN model using back propagation, which allows capturing of images with high light efficiency and color-coded depth cues, such that each color channel responds differently to OOF scenarios. Taking advantage of this coded information, a simple convolutional neural network architecture is proposed to recover the depth map of the captured scene.

This proposed scheme outperforms state-of-the-art monocular depth estimation methods by having better accuracy, more than an order of magnitude speed acceleration, less memory requirements and hardware parallelization compliance. In addition, our simple and low-cost solution shows comparable performance to commercial solutions such as the Lytro cameras that cost hundreds of dollars.

Moreover, as opposed to the relative depth maps produced by those monocular methods and the Lytro camera, our system provides an absolute (metric) depth estimation, which can be useful to many computer vision applications, such as 3D modeling and augmented reality.

6. Thesis summary

The synergy between hardware and software is what make the field of computational photography so exciting. The art in computational photography is to bring together different methods to create new type of systems. The unlimited design possibilities can overcome some of the most challenging problems in imaging and processing. As conventional cameras are bound by optical and sampling lows, the new and exciting field of CP is the obvious next step in cameras evolution.

In this research, a low cost, thin phase mask plate, is utilized alongside with several dedicated computational stages, to produce computational camera which enable EDOF imaging the depth estimation in real-time. A specifically attention was given to creating a small-scale camera for future implementation in smartphones and to allow design flexibility for various of different camera configurations. This dissertation demonstrates the complete design process of creating a computational camera, form optical design and fabrication, through algorithm implementation. The capabilities of this system have been demonstrated with real-life scenarios which offered some competitive results to commercial cameras such as Lytro. With the rise of deep learning in the last years, future studies should be dedicated to finding other usage for optic manipulation based imaging system.

7. References

[1] J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed. New York: McGraw-Hill, 1996.

[2] J. C. Wyant and K. Creath, *APPLIED OPTICS AND OPTICAL ENGINEERING, VOL. Xl Basic Wavefront Aberration Theory for Optical Metrology*. Boston : Academic Press, 1992.

[3] M. Born and E. Wolf, *Principles of Optics 7th edition*. 1999.

[4] H. J. Matthews, D. K. Hamilton, and C. J. R. Sheppard, "Aberration Measurement by Confocal Interferometry," *J. Mod. Opt.*, vol. 36, no. 2, pp. 233–250, Feb. 1989.

[5] H. Nomura and T. Sato, "Techniques for measuring aberrations in lenses used in photolithography with printed patterns," *Appl. Opt.*, vol. 38, no. 13, p. 2800, May 1999.

[6] H. Nomura, K. Tawarayama, and T. Kohno, "Aberration measurement from specific photolithographic images: a different approach," *Appl. Opt.*, vol. 39, no. 7, p. 1136, Mar. 2000.

[7] F. Wang, X. Wang, M. Ma, D. Zhang, W. Shi, and J. Hu, "Aberration measurement of projection optics in lithographic tools by use of an alternating phase-shifting mask," *Appl. Opt.*, vol. 45, no. 2, p. 281, Jan. 2006.

[8] A. R. Lupini, P. Wang, P. D. Nellist, A. I. Kirkland, and S. J. Pennycook, "Aberration measurement using the Ronchigram contrast transfer function," *Ultramicroscopy*, vol. 110, no. 7, pp. 891–898, Jun. 2010.

[9] X. Chen, J. Zhou, and W. Shen, "Analytical computation of the derivative of PSF for the optimization of phase mask in wavefront coding system," *Opt. Express*, vol. 24, no. 18, p. 21070, Sep. 2016.

[10] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," 1991.

[11] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, 1996, pp. 31–42.

[12] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, 1996, pp. 43–54.

[13] B. Wilburn *et al.*, "High performance imaging using large camera arrays," *ACM*

Trans. Graph., vol. 24, no. 3, p. 765, 2005.

[14] E. H. Adelson and J. Y. A. Wang, "Single Lens Stereo with a Plenoptic Camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, 1992.

[15] R. Ng, M. Levoy, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-held Plenoptic Camera," *Informational*, pp. 1–11, 2005.

[16] T. Georgiev, Z. Yu, A. Lumsdaine, and S. Goma, "Lytro camera technology: theory, algorithms, performance analysis," 2013.

[17] H. G. Jeon *et al.*, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 1547–1555.

[18] Z. Ma, Z. Cen, and X. Li, "Depth estimation algorithm for light field data by epipolar image analysis and region interpolation," *Appl. Opt.*, vol. 56, no. 23, p. 6603, Aug. 2017.

[19] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: Motion deblurring using fluttered shutter," *Acm Trans. Graph.*, vol. 25, no. 3, pp. 795–804, 2006.

[20] A. Agrawal and R. Raskar, "Optimal single image capture for motion deblurring," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, pp. 2560–2567.

[21] Y. W. Tai, N. Kong, S. Lin, and S. Y. Shin, "Coded exposure imaging for projective motion deblurring," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2408–2415.

[22] S. McCloskey, "Temporally coded flash illumination for motion deblurring," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 683–690.

[23] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 287–294.

[24] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: Programmable pixel compressive camera for high speed imaging," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 329–336.

[25] G. Huang, H. Jiang, K. Matthews, and P. Wilford, "Lensless imaging by

compressive sensing," in *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 2013, pp. 2101–2105.

[26] M. S. Asif, A. Ayremlou, A. Veeraraghavan, R. Baraniuk, and A. Sankaranarayanan, "FlatCam: Replacing Lenses with Masks and Computation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2016, vol. 2016–Febru, pp. 663–666.

[27] G. Kim, K. Isaacson, R. Palmer, and R. Menon, "Lensless photography with only an image sensor," *Appl. Opt.*, vol. 56, no. 23, p. 6450, 2017.

[28] A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," *Optica*, vol. 4, no. 9, p. 1117, Sep. 2017.

[29] E. R. Dowski and W. T. Cathey, "Extended depth of field through wave-front coding," *Appl. Opt.*, vol. 34, no. 11, p. 1859, Apr. 1995.

[30] W. T. Cathey and E. R. Dowski, "New paradigm for imaging systems," *Appl. Opt.*, vol. 41, no. 29, p. 6080, Oct. 2002.

[31] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, 2007.

[32] E. Ben-Eliezer, Z. Zalevsky, E. Marom, and N. Konforti, "All-optical extended depth of field imaging system," *J. Opt. A Pure Appl. Opt.*, vol. 5, no. 5, pp. S164–S169, Sep. 2003.

[33] E. Ben-Eliezer, N. Konforti, B. Milgrom, and E. Marom, "An optimal binary amplitude-phase mask for hybrid imaging systems that exhibit high resolution and extended depth of field," *Opt. Express*, vol. 16, no. 25, p. 20540, Dec. 2008.

[34] B. Milgrom, N. Konforti, M. A. Golub, and E. Marom, "Pupil coding masks for imaging polychromatic scenes with high resolution and extended depth of field," *Opt. Express*, vol. 18, no. 15, pp. 15569–15584, 2010.

[35] B. Milgrom, N. Konforti, M. A. Golub, and E. Marom, "Novel approach for extending the depth of field of Barcode decoders by using RGB channels of information," *Opt. Express*, vol. 18, no. 16, pp. 17027–17039, 2010.

[36] H. Haim, A. Bronstein, and E. Marom, "Computational multi-focus imaging combining sparse model with color dependent phase mask.," *Opt. Express*, vol. 23, no. 19, pp. 24547–56, Sep. 2015.

[37] O. Cossairt and S. Nayar, "Spectral Focal Sweep: Extended depth of field from chromatic aberrations," in *2010 IEEE International Conference on Computational*

Photography (ICCP), 2010, pp. 1–8.

[38] F. Guichard, H.-P. Nguyen, R. Tessières, M. Pyanet, I. Tarchouna, and F. Cao, "Extended depth-of-field using sharpness transport across color channels," in *IS&T/SPIE Electronic Imaging*, 2009, p. 72500N–72500N–12.

[39] P. Mouroulis, "Depth of field extension with spherical optics," *Opt. Express*, vol. 16, no. 17, p. 12995, Aug. 2008.

[40] H. Tang and K. N. Kutulakos, "Utilizing Optical Aberrations for Extended-Depth-of-Field Panoramas," Springer, Berlin, Heidelberg, 2013, pp. 365–378.

[41] J. L. Starck, E. Pantin, and F. Murtagh, "Deconvolution in Astronomy: a review," *Publ. Astron. Soc. Pacific*, vol. 114, no. 800, pp. 1051–1069, Oct. 2002.

[42] S. Elmalem, N. Konforti, and E. Marom, "Polychromatic imaging with extended depth of field using phase masks exhibiting constant phase over broad wavelength band," *Appl. Opt.*, vol. 52, no. 36, p. 8634, Dec. 2013.

[43] S. Kuthirummal, H. Nagahara, C. Zhou, and S. K. Nayar, "Flexible depth of field photography.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 58–71, Jan. 2011.

[44] O. Cossairt, C. Zhou, and S. Nayar, "Diffusion coded photography for extended depth of field," *ACM Trans. Graph.*, vol. 29, no. 4, p. 1, Jul. 2010.

[45] C. Zhou, S. Lin, and S. K. Nayar, "Coded aperture pairs for depth from defocus and defocus deblurring," *Int. J. Comput. Vis.*, vol. 93, no. 1, pp. 53–72, 2011.

[46] A. Chakrabarti and T. Zickler, "Depth and Deblurring from a Spectrally-Varying Depth-of-Field," Springer, Berlin, Heidelberg, 2012, pp. 648–661.

[47] H. Haim, A. Bronstein, and E. Marom, "Multi-Focus imaging using optical phase mask," in *Classical Optics 2014*, p. CTh2C.6.

[48] H. H. Baker and T. Binford, "Depth From Edge and Intensity Based Stereo," *7th Int. Jt. Conf. Artif. Intell.*, pp. 631–636, 1981.

[49] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Imaging*, vol. 130, no. x, pp. 674–679, 1981.

[50] F. H. Sinz, J. Q. Candela, G. H. Bakır, C. E. Rasmussen, and M. O. Franz, "Learning Depth from Stereo," 2004, pp. 245–252.

[51] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *Int. J. Comput. Vis.*, vol. 35, no. 3, pp. 269–293, 1999.

[52] R. Horaud, G. Csurka, and D. Demirdijian, "Stereo calibration from rigid motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1446–1452,

2000.

[53] S. D. Blostein and T. S. Huang, “Error Analysis in Stereo Determination of 3-D Point Positions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 6, pp. 752–765, 1987.

[54] S. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor-system description, issues and solutions,” *Comput. Vis. Pattern …*, vol. 00, no. C, pp. 35–35, 2004.

[55] F. Blais, “Review of 20 years of range sensor development,” *J. Electron. Imaging*, vol. 13, no. 1, p. 231, 2004.

[56] S. S. Gorthi and P. Rastogi, “Fringe projection techniques: Whither we are?,” *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 133–140, 2010.

[57] C. Guan, L. Hassebrook, and D. Lau, “Composite structured light pattern for three-dimensional video.,” *Opt. Express*, vol. 11, no. 5, pp. 406–417, 2003.

[58] V. G. Yalla and L. G. Hassebrook, “Very high resolution 3D surface scanning using multi-frequency phase measuring profilometry,” in *Proc. SPIE 5798, Spaceborne Sensors II*, 2005, vol. 5798, pp. 44–53.

[59] H. Sarbolandi, D. Lefloch, and A. Kolb, “Kinect range sensing: Structured-light versus Time-of-Flight Kinect,” *Comput. Vis. Image Underst.*, vol. 139, pp. 1–20, 2015.

[60] N. Silberman and R. Fergus, “Indoor scene segmentation using a structured light sensor,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 601–608.

[61] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor Segmentation and Support Inference from RGBD Images,” Springer, Berlin, Heidelberg, 2012, pp. 746–760.

[62] P. Grossmann, E. S. Publishers, P. Grossmann, and E. S. Publishers, “Depth from focus,” *Pattern Recognit. Lett.*, vol. 5, no. 1, pp. 63–69, 1987.

[63] Y. Y. Schechner and N. Kiryati, “Depth from Defocus vs. stereo: How different really are they?,” *Int. J. Comput. Vis.*, vol. 39, no. 2, pp. 141–162, 2000.

[64] S. Suwajanakorn, C. Hernandez, and S. M. Seitz, “Depth From Focus With Your Mobile Phone.” pp. 3497–3506, 2015.

[65] H. Kim, C. Richardt, and C. Theobalt, “Video depth-from-defocus,” in *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 2016, pp. 370–379.

[66] M. Subbarao and G. Surya, “Depth from defocus: A spatial domain approach,” *Int. J. Comput. Vis.*, vol. 13, no. 3, pp. 271–294, 1994.

[67] S. Chaudhuri and A. N. Rajagopalan, *Depth From Defocus: A Real Aperture Imaging Approach*. New York, NY: Springer New York, 1999.

[68] Y. W. Tai and M. S. Brown, “Single image defocus map estimation using local contrast prior,” in *Proceedings - International Conference on Image Processing, ICIP*, 2009, pp. 1797–1800.

[69] S. Zhuo and T. Sim, “Defocus map estimation from a single image,” in *Pattern Recognition*, 2011, vol. 44, no. 9, pp. 1852–1858.

[70] C. Tang, C. Hou, and Z. Song, “Defocus map estimation from a single image via spectrum contrast,” *Opt. Lett.*, vol. 38, no. 10, p. 1706, 2013.

[71] S. Liu, F. Zhou, and Q. Liao, “Defocus Map Estimation from a Single Image Based on Two-Parameter Defocus Model,” *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5943–5956, 2016.

[72] E. B. Goldstein, *Sensation and perception*, 9th ed. 2013.

[73] A. Saxena, Min Sun, and A. Y. Ng, “Make3D: Learning 3D Scene Structure from a Single Still Image,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[74] D. Eigen, C. Puhrsch, and R. Fergus, “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network.” pp. 2366–2374, 2014.

[75] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[76] W. Chen, Z. Fu, D. Yang, and J. Deng, “Single-Image Depth Perception in the Wild.” pp. 730–738, 2016.

[77] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, “Learning ordinal relationships for mid-level vision,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 388–396.

[78] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[79] E. J. Candes, “Compressive sampling,” *Univ. PA. Law Rev.*, vol. 3, pp. 1433–1452, 2006.

[80] E. J. Candes and M. B. Wakin, “An Introduction To Compressive Sampling,” *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[81] Y. C. Eldar and G. Kutyniok, *Compressed sensing : theory and applications*. Cambridge University Press, 2012.

[82] S. Vasanawala *et al.*, "Practical parallel imaging compressed sensing MRI: Summary of two years of experience in accelerating body MRI of pediatric patients," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 1039–1043.

[83] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed Sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.

[84] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (PICCS): A method to accurately reconstruct dynamic CT images from highly undersampled projection data sets," *Med. Phys.*, vol. 35, no. 2, pp. 660–663, Jan. 2008.

[85] Z. Chen, A. Basarab, and D. Kouame, "Compressive Deconvolution in Medical Ultrasound Imaging," *IEEE Trans. Med. Imaging*, vol. 35, no. 3, pp. 728–737, 2016.

[86] B. Shin, S. Jeon, J. Ryu, and H. J. Kwon, "Compressed Sensing for Elastography in Portable Ultrasound," *Ultrason. Imaging*, p. 016173461771693, 2017.

[87] M. Elad, *Sparse and Redundant Representations : From Theory to Applications in Signal and Image Processing*. New York: Springer, 2010.

[88] F. Couzinie-Devy, J. Mairal, F. Bach, and J. Ponce, "Dictionary learning for deblurring and digital zoom," *arXiv Prepr. arXiv1110.0957*, 2011.

[89] Z. Hu, J. Bin Huang, and M. H. Yang, "Single image deblurring with adaptive dictionary learning," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 1169–1172.

[90] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, "Sparse representation based blind image deblurring," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 2011, pp. 1–6.

[91] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *Image Process. IEEE Trans.*, vol. 21, no. 8, pp. 3467–3478, 2012.

[92] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *ACM Transactions on Graphics (TOG)*, 2008, vol. 27, no. 3, p. 73.

[93] M. S. C. Almeida and L. B. Almeida, "Blind and semi-blind deblurring of natural images.," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 36–52, Jan. 2010.

[94] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen, "Blind motion deblurring from a single image using sparse approximation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 104–111.

[95] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[96] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, 2008.

[97] M. J. Fadili, J. L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *Comput. J.*, vol. 52, no. 1, pp. 64–79, Feb. 2009.

[98] M. Elad, J. L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, Nov. 2005.

[99] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[100] X. Huang and O. Cossairt, "Dictionary Learning Based Color Demosaicing for Plenoptic Cameras," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 455–460.

[101] R. Tibshirani, "Regression Selection and Shrinkage via the Lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1. pp. 267–288, 1996.

[102] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[103] W. J. Fu, "Penalized regressions: The bridge versus the lasso?," *J. Comput. Graph. Stat.*, vol. 7, no. 3, pp. 397–416, 1998.

[104] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *Ann. Appl. Stat.*, vol. 2, no. 1, pp. 224–244, 2008.

[105] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Math. Program.*, vol. 117, no. 1–2, pp. 387–423, 2009.

[106] B. Efron *et al.*, "Least angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

[107] R. D. Nowak and M. a. T. Figueiredo, "Fast wavelet-based image deconvolution using the EM algorithm," *Conf. Rec. Thirty-Fifth Asilomar Conf. Signals, Syst.*

Comput., vol. 1, 2001.

[108] J. L. Starck, D. L. Donoho, and E. J. Candès, "Astronomical image representation by the curvelet transform," *Astron. Astrophys.*, vol. 398, no. 2, pp. 785–800, 2003.

[109] M. a T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration.," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, 2003.

[110] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[111] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, 2007.

[112] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[113] K. Gregor and Y. Lecun, "Learning Fast Approximations of Sparse Coding," *Vision, Image Signal Process. IEE Proc.* -, 2010.

[114] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, pp. 40–44, 1993.

[115] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[116] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Inf. Theory, IEEE Trans.*, vol. 53, no. 12, pp. 4655–4666, 2007.

[117] H. Shan, J. Ma, and H. Yang, "Comparisons of wavelets, contourlets and curvelets in seismic denoising," *J. Appl. Geophys.*, vol. 69, no. 2, pp. 103–115, 2009.

[118] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[119] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[120] S. Mallat, *A Wavelet Tour of Signal Processing*. 2009.

[121] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no.

12, pp. 2091–2106, 2005.

[122] M. N. Do and M. Vetterli, “Framing pyramids,” *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2329–2342, 2003.

[123] E. Le Pennec and S. Mallat, “Sparse geometric image representations with bandelets,” *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, 2005.

[124] E. Le Pennec and S. Mallat, “Bandelet Image Approximation and Compression,” *Multiscale Model. Simul.*, vol. 4, no. 3, pp. 992–1039, 2005.

[125] E. Candes, D. L. Donoho, E. J. Candès, and D. L. Donoho, “Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges,” *Curves Surf. Fitting*, vol. C, no. 2, pp. 1–10, 2000.

[126] E. J. Candès and D. L. Donoho, “New tight frames of curvelets and optimal representations of objects with piecewise C 2 singularities,” *Commun. Pure Appl. Math.*, vol. 57, no. 2, pp. 219–266, 2004.

[127] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *Ieee Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[128] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968.

[129] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[130] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[131] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097--1105.

[132] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 1–9.

[133] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Int. Conf. Learn. Represent.*, pp. 1–14, 2015.

[134] a Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” *Icassp*, no. 3, pp. 6645–6649, 2013.

[135] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Aug. 2014.

[136] D. Chicco, P. Sadowski, and P. Baldi, “Deep autoencoder neural networks for gene ontology annotation predictions,” *Proc. 5th ACM Conf. Bioinformatics, Comput. Biol. Heal. Informatics - BCB ’14*, pp. 533–540, 2014.

[137] A. Badhe, “Using Deep Learning Neural Networks To Find Best Performing Audience Segments,” *Int. J. Sci. Technol. Res.*, vol. 5, no. 04, 2016.

[138] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for Simplicity: The All Convolutional Net,” Dec. 2014.

[139] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 448–456.

[140] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[141] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.

[142] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation ppt,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 8828, no. CNN;, pp. 3431–3440, 2015.

[143] O. Matan, J. C. Burges, Y. LeCun, and J. S. Denker, “Multi-digit recognition using a space displacement neural network,” *Proc. NIPS*, pp. 488–495, 1992.

[144] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” *Proc. COMPSTAT’2010*, pp. 177–186, 2010.

[145] A. Choromanska, M. Henaff, M. Mathieu, … G. A.-A. I. and, and U. 2015, “The loss surfaces of multilayer networks,” *Artif. Intell. Stat.*

[146] S. Franssila, *Introduction to microfabrication*. John Wiley & Sons, 2010.

[147] T. Remez, O. Litany, S. Yoseff, H. Haim, and A. Bronstein, “FPGA system for real-time computational extended depth of field imaging using phase aperture coding,” *arXiv Prepr. arXiv1608.01074*, Aug. 2016.

[148] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Supervised dictionary learning,” *Adv. Neural Inf. Process. Syst.*, pp. 1033–1040, 2008.

[149] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, 2012.

[150] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, “Domain Adaptive Dictionary

Learning," Springer, Berlin, Heidelberg, 2012, pp. 631–645.

[151] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 233–240.

[152] M. S. C. Almeida and L. B. Almeida, "Blind deblurring of foreground-background images," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 1301–1304.

[153] W. Zhang and W. K. Cham, "Single-image refocusing and defocusing," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 873–882, Feb. 2012.

[154] Kodak, "KODAK Dataset." [Online]. Available: http://r0k.us/graphics/kodak/.

[155] Brodatz, "Colored Brodatz Texture Database." [Online]. Available: http://multibandtexture.recherche.usherbrooke.ca/colored _brodatz.html.

[156] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning Efficient Sparse and Low Rank Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1821–1833, Dec. 2015.

[157] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[158] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-Sparse Signals: Uncertainty Relations and Efficient Recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[159] J. Huang and T. Zhang, "The benefit of group sparsity," *Ann. Stat.*, vol. 38, no. 4, pp. 1978–2004, Aug. 2010.

[160] Y. Cao, Z. Wu, and C. Shen, "Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2017.

[161] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," Sep. 2016.

[162] H. Jung and K. Sohn, "Single Image Depth Estimation With Integration of Parametric Learning and Non-Parametric Sampling," *J. Korea Multimed. Soc.*, vol. 19, no. 9, pp. 1659–1668, Sep. 2016.

[163] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 239–248.

[164] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image

Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778, 2016.

[165] P. A. Shedligeri, S. Mohan, and K. Mitra, "Data Driven Coded Aperture Design for Depth Recovery," May 2017.

[166] M. Martinello *et al.*, "Dual Aperture Photography: Image and Depth from a Mobile Camera," in *2015 IEEE International Conference on Computational Photography (ICCP)*, 2015, pp. 1–10.

[167] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely Connected Convolutional Networks," Aug. 2016.

[168] A. Vedaldi, "Describing Textures in the Wild." pp. 3606–3613, 2014.

[169] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7577 LNCS, no. PART 6, Springer, Berlin, Heidelberg, 2012, pp. 611–625.

Appendix A – Fast mask search

As presented in Section 3.3, the search for the optimal mask parameters required calculation of the PSF derivative, whereby according to Eq. (46), one needs to compute the FFT of two $n \times n$ matrices (size of the sample grid of the pupil function). The FFT operation is the most computationally expensive part of this process. If we set the phase search to $\phi = [0,15] rad$ with a $0.5 rad$ accuracy, the number of phases under consideration for a configuration consisting of three rings (see Fig. 57) is ~30k. The hereby presented method reduces the required FFT operation for each scenario, by replacing it with matrix addition. This produces a speed up of up to 30 times.

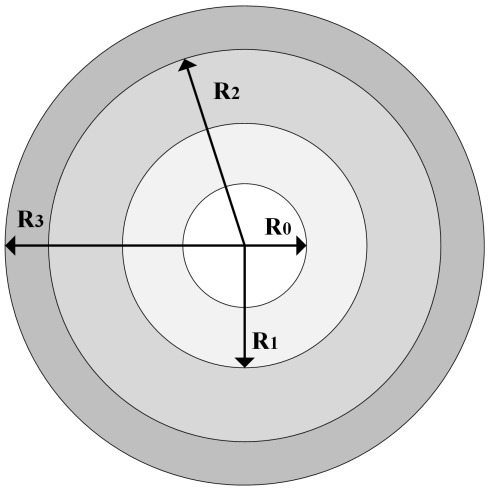


Fig. 57: Phase mask's rings location

The radial symmetric pupil consists of mask radii R_0, R_1, R_2, R_3 as illustrated in Fig. 57, and provides four phases $\phi_0, \phi_1, \phi_2, \phi_3$ (from center to outer ring). Since the phase is relative the first phase ϕ_0 is set to zero. For a given focus scenario, this pupil can be represented as a sum of four circular aperture:

$$
\begin{aligned}
P(r,\Psi) = P_\Psi(r) \quad &= \quad circ\left(\frac{r}{R_3}\right) e^{j\Psi\left(\frac{r^2}{R_3^2}\right)} \cdot \exp\{j\phi_3\} \\
&+ \quad circ\left(\frac{r}{R_2}\right) e^{j\Psi\left(\frac{r^2}{R_3^2}\right)} \cdot \left(\exp\{j\phi_2\} - \exp\{j\phi_3\}\right) \\
&+ \quad circ\left(\frac{r}{R_1}\right) e^{j\Psi\left(\frac{r^2}{R_3^2}\right)} \cdot \left(\exp\{j\phi_1\} - \exp\{j\phi_2\}\right) \\
&+ \quad circ\left(\frac{r}{R_0}\right) e^{j\Psi\left(\frac{r^2}{R_3^2}\right)} \cdot \left(\exp\{j\phi_0\} - \exp\{j\phi_1\}\right)
\end{aligned}
$$

If we define $\Phi_m = \exp\{j\phi_m\} - \exp\{j\phi_{m+1}\}$ for $m = \{0,1,2\}$ and $\Phi_3 = \exp\{j\phi_3\}$,the pupil be expressed as:

$$\begin{aligned} P(r;\Psi) &= \sum_{m=0}^{3} P_m(r;\Psi)\cdot\Phi_m \\ P_m(r;\Psi) &= e^{j\Psi\left(\frac{r^2}{R_3^2}\right)}\cdot circ\left(\frac{r}{R_m}\right) \end{aligned} \tag{69}$$

Now, using Eq. (46), the PSF derivative is:

$$\begin{aligned} \frac{dh_\Psi(r)}{d\Psi} &= \mathrm{Im}\left\{\mathbb{F}_{2D}\left\{P(r,\Psi)r^2\right\}\cdot\overline{\mathbb{F}_{2D}\left\{P(r,\Psi)\right\}}\right\} \\ &= \mathrm{Im}\left\{\mathbb{F}_{2D}\left\{\sum_{m=0}^{3}P_m(r;\Psi)\cdot\Phi_m\cdot r^2\right\}\cdot\overline{\mathbb{F}_{2D}\left\{\sum_{m=0}^{3}P_m(r;\Psi)\cdot\Phi_m\right\}}\right\} \\ &= \mathrm{Im}\left\{\sum_{m=0}^{3}\sum_{n=0}^{3}\Phi_m\overline{\Phi_n}\cdot\mathbb{F}_{2D}\left\{P_m(r;\Psi)\cdot r^2\right\}\overline{\mathbb{F}_{2D}\left\{P_n(r;\Psi)\right\}}\right\} \end{aligned} \tag{70}$$

Defining a new function F_{mn} as:

$$F_{mn} = \mathbb{F}_{2D}\left\{P_m(r;\Psi)\cdot r^2\right\}\cdot\overline{\mathbb{F}_{2D}\left\{P_n(r;\Psi)\right\}} \tag{71}$$

Eq. (70) reduces to:

$$\begin{aligned} \frac{dh_\Psi(r)}{d\Psi} &= \mathrm{Im}\left\{\sum_{m=0}^{3}\sum_{n=0}^{3}\Phi_m\overline{\Phi_n}\cdot F_{mn}\right\} \\ &= \sum_{m=0}^{3}\left|\Phi_m\right|^2\cdot\mathrm{Im}\left\{F_{mm}\right\} + \sum_{n=0}^{3}\sum_{\substack{m=0\\m\neq n}}^{3}\mathrm{Im}\left\{\Phi_m\overline{\Phi_n}\cdot F_{mn}\right\} \end{aligned} \tag{72}$$

Notice that the second term in Eq. (72) consists of the imaginary part of the multiplication between the mask parameters parameter and F_{mn}. We can isolate the mask parameters from the F_{mn} matrices such that:

$$\begin{aligned} \frac{dh_\Psi(r)}{d\Psi} &= \sum_{m=0}^{2}\left|\Phi_m\right|^2\cdot\mathrm{Im}\left\{F_{mm}\right\} \\ &+ \sum_{n=0}^{2}\sum_{\substack{m=1\\m>n}}^{3}\mathrm{Im}\left\{\Phi_m\overline{\Phi_n}\right\}\mathrm{Re}\left\{F_{mn}-F_{nm}\right\} + \mathrm{Re}\left\{\Phi_m\overline{\Phi_n}\right\}\mathrm{Im}\left\{F_{mn}+F_{nm}\right\} \end{aligned} \tag{73}$$

For a given Ψ and a mask parameters, numerical solution of Eq. (46) requires only two FFT operations while the expressing in Eq. (73) requires eight FFT operation. However, the FFT stage in Eq. (73) is done only ones for each Ψ; given a new mask parameters,

Now we only require calculating ten matrix additions. This results in a speed up of about 30 times in comparison to using the direct expression of Eq. (46) (Using MATLAB).

In practice, we accelerated the process by calculating a known RGB mask (4π for example) merit function (Eq. (49)). For the first Ψ parameter, the Joint PSFDE value was calculated for each mask. Masks whose Joint PSFDE score was lower than our benchmark mask merit score, were removed from the search, making the next iteration faster.

תקציר

הזמינות של הטלפונים החכמים, אשר הופיעו לראשונה בעשור האחרון, הפכה את כולם לצלמים. נכון ל-2017 בארה"ב יש יותר מצלמות טלפונים חכמים מאשר אנשים. הופעתם של חיישני התמונה הדיגיטליים יצרו מהפכה בתחום הצילום. האיכות של צילום דיגיטלי נקבעת על ידי האופטיקה, החיישן הדיגיטלי ועיבוד התמונה. עבור מצלמות טלפונים חכמים, הנפח הזמין עבור מערכת העדשות הינו קטן ולכן הפתרונות האופטיים לשיפור איכות התמונה מוגבלים. בעוד שהטכנולוגיות הקונבנציונליות בתחום האופטיקה והחיישנים הדיגיטליים הגיעו לשיאם, רב הפיתוחים בתחום כיום מתמקדים בתחום עיבוד התמונה.

תחום ה"צילום הממוחשב" גדל בשנים האחרונות ומשך את תשומת ליבם של חברות הטכנולוגיה המובילות בעולם כגון גוגל, אפל וסמסונג. בצילום ממוחשב, מבוצעת מניפולציה בתהליך הרכשת התמונות על מנת לאפשר עיבוד יעיל יותר, אשר יכול לשמש למגוון רחב של יישומים בעיבוד תמונה ובתחום והראיה הממוחשבת. במסגרת עבודה זו נחקרה מצלמה ממוחשבת אשר תוכננה במיוחד עבור הטמעה במצלמות קטנות (מצלמות טלפונים). מצלמה זו מאפשרת צילום בעומק שדה מוגדל בנוסף ליצירת מפת עומק מתמונה בודדת. יכולות אלו מאפשרות פונקציות צילום מתקדמות כגון שינוי הפוקוס לאחר הצילום ומידול תלת-ממדי, אשר יכולים להשתלב ביישומים כדוגמת מציאות רבודה ומכוניות אוטונומיות.

אחד האתגרים המסובכים בתחום הצילום הינו שיחזור של תמונות אשר לא בפוקוס. בעיה זו ידועה כ"לא מוגדרת היטב" מאחר שחלק מהמידע נעלם בתהליך ההדמיה. מסיכה פאזה בינארית וסימטרית, מציעה פתרון אופטי זול להגדלת עומק שדה, אשר מספק איכות תמונה מספקת עבור אפליקציות בתחום הראיה הממוחשבת כגון קריאת ברקודים וזיהוי פנים. במהלך עבודה זו נעשה שימוש במסכת RGB, אשר יוצרת תגובה ייחודית לאדום, ירוק וכחול, כך שמתקבלות שלוש תמונות בו זמנית, כל אחת עם תגובת פוקוס שונה.

חלקה העיקרי של עבודה זו מוקדש לפיתוח שיטות לאיחוד שלושת ערוצי הצבע לתמונה צבעונית אחת בעלת עומק שדה מוגדל ומראה משופר, על-ידי אלגוריתם עיבוד יעיל אשר מבוסס על מודל ייצוג דליל, אשר הותאם ספציפי למערכת צילום זו. שלב העיבוד מומש גם על גבי לוח FPGA כך שהתקבלה מערכת צילום ממוחשב בזמן אמת אשר יודעת להתמודד באופן עיוור עם סצנות שבהן מספר אובייקטים במרחקים שונים מהמצלמה. מערכת זו תהיה אידיאלית עבור צילום יום-יומי של סצנות "טבעיות".

בחלקה השני של עבודה זו יוצגו שתי שיטות ליצירת מפת עומק מתמונה בודדת. מסיכת ה-RGB מספקת תגובת צבע תלוית עומק, היוצרת תמונה עם רמזים כרומטיים. רמזים אלו מכילים מידע על מצב הפוקוס עבור כל פיקסל ויכולים לשמש לשערוך מפת פוקוס, אותה ניתן להמיר בנקל למפת עומק מטרית. השיטה הראשונה מבוססת על מודל הייצוג הדליל אשר שימש לצילום בעומק שדה מוגדל. השיטה השנייה מתמקדת על מימוש מהיר של מפת העומק אשר מתבסס על רשתות עצביות מלאכותיות. התוצאות סימולטיביות והניסיוניות המוצגות בעבודה זו מדגימים יצירת מפת עומק מדויקת בזמן אמת.

תוכן עניינים

אוניברסיטת תל-אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן

בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

שילוב מסיכת פאזה במערכות אופטיות לשחזור תמונות מטושטשות ומדידת עומק מתמונה יחידה

הראל חיים

חיבור לשם קבלת תואר ״דוקטור לפילוסופיה״

הוגש לסנאט של אוניברסיטת תל אביב

עבודה זו נעשתה באוניברסיטת תל אביב בפקולטה להנדסה
בהדרכת פרופ׳ עמנואל מרום פרופ׳ אלכס ברונשטיין

כסלו תשע״ח

אוניברסיטת תל-אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן

בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

שילוב מסיכת פאזה במערכות אופטיות לשחזור תמונות מטושטשות ומדידת עומק מתמונה יחידה

חיבור לשם קבלת תואר ״דוקטור לפילוסופיה״

הראל חיים

הוגש לסנאט של אוניברסיטת תל אביב

כסלו תשע״ח