

# Extended Scene Deep Learning Wavefront Sensing for Real Time Image Deconvolution

V.S. de Bruijne

Master of Science Thesis



# **Extended Scene Deep Learning Wavefront Sensing for Real Time Image Deconvolution**

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at  
Delft University of Technology

V.S. de Bruijne

March 19, 2021



Copyright © Delft Center for Systems and Control (DCSC)  
All rights reserved.



---

# Abstract

Ground based telescope imaging suffers from interference from the earth's atmosphere. Fluctuations in the refractive index of the air delay incoming light randomly, resulting in blurred images. A deconvolution from wavefront sensing system is an adaptive optics system that measures the modes in which the light is corrupted (i.e. the wavefront) and corrects it using a process called deconvolution. The wavefront is measured using a wavefront sensor, which consists of an array of microlenses combined with an imaging sensor. Each microlens casts an image of the object unto the imaging sensor, resulting in a collection of images that are differently aberrated depending on their location on the sensor. Conventionally, the wavefront is calculated by measuring the shifts of each microlens image and integrating these shifts over the aperture. This method, however, discards information about the higher order deformations of the microlens images.

In this thesis, a novel method of wavefront reconstruction has been developed which makes use of artificial neural networks in order to extract this higher order information. In order to do this, the images produced by the microlenses are normalized, which is done using a modified version of the blind deconvolution algorithm called TIP. After the normalization, the microlens images are reduced to what they would look like if a point source was observed, instead of the object. With the influence of the object removed, an artificial neural network is used for the estimation of the wavefront.

By using this method, the wavefront can be reconstructed with twice the turbulence strength compared to what is possible with conventional methods. Combining this method with an image deconvolution step results in a real-time image correction system that works up to  $10Hz$  on the tested system, consisting of a desktop PC with an Intel Xeon E5-2630 DUAL CPU and a NVIDIA GeForce GTX 970 GPU.



---

# Samenvatting

Het beeld van een telescoop wordt vaak verstoord door de atmosfeer. Schommelingen in de brekingsindex van de lucht vertragen het licht in een willekeurige wijze, wat zorgt voor wazige beelden. Deconvolutie uit golffront metingen is een systeem dat dit soort beelden kan corrigeren door eerst het golffront (de willekeurige vertragingen van het licht) te meten, en er daarna voor te compenseren door middel van deconvolutie. Het meten van het golffront gebeurt door middel van een golffront sensor, die bestaat uit een raster van microlenzen en een camera sensor. Elke microlens vormt een beeld van het object op de camera sensor, en elk van deze beelden is op een andere manier vervormd afhankelijk van zijn locatie in het raster. Normaal gezien wordt de golffrontsensor uitgelezen door het bepalen van de verplaatsing van elk microlensbeeld en het integreren van deze verplaatsingen over de hele sensor. Door aan te nemen dat alle informatie over het golffront verscholen zit in de verplaatsing van het beeld van de microlens gaat helaas veel informatie over het golffront verloren.

In dit proefstuk wordt een nieuwe methode van het uitlezen van de golffront sensor voorgesteld. Deze methode maakt gebruik van kunstmatig neurale netwerk om meer informatie uit de golffront sensor af te leiden. Hiervoor moet eerst de invloed van het specifieke object op de beelden van de golffront sensor verwijderd worden, wat gebeurt door middel van de blinde deconvolutie methode TIP. Na deze stap ziet elk beeld van de golffront sensor eruit alsof de telescoop naar een punt bron kijkt (een ster bijvoorbeeld). Dit genormaliseerde beeld kan vervolgens worden geïnterpreteerd door een neurale netwerk.

Door het gebruik van deze methode kan het golffront worden gemeten met een dubbele turbulentie sterkte vergeleken bij conventionele methoden. Door deze methode te combineren met een deconvolutie stap kan het beeld van een telescoop in real time worden gecorrigeerd met een frequentie van tot wel  $10Hz$ . De computer die hiervoor gebruikt is heeft een Intel Xeon E5-2630 DUAL processor en een NVIDIA GeForce GTX 970 videokaart.



---

# Table of Contents

<b>Abstract</b>	<b>i</b>
<b>Samenvatting</b>	<b>iii</b>
<b>Glossary</b>	<b>xi</b>
<b>Acknowledgments</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1-1 Adaptive Optics: Theoretical Background . . . . .	1
1-1-1 Atmospheric Degradation . . . . .	1
1-1-2 Wavefront Sensing . . . . .	4
1-2 Deconvolution From Wavefront Sensing . . . . .	6
<b>2 Image Deconvolution</b>	<b>11</b>
2-1 Deconvolution . . . . .	12
2-1-1 Least-Squares Deconvolution . . . . .	12
2-1-2 Landweber Deconvolution . . . . .	13
2-2 Blind Deconvolution . . . . .	15
2-2-1 Tangential Iterative Projections . . . . .	16
<b>3 Deep Learning Wavefront Sensing</b>	<b>17</b>
3-1 Artificial Neural Networks: Theoretical Background . . . . .	17
3-1-1 Convolutional neural networks . . . . .	19
3-2 State of the art deep learning wavefront sensing . . . . .	19
3-2-1 U-net . . . . .	20
<b>4 Numerical Simulations of Optics</b>	<b>23</b>
4-1 Wavefront Simulation . . . . .	23
4-2 Point Spread Function Simulation . . . . .	25

4-2-1	Shack-Hartmann Pattern Simulation . . . . .	25
4-2-2	Main Point Spread Function Simulation . . . . .	27
4-3	Imaging Simulation . . . . .	27
<b>5</b>	<b>Deconvolution From Wavefront Sensing: System Design</b>	<b>29</b>
5-1	Pre-Processing Shack-Hartmann Images . . . . .	29
5-2	Deep Learning Wavefront Sensing . . . . .	32
5-2-1	Selection of Architecture . . . . .	32
5-2-2	Data Normalization . . . . .	34
5-3	Image Deconvolution . . . . .	35
<b>6</b>	<b>Experimental Results</b>	<b>39</b>
6-1	Experimental Setup . . . . .	39
6-2	Wavefront Sensing . . . . .	40
6-3	Real Time System Integration . . . . .	43
6-4	Examples . . . . .	44
6-5	Conclusions . . . . .	45
<b>7</b>	<b>Conclusion and Discussion</b>	<b>47</b>
7-1	Future work . . . . .	47
<b>A</b>	<b>Appendix</b>	<b>49</b>
A-1	Python Code . . . . .	49
A-1-1	Implementation of the convolutional neural network used for deep learning wavefront sensing . . . . .	49
	<b>Bibliography</b>	<b>51</b>

---

# List of Figures

1-1	The first 15 Zernike polynomials. . . . .	3
1-2	Example of a turbulent wavefront with a corresponding PSF, $D/r_0 = 12$ . . . . .	3
1-3	Schematic overview of a cross-section of a SHWFS (after [37]). . . . .	5
1-4	SH-PSF from a turbulent wavefront. . . . .	6
1-5	Example of an extended scene SH-pattern. . . . .	7
1-6	Optical setup for a deconvolution from wavefront sensing system. . . . .	8
2-1	Simulated example of the performance of a least-squares deconvolution method. . . . .	13
2-2	Simulated performance of the deconvolution methods. . . . .	15
3-1	Schematic overview of artificial neural networks . . . . .	18
3-2	Example of a CNN with two convolution and pooling layers. . . . .	20
3-3	A visualization of the general <i>U-net</i> architecture. . . . .	21
3-4	The residual block used by Bekendam [1]. . . . .	21
3-5	The residual block used by Hu et al. [12]. . . . .	22
4-1	A visualization of the generation of a turbulent phase screen as described by Lane et al. [14]. . . . .	24
4-2	Visualization of $\phi_{MLA}$ , which consists of a piece-wise linear defocus. . . . .	26
4-3	Visualization of the measured CCD noise distribution. . . . .	28
5-1	Simulated SH-image and main image. . . . .	29
5-2	The estimated SH-pattern in the first TIP iteration. . . . .	30
5-3	An example of the performance of the TIP algorithm. . . . .	32
5-4	A visualization of the modified <i>U-net</i> architecture. . . . .	33
5-5	Residual block used in the proposed CNN architecture. . . . .	34
5-6	Data cropping procedure. . . . .	35
5-7	Visualization of wavefront normalization. . . . .	36
5-8	The convergence of the Landweber and steepest descent deconvolution methods. . . . .	37

---

6-1	A schematic of the optical test setup used. . . . .	39
6-2	A visualization of the wavefronts introduced by the 2 turbulence simulators. . . . .	40
6-3	An overview of the wavefront estimation performance of the different developed deep learning wavefront sensing methods. . . . .	41
6-4	An example of the wavefront sensing performance of the 3 tested DLWS architectures. . . . .	42
6-5	A visualization of the phase unwrapping behavior of the DLWS methods. . . . .	43
6-6	A display of the performance of the system on in a simulated situation. . . . .	45
6-7	Figure displaying the real time, real world correction capabilities of the system developed in this Thesis. . . . .	45

---

## List of Tables

- 6-1 Computational efficiency evaluation of the 3 tested DLWS CNNs. The evaluation times depend greatly on the amount of background processes active. These tests were performed while the other parts of the DFWS system were also running. . . 43
- 6-2 An overview of the different sections that the software implementation is divided in, combined with their respective average computation time. Dividing the system up into these discrete parts allows for efficient parallel computation. This increases the time between receiving the image and estimating the object slightly, but allows for the calculation of multiple objects from sequential images simultaneously. . . 44



---

# Glossary

## List of Acronyms

<b>ANN</b>	Artificial neural network
<b>AO</b>	Adaptive Optics
<b>CCD</b>	Charge couple device (imaging sensor)
<b>CNN</b>	Convolutional neural network
<b>DFWS</b>	Deconvolution from wavefront sensing
<b>DLWS</b>	Deep learning wavefront sensing
<b>DM</b>	Deformable mirror
<b>FFH</b>	Frozen flow hypothesis
<b>LED</b>	Light Emitting Diode
<b>LR</b>	Lucy-Richardson
<b>MLA</b>	Microlens array
<b>MSE</b>	Mean Squared Error
<b>PSF</b>	Point spread function
<b>ReLU</b>	Rectified linear unit
<b>RMS</b>	Root mean squared
<b>RMSE</b>	Root mean squared Error
<b>SH</b>	Shack-Hartmann
<b>SHWFS</b>	Shack-Hartmann wavefront sensor

## List of Symbols

$\theta$	Arc distance
$\lambda$	Wavelength
$\phi$	Wavefront
$D$	Telescope diameter
$h$	Altitude
$i$	Image (image domain)
$I$	Image (frequency domain)
$k$	Point spread function
$K$	Point spread function (frequency domain)
$L_0$	Outer scale of wavefront
$l_0$	Outer scale of wavefront
$N$	Subaperture length (used inconsistently)
$n$	Refractive index
$o$	Object (image domain)
$O$	Object (frequency domain)
$P$	Pupil function
$r_0$	Fried parameter
$s$	Pixel spacing
$w$	Weight
$\tilde{x}$	Estimate of $x$
$x^*$	Complex conjugate of $x$
$X$	Fourier transform of $x$
$Z$	Zernike polynomial

---

# Acknowledgments

I would like to thank my daily supervisor Oleg Soloviev and the students in the Smart Imaging Lab for the fruitful discussions and, at times much needed, guidance. A special thanks to my supervisor, Gleb Vdovin, for his expert opinions and his support.

I would also like to thank Alexander Dobrovolskiy for his help in the lab and Emily Fenton for her thorough proofreading of this report.

Delft, University of Technology  
March 19, 2021

V.S. de Bruijne



---

# Chapter 1

---

## Introduction

This introduction will start by discussing the general background of adaptive optics (AO) which will be necessary to understand the relevance of this research. Section 1-2 then discusses the principles of deconvolution from wavefront sensing, which is a class of AO systems which will be developed further in this Thesis.

### 1-1 Adaptive Optics: Theoretical Background

When looking through a telescope, atmospheric turbulence interferes with the light coming from the source, leaving discrepancies between the image and the actual object. This Section will begin with discussing what the exact influence is of the turbulence on the light. Then, wavefront sensing will be discussed, which encompasses the retrieval of information about the degradation modes of the light, an essential part of AO.

#### 1-1-1 Atmospheric Degradation

The main source of image degradation in properly aligned telescopes is a result of interference of the light with the earth's atmospheric turbulence. Atmospheric turbulence is a product of the heating effects of the sun. Temperature gradients in the atmosphere bring about large scale motions in the air. This air eventually becomes turbulent and the turbulent eddies break up into progressively smaller scale motions, resulting in randomly sized pockets of air, each having a characteristic temperature [25]. The size of the turbulent eddies ranges from the inner scale  $l_0$  to the outer scale  $L_0$ . The refractive index of the air is related to its temperature, hence the pockets of air delay the incoming light in a random manner. After passing through a number of turbulence layers, light rays traveling in the same plane before entering the earth's atmosphere (i.e. a *flat* wavefront) are delayed relative to each other by the time they reach the ground.

An important parameter to quantify the severity of the atmospheric turbulence is the Fried parameter  $r_0$  [8].  $r_0$  is defined by the diameter of an area in which the root mean squared (RMS) wavefront aberration is equal  $1rad$ . Below  $1rad$ , the wavefront aberrations have an influence on the image that is negligible to human interpretation. The Fried parameter can be seen as the aperture size above which further increase in diameter does not result in an

increase in resolution [25]. This limitation can be overcome using AO.  $r_0$  generally ranges from less than 5cm in strong turbulence to over 20cm in good seeing conditions.

Because of its random nature, atmospheric turbulence can only be described statistically. In Kolmogorov's model for turbulence, the relative refractive index  $n(\mathbf{r})$  at point  $\mathbf{r}$  depends only on the distance  $|\rho|$  to point  $\mathbf{r} + \rho$ . For  $l_0 < |\rho| < L_0$ , the variance of relative refractive index is given by the index structure function  $D_N(\rho)$ :

$$D_N(\rho) = \langle |n(\mathbf{r}) - n(\mathbf{r} + \rho)|^2 \rangle = C_N^2 \rho^{\frac{2}{3}}. \quad (1-1)$$

Where the index structure coefficient  $C_N^2$  is assumed to only be depending on the height  $h$  above the ground [34].

### The wavefront and point spread function

The influence of the atmospheric turbulence of the light can be expressed in terms of the wavefront  $\phi$ . A wavefront is a surface perpendicular to the rays of the object, intersecting them at the points of equal phase. In free space, the wavefront is not obstructed by any medium, and the location of equal phase of all the rays in the wavefront are the exact same distance from the object, i.e. the wavefront is flat. The shape of the wavefront after the turbulence is given by

$$\phi(\mathbf{x}) = k \int_0^\infty n(\mathbf{x}, h) dh. \quad (1-2)$$

With the wave number  $k = \frac{2\pi}{\lambda}$  at wavelength  $\lambda$  and  $n(\mathbf{x}, h)$  the refractive index at altitude  $h$ .

There are two ways to numerically express the shape of the wavefront as a 2 dimensional plane. In the zonal approach, the wavefront is expressed as a matrix in which each entity represents the phase delay of the wavefront in a grid defined by the resolution of the matrix. This method makes it possible to express any shape of wavefront as long as it can be properly discretized, which makes it ideal for expressing the shape of turbulent wavefronts. With a modal approach, the wavefront is expressed by a number of 2 dimensional polynomials.

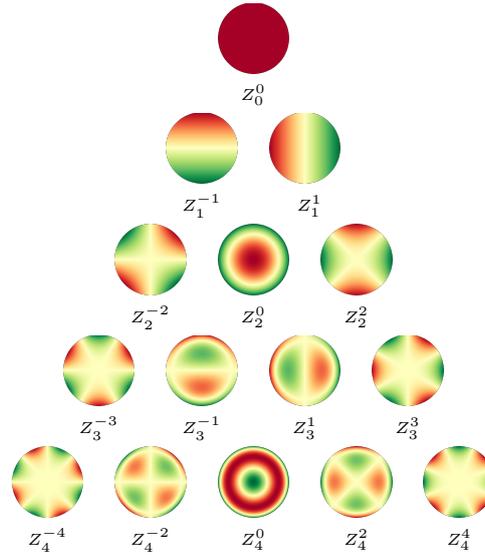
A commonly used set of polynomials are the Zernike polynomials (visualized in Figure 1-1) [38]. Zernike polynomials are based on common aberrations found in optical components, which makes them ideal for the expression of wavefront error introduced within the optical system, such as defocus or astigmatism. As there are an infinite number of orthogonal polynomials available, any continuous wavefront can be fully defined by a linear combination of Zernike polynomials. While this approach does provide a wavefront representation that is not bounded to a specific imaging resolution, the large amount of polynomials needed to properly express a turbulent wavefront makes this approach often impractical.

The influence of the wavefront on the observed image is expressed in terms of a point spread function (PSF). The PSF is a matrix that defines how each point in the object is spread out into the image.

The relation between the PSF ( $k$ ) and the wavefront ( $\phi$ ) is given by

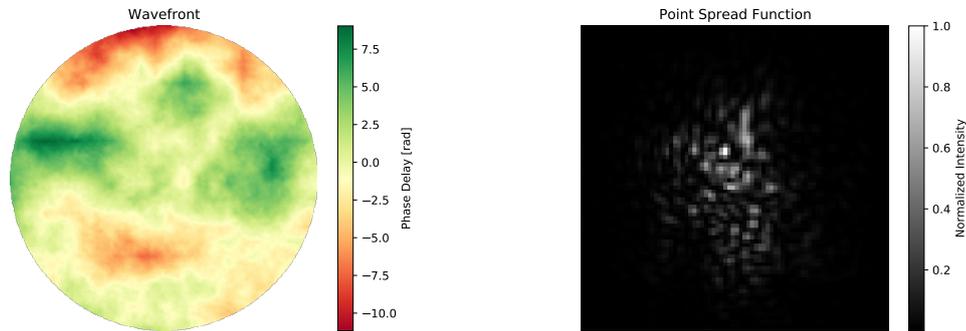
$$k(\mathbf{u}) \propto \left| \mathcal{F} \left\{ P(\mathbf{x}) e^{i\phi(\mathbf{x})} \right\} (\mathbf{u}) \right|^2. \quad (1-3)$$

Where  $P(\mathbf{x})$  represents the pupil function, consisting of ones inside and zeros outside the aperture. Figures 1-2 show an example of a turbulent wavefront with corresponding PSF.



**Figure 1-1:** The first 15 Zernike polynomials (after [10]). The color of the pixel represents its pixel value.

The PSF shown corresponds to a short exposure image.



**(a)** Turbulent wavefront with phase delay in *rad*. **(b)** Wavefront PSF from turbulent wavefront.

**Figure 1-2:** Example of a turbulent wavefront with a corresponding PSF,  $D/r_0 = 12$ .

Looking at Equation 1-3, one would expect the PSF corresponding to a perfectly flat wavefront (i.e.  $\phi = \vec{0}$ ) to be a delta function. In practice, diffraction effects limit the PSF to an 'Airy disk', which is a disk surrounded with rings on a fixed spacing. This effect is encompassed in Equation 1-3 by the pupil function  $P$ , which adds a padding around the wavefront of a size depending on the numerical aperture of the system. If the RMS wavefront aberration is below  $1\text{rad}$ , the diffraction effects are the dominant source of image degradation and the seeing is called 'diffraction limited'.

It is important to note that while the PSF is easily calculated given the wavefront, the wavefront can not easily be calculated back when a PSF is observed. The squared term in Equation 1-3 discards the information about the sign and complex values of the Fourier transform, thereby losing information about the original wavefront.

## Convolution

The PSF interferes with the object through a process called convolution. If the object is a point source (e.g. a star) the image is equal to the PSF. Extended objects (i.e. anything except point sources) can be seen as a collection of point sources, and the resulting image is equal to the superposition of the distorted image cast by each of the point sources. Convolution is denoted by

$$i = o * k + n. \quad (1-4)$$

Here,  $n$  is an additive noise component. The convolution operator reduces to an element-wise multiplication in the frequency domain:

$$I = OK + N. \quad (1-5)$$

Throughout this Thesis, the Fourier transform of a variable is denoted by the capital of the variable, e.g.  $\mathcal{F}\{i\} = I$ . Many calculations can be simplified by using the frequency domain counterparts of variables because of this property.

## Isoplanatism

Consider for now that the turbulence exists in one single layer in the atmosphere. Light from a star passes through a section of the turbulence with an area  $A$  before entering the telescope. The light from a neighboring star that is an arc distance of  $\theta$  away passes through a slightly different section of the turbulence. The wavefront of this second star will therefore be different than the wavefront of the initial star. Whether this difference is significant depends on  $\theta$ . Since aberrations of less than  $1rad$  can generally be ignored, the aberrations from the two stars can be considered equal if  $\theta < \theta_0$  with

$$\theta_0 \approx 0.31 \frac{r_0}{h}. \quad (1-6)$$

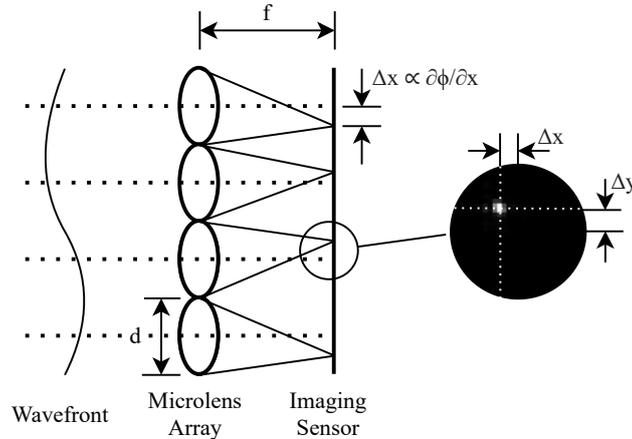
Here,  $h$  is the altitude of the turbulence (typically  $h \approx 5km$ ).  $\theta_0$  is called the isoplanatic angle. When the observed scene is within the isoplanatic angle, the assumption that the atmospheric interference is equivalent for all points in the observed scene is valid [34].

In practice, an AO system designed for isoplanatic conditions will assume that in Equation 1-4, the PSF corrupting the object is equal across the entire image. This assumption simplifies the image correction process significantly. In this Thesis, only isoplanatic observations are taken into account.

### 1-1-2 Wavefront Sensing

The wavefront sensor is the device in the AO system that helps to infer the shape of the wavefront. Several types of wavefront sensors exist, such as the pyramid wavefront sensor and the Plenoptic Sensor. The most commonly used sensor is the Shack-Hartmann (SH) wavefront sensor or SHWFS.

The SHWFS consists of two parts: a microlens array (MLA) and a charge coupled device (CCD) or imaging sensor. The MLA divides the light into  $N \times N$  separate sections which all cast an image on the CCD. These images are called the subaperture images. Figure 1-3 shows a schematic overview of a SHWFS.



**Figure 1-3:** Schematic overview of a cross-section of a SHWFS (after [37]).

Conventionally, the SHWFS works by dividing the wavefront up into sections of which the wavefront aberrations are fully divided by the first 3 Zernike modes. These wavefront modes do not influence the shape of the subaperture images, but only its  $x$  and  $y$  position from the 2<sup>nd</sup> and 3<sup>rd</sup> modes respectively. This simplification makes it possible to retrieve the wavefront shape without the use of complex phase retrieval methods, by measuring the  $x$  and  $y$  shifts of the subapertures and calculating them back to the actual wavefront. For this calculation, the found wavefront slopes can either be fit to a number of Zernike polynomials (modal wavefront reconstruction) or the wavefront slopes can be integrated into a  $2D$  surface (zonal wavefront reconstruction) [19].

Note that the Figures 1-4 and 1-3 both illustrate SH-patterns for point sources. In this case, the shift of the subaperture image can be found by calculating the centroid of the image. For extended scenes, the SH-pattern is equal to the point sources pattern convoluted with the object. Figure 1-5 shows an example of this. In this case, the shift of each of the subaperture images can be determined by the use of a correlation algorithm [15]. This type of algorithm selects one subaperture image as a reference image, and correlates it with the entire SH-pattern. The peak in the correlation function indicates a match between the reference image and other subaperture images. The shift of each subaperture image can be determined from the location of these peaks.

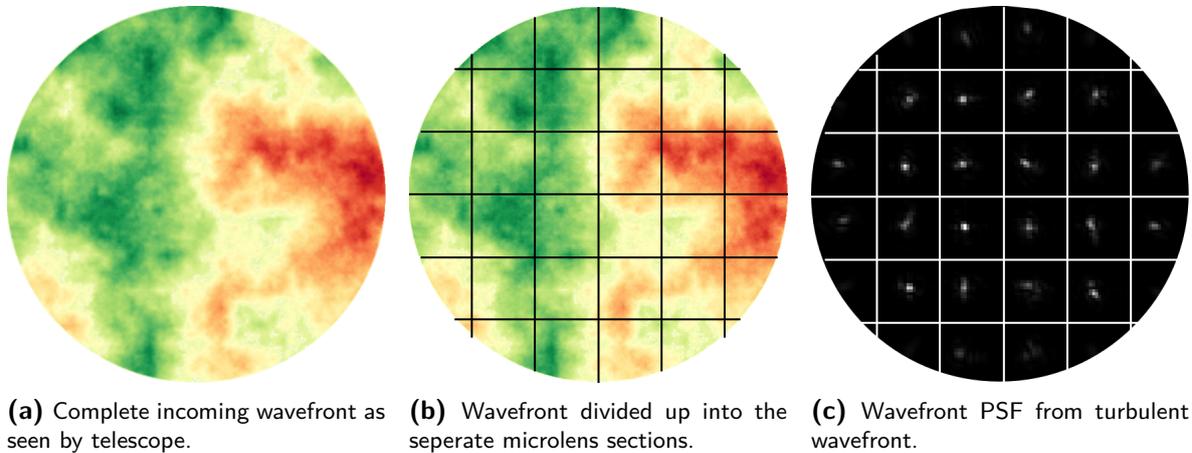
The performance of this algorithm can depend on the specific scene content. An observation of a checkerboard pattern, for example, would show distinct peaks in the correlation function on more places than the number of microlenses. It is important that the CCD's pixel count and the number of microlenses are coordinated in order to have a sufficient resolution for each subaperture image for the correlation function to work properly.

Using this method, a SHWFS with a  $N \times N$  MLA can be used to retrieve a wavefront for a telescope with a diameter of  $D \approx Nr_0$ . The diameter of the individual microlenses is not related to the diameter of the telescope, but only to the seeing conditions that the system is designed for. This means that the use of a bigger telescope that receives more light from a source is limited by the SHWFS, which needs to divide this light over more microlenses in order to still reconstruct the wavefront sufficiently.

A solution to this limitation has recently been proposed by using machine learning to reconstruct the wavefront, also called deep learning wavefront sensing (DLWS). It has been

shown by Nishizaki et al. [20], Hu et al. [11] and Bekendam [1] that artificial neural networks (ANNs) can be used to estimate the Zernike coefficients of a particular wavefront from a SH-pattern of a point source. Bekendam as well as Hu et al. [12] showed that ANNs can also be used to reconstruct the wavefront directly. Bekendam in particular focused on comparing the wavefront reconstruction capabilities of SH-patterns with different sizes of MLAs, which can be useful for low-light situations. The key in all these papers is the use of convolutional neural networks, which will be discussed in Chapter 3.

The ANN can be trained in order to reconstruct the wavefront not only based on the shifts of the subaperture images, but also on the shape of these images. Figures 1-4 show a wavefront overlaid with a grid that represents the MLA, as well as the PSFs from the individuals MLAs. While some of the PSFs are seemingly diffraction limited (meaning that the local wavefront is sufficiently represented by its tip and tilt modes), other microlenses have aberrated PSFs, indicating that higher order information is available.



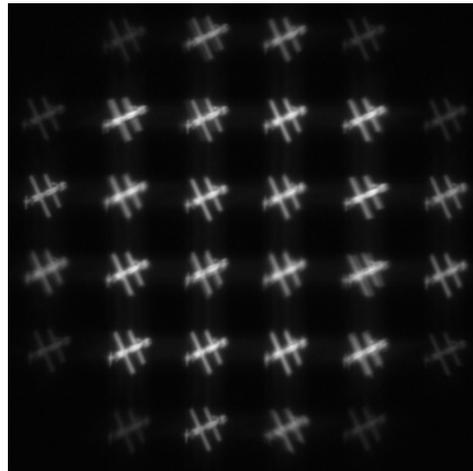
**Figure 1-4:** Resulting PSF's from each microlens. The wavefront aberrations are approximately  $D/r_0 = 19$ .

To the authors best knowledge, no machine learning approach to general extended scene SH-images has been proposed yet. The development of such an approach will be the focus of this Thesis.

## 1-2 Deconvolution From Wavefront Sensing

So far, this Chapter has discussed the origin and effect of atmospheric aberrations. Now, the task is to eliminate these aberrations in order to get a clear image of the object. A common way to do this is by the use of a deformable mirror (DM). The DM is an optical component that can induce specific wavefront aberrations based on a signal from a controller. A DM can be used to induce aberrations opposite to that of the atmosphere, thereby eliminating the distortions.

While the DM is a powerful component in the context of AO, there are a number of reasons not to use them. Firstly, DMs are expensive due to the complicated mechanics involved. Secondly, the control of a DM demands a large amount of computing power. Due to the rapidly changing nature of atmospheric turbulence, any delay between the readout of the SHWFS and the actuation of the DM has a direct influence on the quality of the wavefront



**Figure 1-5:** Example of an extended scene SH-pattern. Rather than shifted PSFs, the subaperture images now consist of the convolution of the subaperture PSFs with the object. If the subaperture diameters are lower than  $r_0$ , the subaperture images are shifted version of the object. If the subaperture diameters are larger than  $r_0$ , the subaperture images are differently distorted images of the object (as is the case in this image). No wavefront sensing method yet exists that takes the distortions of the individual subaperture images into account for the reconstruction of the wavefront. This Thesis will develop such method.

compensation. The system has to be able to operate in the order of  $10^2 Hz$  [35] and, depending on the DMs resolution, this requires a significant amount of computing power.

Image deconvolution, an alternative to the DM, does not have these drawbacks as it is software based and is less computationally demanding. Given the PSF (calculated from the wavefront) and image, one can make an estimate of the object by reversing the convolution operator. This method is called deconvolution from wavefront sensing (DFWS), and will be used in this Thesis.

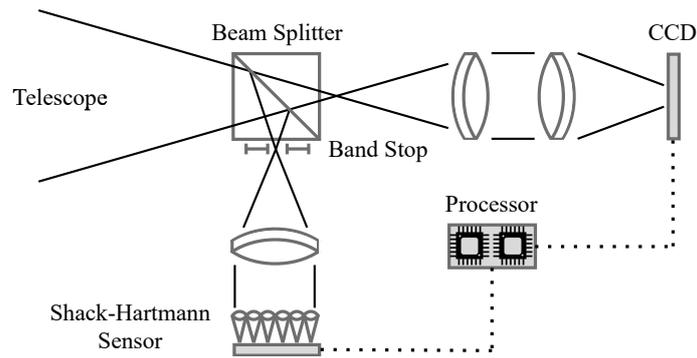
### State of the art deconvolution from wavefront sensing systems

Figure 1-6 shows a typical setup used in DFWS systems. The light from the telescope is split by the beam splitter into two paths. One path goes through two lenses and to the main CCD to form the image. The other path provides light to the SHWFS. The band stop is in place in order to prevent the subaperture images from overlapping.

The processor reconstructs the incoming wavefront based on the received SH-pattern. From this reconstruction, an estimate of the PSF of the main image is calculated, which is then used to deconvolute the main image.

DFWS was first proposed by Primot et al. [22] as a compromise between AO systems with a DM and speckle interferometry methods, which generally required more complex a posteriori computation and more images. Because DFWS combines several methods that are also found in other branches of AO and mathematics (most notably SHWFS interpretation and deconvolution), research not specifically focused on DFWS has greatly improved image reconstruction performance over the years.

Rimmele and Radick [24] showed that the use of correlation algorithms for determining the shift of the SH subaperture images make the use of DFWS very reliable on extended scenes. Jefferies and Hart [13] applied the frozen flow hypothesis (FFH) to DFWS. FFH



**Figure 1-6:** Optical setup for a deconvolution from wavefront sensing system.

uses the assumption that, on short time scales, the only significant movement in atmospheric turbulence is a linear regression [32]. This allows for the combination of adjacent SHWFS measurements in order to improve the resolution of the reconstructed wavefront. Mugnier et al. [18] introduced Myopic deconvolution, in which the object and wavefront are jointly estimated, contrary to earlier methods, in which the wavefront originating from the SHWFS is seen as fully determined. In this Thesis, improvement in wavefront sensing in the form of deep learning wavefront sensing will be applied to DFWS.

### Proposed system

This Thesis will explore the possibility of preprocessing the SH-pattern using blind deconvolution algorithms in order to open up DLWS to extended field observations. The goal is to implement this method in a real-time DFWS system.

DFWS is the ideal testing ground for this novel method of wavefront sensing because, unlike AO systems using a DM, the operation speed of the processor has no direct influence on the image reconstruction performance on the system. The reason for this is that in DFWS, the main image and the SH-pattern are captured simultaneously. A DM command can only be sent after the processing of the SH-pattern, meanwhile the wavefront can change. This method therefore requires less computational speed during early research without loss in performance. Nevertheless, the aim is to make a real-time image reconstruction system with a refresh rate in the order of  $10Hz$ .

In this proposed system, the processor will perform the following tasks:

1. Collect the image and the SH-image
2. Use blind deconvolution in order to reduce the SH-image to its point-source equivalent (i.e. the SH-pattern)
3. Use an artificial neural network to reconstruct the wavefront based on the SH-pattern
4. Calculate the PSF from the wavefront
5. Estimate the object by deconvoluting the image with the PSF

This Thesis will start by discussing methods for image deconvolution and deep learning wavefront sensing found in literature in Chapters 2 and 3, respectively. An essential part of DLWS is training data. This training data is generated from a software simulation of the optical system. The workings of the software simulation are explained in Chapter 4. Chapter

5 then discusses the adaptation and integration of the methods found in the literature review. Finally, Chapter 6 will elaborate on the performance of the novel system and compare it to conventional DFWS and AO systems.

Given the versatility of ANNs, the question can be asked why the ANNs are only applied to step 3. Indeed, Sánchez-Lasheras et al. [28] showed that it is possible to train an ANN in order to reconstruct the wavefront from an extended scene SH-image directly, thereby letting the ANN encompass both steps 2 and 3. Díaz-Baso and Asensio-Ramos [6] and Chen et al. [4] showed that ANNs can be used to estimate the object directly from the corrupted main image, thereby eliminating the need for a SHWFS and image deconvolution all together. The need for step 2 is, however, highlighted by the limited application of both these papers. Sánchez-Lasheras et al. [28] and Díaz-Baso and Asensio-Ramos [6] only use images of solar spots. The work of Chen et al. [4] is limited to the observation of man-made satellites. If the training data for the ANN and the application of the network are limited to similar objects, more direct ANN reconstruction approaches have been shown to work. If the ANN is used to interpret the extended scene SH-image directly, it would be nearly impossible to verify that the ANNs performance is independent of the scene content. In this Thesis, a system is developed that is meant to perform well independent of the scene content.

Additionally there are the constraints imposed by hardware limitations. The information of a point-source SH-pattern can be encompassed in a significantly lower resolution image compared to a extended scene SH-image. This reduction allows for the implementation of large-scale state of the art ANNs, which would not be possible at the original resolution on the hardware available for the research of this Thesis.

### Research Question

As a guideline for this Thesis, the following research questions are formulated:

*Can blind image deconvolution be applied in order to extend deep learning wavefront sensing methods to extended scene observations?*

*Can the proposed methods be used in the context of deconvolution from wavefront sensing for real time image reconstruction?*



# Image Deconvolution

In this Chapter, a number of deconvolution methods found in literature are reviewed. In the design of the DFWS system as discussed in the Introduction, the processor uses two different types of image deconvolution for two different tasks. Firstly, deconvolution is needed to convert the received image and the estimate of the PSF into an estimate of the object. This process is discussed in Section 2-1. Secondly, multi-frame blind image deconvolution (MFBD) is used to eliminate the dependency of the object on the SH-pattern. This process is called 'blind' because at this point both the object and PSF are unknown while the subaperture images provide multiple, differently aberrated images of the object. This process is discussed in Section 2-2.

The PSF interferes with the object through a convolution operator, denoted by  $*$ . The model describing the distortions induced by the PSF in an isoplanatic system is as follows:

$$i(x, y) = k(x, y) * o(x, y) + n(x, y). \quad (2-1)$$

Here,  $i(x, y)$  is the convoluted image as observed by the telescope,  $o(x, y)$  is the undistorted object,  $k(x, y)$  is the PSF and  $n(x, y)$  is noise that is present in the system (e.g. due to light pollution or sensor noise). The convolution of two matrices in the image domain is equivalent to the element-wise multiplication of the two matrices in the frequency domain:

$$I(u, v) = K(u, v)O(u, v) + N(u, v). \quad (2-2)$$

Here, the capitalized variable represents the 2D Fourier transform of the signal. For readability, the indices  $(x, y)$  and  $(u, v)$  are omitted throughout the remaining Chapter. It can be assumed that, unless mentioned otherwise, all the matrices represent two dimensional variables.

Image deconvolution mainly revolves around finding a solution to an ill-posed inverse problem of Equation 2-2. This type of problem is very common in mathematics and many methods have been proposed to solve them. However, due to the nature of ill-posed problems, there is not a unique solution and no single best method to find a solution.

## 2-1 Deconvolution

This Section will discuss two of deconvolution methods that have been found to provide a good trade-off between operation speed and object restoration, namely least-squares deconvolution and the Landweber method.

As the focus of this Thesis lies on wavefront reconstruction, the scope of the discussed deconvolution methods is limited. For an overview of state-of-the-art deconvolution methods, the reader is referred to Berisha and Nagy [2].

The method that is most popular in the DFWS literature is left out, namely Lucy-Richardson (LR) deconvolution [23] [16]. LR deconvolution is a powerful tool that is able to make accurate and low noise estimations of the object. The image domain implementation of the method, however, makes it resort to computationally expensive mathematical functions like the convolution and correlation operators. Even with modern adjustments attempting to speed the algorithm up (e.g. [3] [39] [17] [27]) it is unable to surpass the Landweber algorithm in terms of computational speed, while providing near identical image reconstruction performance.

### 2-1-1 Least-Squares Deconvolution

Looking at Equation 2-2, it is not immediately evident why restoring the original object can be a challenging task. It is straightforward to find a least-squares estimate of the object [7]

$$\tilde{O} = \arg \min_O \|I - KO\|_2^2 = \frac{\tilde{K}^* I}{|\tilde{K}|^2}. \quad (2-3)$$

Here, the superscript \* denotes the complex conjugate. The accuracy of the least-squares method depends on the particular PSF. In many cases, the PSF matrix is ill-conditioned, meaning that it has a number of singular values close or equal to zero. The low singular values often correspond to high-frequency components. Dividing the noise by these singular values results in an amplification of high frequency noise [2].

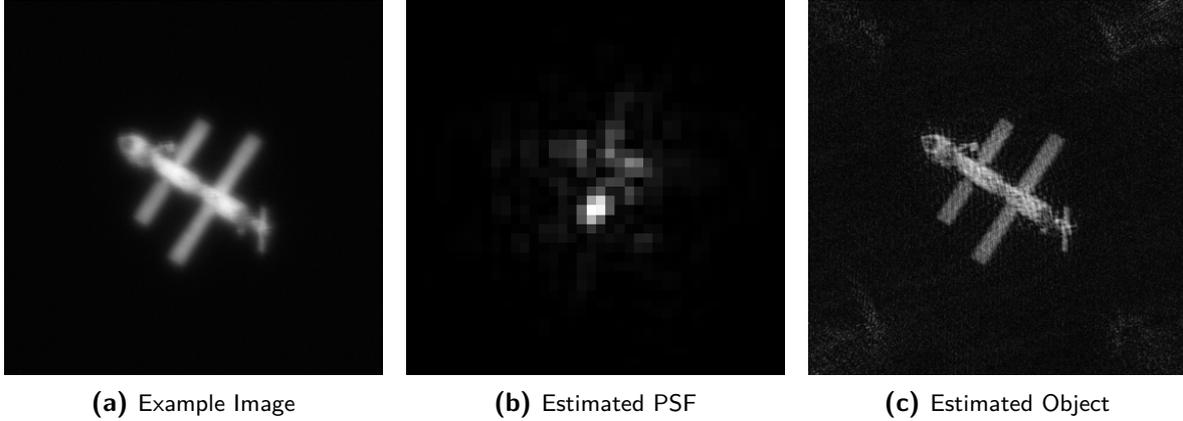
An easy way to suppress the effect of the small singular values of  $K$ , is to add a small regularization constant  $\epsilon$  to the least-squares solution to ensure that the estimate  $\tilde{O}$  remains finite:

$$\tilde{O} = \frac{\sum_{i=1}^m \tilde{K}^{i*} I^i}{\sum_{i=1}^m |\tilde{K}^i|^2 + \epsilon}. \quad (2-4)$$

A special case of Equation 2-4 is the Wiener filter. In the Wiener filter, the parameter  $\epsilon = N/S$  is set to the noise-to-signal ratio [5]. Here,  $S$  and  $N$  are the power spectral densities of the object and noise, respectively. This Wiener filter minimizes the mean squared error of the solution. In practice, however, the power spectral densities of the object and noise are not known, but need to be estimated from the image data, prior knowledge of the imaging setup and seeing conditions. Other regularization methods (e.g. Tikhonov regularization or Total variation) provide other guidelines for choosing the regularization parameter, but require information about the image and PSF that is not always known, leaving the user to estimate certain image parameters.

Figure 2-1 shows an example of the performance of the least-squares deconvolution. It can be seen that the ill-posedness of the PSF introduces periodic high-frequency artifacts in

the image.



**Figure 2-1:** Simulated example of the performance of a least-squares deconvolution method with a regularization parameter of 0.1. It can be seen that periodic artifacts are introduced by the deconvolution method as a result of the ill-posedness of the PSF. Note that the PSF is not the actual PSF but estimated using the methods discussed in Chapter 3. Due to the errors in the PSF, a perfect reconstruction is not possible.

Figure 2-1c shows that with least-squared deconvolution, high frequency periodic artifacts can show up in the estimated object. There are ways to remove these artifacts by filtering the frequency domain image (e.g. Varghese et al. [33]), which work by identifying peaks in the frequency domain image and removing them. These methods are, however, not time efficient and do not have the ability to distinguish noise from scene content, which can result in the removal of periodic scene content.

## 2-1-2 Landweber Deconvolution

Iterative deconvolution methods generally handle ill-conditioned PSFs better, and are well suited for image deconvolution. Unfortunately, the iterative nature of these methods has an impact on their calculation speed. Frequency domain iterative image deconvolution methods take the general shape of [2]

$$\tilde{O}^{n+1} = \tilde{O}^n + \tau K^T (I - K^* \tilde{O}^n). \quad (2-5)$$

Here  $\tilde{O}^n$  is the estimate of the object at iteration step  $n$ . In the iteration, the estimated object is added to the residual error  $(I - K^* \tilde{O}^n)$  and multiplied by  $\tau K^T$ . For stability of the iteration, it must satisfy that

$$0 < \tau < \frac{2}{\sigma_{\max}^2}. \quad (2-6)$$

There are several methods to choose the exact value of  $\tau$ , for example:

$$\begin{aligned} \text{Steepest Descent:} \quad \tau &= \frac{\|K^T(I - K\tilde{O}^n)\|_2^2}{\|KK^T(I - K\tilde{O}^n)\|_2^2} \\ \text{Landweber Iteration:} \quad \tau &= \frac{1}{\sigma_{\max}^2} \\ \text{Quasi Newton:} \quad \tau &= -(\tilde{P}'_n)^{-1} \end{aligned}$$

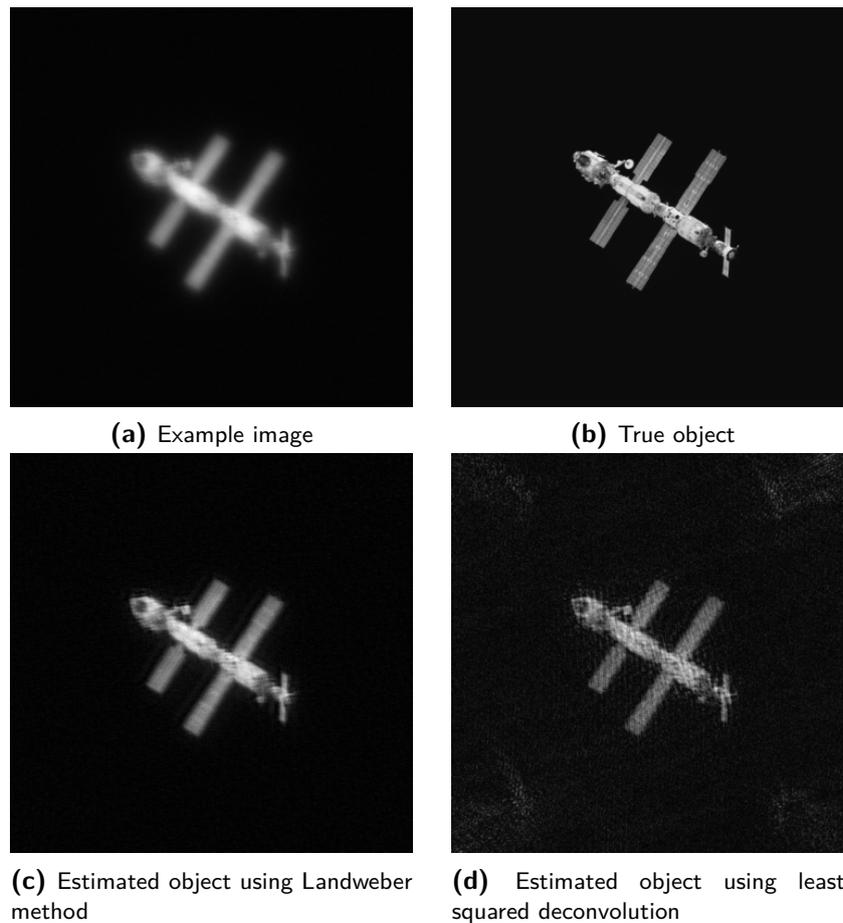
With  $\sigma_{\max}$  being the highest singular value and  $\hat{P}'_n$  an approximate Jacobian of  $K$ . A proper choice of  $\tau$  has the potential to decrease the number of iterations needed for convergence. This does, however, not directly relate to an increase in computational speed because some calculations of  $\tau$  make use of expensive math operations and need to be repeated at every iteration.

The performance of the iteration in terms of an image quality metric (e.g. mean squared error to ground truth) is not necessarily converging for higher numbers of iteration steps. Moreover, the relation between the image quality metric and the number of iterations is not necessarily convex. It is possible to let the amount of iterations depend on a stopping criterion, for example, the iteration ends if the following stopping criterion is met:

$$\|I - K\tilde{O}^n\|_2 \leq \|I - K\tilde{O}^{n-1}\|_2. \quad (2-7)$$

The selection of stopping criterion depends on the application and can be tuned to fit the preference of the user. However, due to the non-convexity of the iteration, it cannot be ascertained that the best image is obtained when the stopping criterion is met.

Figure 2-2 shows a comparison of the performance of the Landweber method and the least-squares deconvolution. Indeed, the iterative method handles the ill-posed PSF better and does not show periodic artifacts in the estimated object.



**Figure 2-2:** Simulated performance of the deconvolution methods as discussed in this Chapter, compared to the true object. Note that the PSF used for the deconvolution is not true to the PSF used for the convolution, hence no perfect reconstruction is to be expected. The Landweber method performs better than the filtered least-squared deconvolution, but also takes longer to compute.

## 2-2 Blind Deconvolution

In blind-deconvolution, there is no access to the PSF. Instead, the PSF needs to be estimated alongside the object. To do this effectively, additional information about the object is often used. In phase diversity for example, a second image is captured with an added known aberration, which is used to reconstruct the object [21]. Another method is to use multiple images with different unknown aberrations. This is called multiframe blind deconvolution (MFBD). These multiple images can be taken at different times, or, as in this case, simultaneously using a SH-sensor.

The method for MFBD used in this Thesis is Tangential Iterative Projections (TIP) as introduced by Wilding et al. [36]. The benefit of using TIP rather than other MFBD methods is its lack of need for prior information and its unique framework that allows it to be tuned for a large variety of applications.

### 2-2-1 Tangential Iterative Projections

Tangential Iterative Projections (TIP) was first introduced by Wilding et al. [36]. TIP is a multiframe blind deconvolution (MFBD) algorithm, meaning that no PSF is available and both the object and PSF must be estimated from a collection of images. In the case of a SHWFS, the collection of images is contained within one single image.

TIP works by alternating between estimating the PSF and the object from the image and the latest estimates of the PSF and object. Given the latest estimate of the SH-PSF ( $\hat{K}$ ) and the SH-image ( $I$ ), an estimate of the object ( $\hat{O}$ ) can be calculated as follows:

$$\hat{O} = \mathcal{P}_O \arg \min_{O \in \mathbb{C}^{M \times M}} \|I - O\hat{K}\|^2. \quad (2-8)$$

Given the updated estimation of the object, the estimate of the SH-PSF follows from:

$$\hat{K} = \mathcal{P}_K \frac{I}{\hat{O}}. \quad (2-9)$$

Here, the projection operators  $\mathcal{P}_X$  project the result of the least-squares deconvolution onto a set of valid objects or PSFs  $X$ . In general, this valid set consists of positive and normalized images. Additional constraints can be added to this set in order to guide the convergence of the algorithm.

By iterating between Equation 2-8 and 2-9, the TIP algorithm will converge to an estimate of the object and PSF that falls within the forced constraints. There is, however, no promise that the estimated object and PSF are equal to the true object and PSF.

# Deep Learning Wavefront Sensing

In this Chapter, state of the art deep learning wavefront sensing (DLWS) methods found in literature are discussed. All the DLWS methods discussed make use of artificial neural networks (ANN) to interpret SH-PSFs. In order to properly convey the difference between the methods, it is necessary to explain the fundamentals of ANNs, which will be done in Section 3-1. After that, different ANNs used for DLWS will be discussed and compared in Section 3-2.

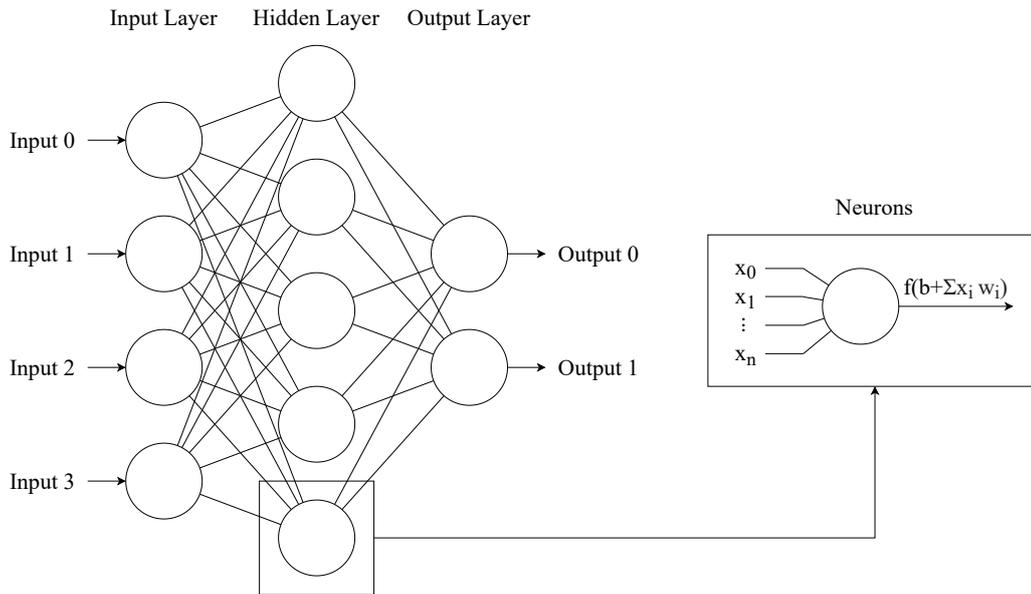
### 3-1 Artificial Neural Networks: Theoretical Background

Artificial neural networks (ANNs) are programmed functions that have an architecture loosely based on biological neural networks. They utilize a network of neurons that individually only use simple calculations, but when combined can form much larger and complex functions. ANNs are not programmed in the same way conventional functions are programmed by following a set of fixed instructions, but are instead trained. The ANN is fed a large amount of input data and desired output data and the training sequence tweaks the parameters of each neuron in order to make the ANN converge to the desired function.

Figure 3-1 shows an example of a neuron within an ANN. The inputs of the neuron are denoted by  $x_i$ , with  $i \leq n$ . The total amount of inputs  $n$  depends on the architecture of the ANN. Within the neuron the inputs are weighted and summed together with a bias  $b$ . The output of the neuron is  $f(b + \sum_{i=0}^n x_i w_i)$ , where  $w_i$  is the weight corresponding to input  $i$  and  $f$  is the activation function. The activation function is a predefined function that can be used to introduce non-linearity in the ANN. Most often in modern ANNs, the rectified linear unit (ReLU) is used as the activation function, defined by

$$f(x) = \begin{cases} x, & \text{if } x \geq 0. \\ 0, & \text{if } x < 0. \end{cases} \quad (3-1)$$

Figure 3-1 also shows an example of how neurons can be combined in order to form an ANN. This particular network can be used to map a 4 dimensional vector to a 2 dimensional vector, with 5 neurons in the hidden layer. This amount of neurons results in 11 biases and 34 weights to be estimated. The weights and biases are estimated using a process called training. In the training process, the network is fed a batch of training data and the outputs



**Figure 3-1:** An example of an ANN that maps a 4 dimensional input to a 2 dimensional output using a single hidden layer with 5 neurons. On the right, the basic process of the neuron is visualized, with  $x_i$  and  $w_i$  the neurons inputs with corresponding weights,  $b$  the neurons bias and  $f$  the activation function.

are calculated. The output is compared to the desired output by means of a loss function  $L$ , which can be, for example, the mean squared error. After that, the contribution of each of the weights on the value of  $L$  is calculated, which finally is used in a gradient descent optimization.

This process is repeated until each data point has passed through the ANN, which is called an epoch. Multiple epochs can be performed in order to improve the prediction accuracy of the network. The prediction accuracy of the ANN is expressed in terms of two variables: the training loss and the validation loss. The training loss is the loss value for data that has been used in the training process. The validation loss is the evaluated loss function on the data that the network has not seen in the training process. The difference between the training and validation loss indicates the extent to which the ANN is able to generalize the patterns between the input and output data or the extent to which the ANN has simply remembered the output data corresponding to each input set.

Even though the training of the ANN is fully automated, the process of making a well functioning ANN is not as straightforward as providing the training data to the code. Only the weights and biases are calculated automatically. The amount of layers, amount of neurons in a layer, the choice of activation function, choice of loss function and many more (not discussed) parameters still need to be chosen manually depending on the exact application. The combination of these parameters is referred to as the network's architecture. Often, ANNs used for similar applications by others are a good starting point for the exact selection of the network's architecture.

### 3-1-1 Convolutional neural networks

ANNs, as discussed, are powerful tools for many applications, but have some limitations. Most significantly for imaging applications, it only accepts one dimensional input and output vectors. Reshaping images into 1D vectors makes it hard for ANNs to compare adjacent pixels and find repeating features in the image. A type of network that is more effective for imaging applications is the convolutional neural network (CNN). In a CNN, the neurons do not consist of weights and biases, but instead of a convolution kernel and a bias. These kernels are  $N \times M$  matrices (usually  $3 \times 3$ ) that are convoluted with the input image to form the neuron's output. The pixel values of the kernels are trained in order to extract specific features from the input image.

As an example, imagine training a CNN to recognize basic shapes, e.g. squares, circles and stars. Figure 3-2 illustrates the CNN with a star shape as input. In the first convolutional layer, the kernels could be trained to recognize very basic features like horizontal lines. In the Figure, the top image is the activation pattern of this convolutional kernel. The second image is activation pattern of a convolution kernel that recognizes vertical lines, the third recognizes a specific orientation of corner and the last one diagonal lines. An examples of a convolution kernel that is activated by horizontal lines is

$$c = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3-2)$$

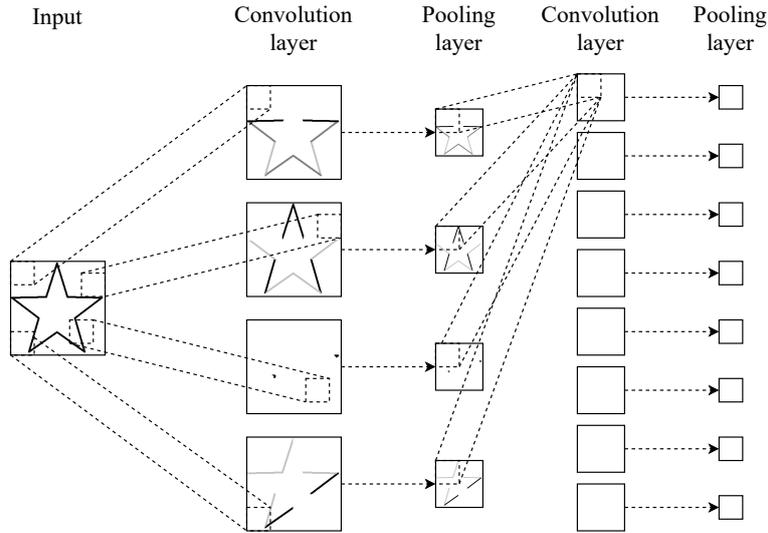
The second convolutional layer now has access to the location of different specific features of the shape and can be trained to recognize, for example, specific relations between horizontal and vertical lines.

In between the convolution layers, there are pooling layers. The most common pooling layer is the max pool, which reduces the resolution of the image by selecting the maximum values over a certain area. This reduction in size is not only beneficial to keep the size of the neural network low, but also allows the deeper convolution layers to find patterns in the data that span across a larger area in the original picture without having a larger convolution kernel.

The amount of convolution layers and the amount of kernels per layer that are necessary depend on the application. In the example where the CNN needs to identify basic shapes, 3 layers might be enough. If the CNN is used to identify, for example, different breeds of dogs in an image, deeper layers are necessary to identify specific features like type of fur.

## 3-2 State of the art deep learning wavefront sensing

ANNs and CNNs have been shown to be useful for the interpretation of SH-PSFs in many different ways. Earlier research showed that ANNs can be used to accurately reconstruct the wavefront from SH slope measurements, either by using Zernike polynomials [9] or by using a CNN to reconstruct the wavefront directly [31]. Suárez Gómez et al. [30] showed that CNNs can be used to reconstruct wavefront slopes from the SH-PSFs, which could then be used to reconstruct the wavefront. Recently, it was shown that CNNs could be used not only for wavefront slope sensing and reconstruction separately, but that one CNN could be used to fulfill both functions [11] [12] [1]. The simultaneous research done by both Hu et al. [11] and

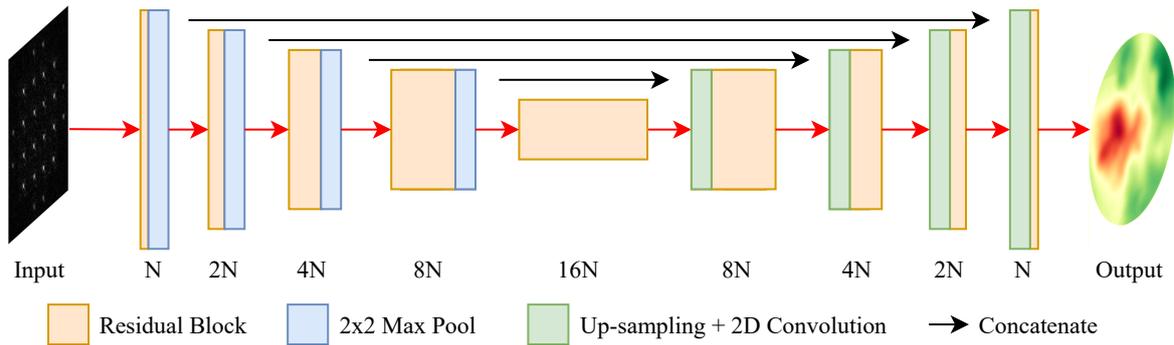


**Figure 3-2:** Example of a CNN with two convolution and pooling layers. Note that the output structure of the network is not shown. It is visualized in the first layer how different convolution kernels can be trained in order to identify different features within the image. Deeper layers can recognize more complex features from the combinations of outputs in the previous layers.

Bekendam [1] came to very similar conclusions about the optimal CNN architectures to be used. In these papers, 2 basic CNN architectures were discussed: *AlexNet* for the estimation of Zernike coefficients and *U-net* for direct wavefront sensing. In this Thesis, the CNNs are trained in order to reconstruct turbulent wavefronts, which are insufficiently approximated by Zernike polynomials. Hence, only the *U-net* based architectures are discussed.

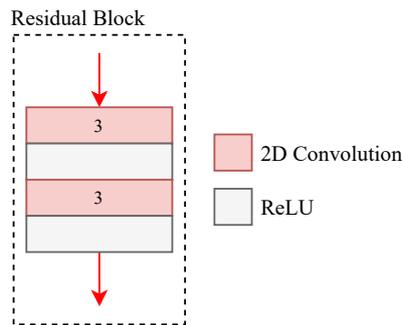
### 3-2-1 U-net

*U-net* was originally developed for image segmentation of biomedical images by Ronneberger et al. [26]. In wavefront sensing, similarly to image segmentation, information does not 'move' within the image, i.e. information of the wavefront on the location  $(x, y)$  can be found close to  $(x, y)$  in the SH-pattern. Hence *U-net* is a very suitable network architecture for the purpose of wavefront reconstruction. This property would not be fulfilled if, for example, the CNN is trained in order to reconstruct the main sensors PSF from the SH-PSF, even though this would be equally suitable for a DFWS application. The fundamental *U-net* architecture is shown in Figure 3-3. *U-nets* unique feature is the concatenation of data in different levels of the network, which is used to preserve data that would otherwise be filtered out by the convolution layers.



**Figure 3-3:** A visualization of the general *U-net* architecture. The black arrow indicates a concatenation operation, which is used in order to preserve information from earlier layers and greatly improves the performance of the network. Different implementation of the network use different residual blocks.

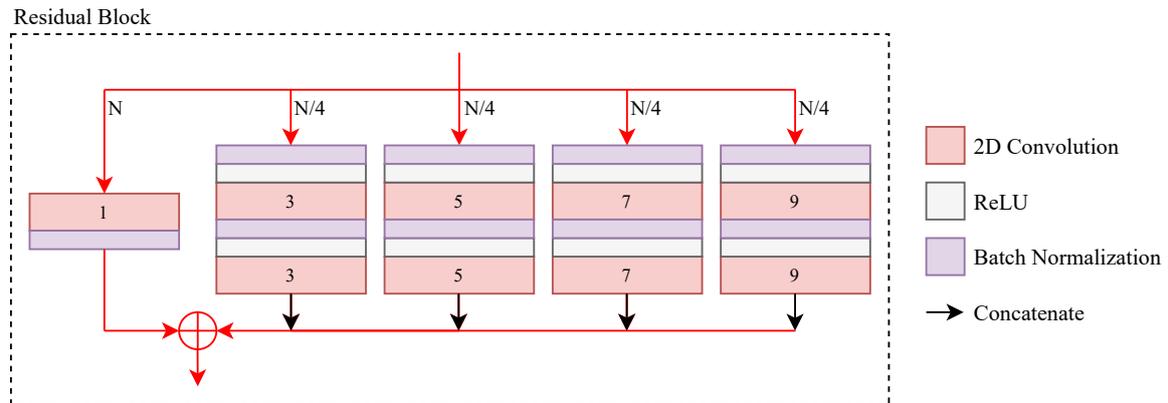
Hu et al. [12] and Bekendam [1] both based their architecture on *U-net* (shown in Figure 3-3), but adapted it in different ways. Bekendam used the residual block as seen in Figure 3-4 and  $N = 8$ . The residual block consists of two convolution layers with kernels of size  $3 \times 3$  followed by a ReLU activation.



**Figure 3-4:** The residual block used in combination with the architecture as shown in Figure 3-3 as used by [1].

The network architecture proposed by Hu et al. uses a significantly larger network with  $N = 32$  and a residual block as shown in Figure 3-5. This architecture has two notable features. Firstly, kernels of different sizes ranging from  $3 \times 3$  to  $9 \times 9$  are used, which allows the network to estimate features that span across multiple pixels in earlier layers. Secondly, a  $1 \times 1$  convolution layer is used in parallel and summed to the concatenation of the other layers. A  $1 \times 1$  convolution layer is not able to recognize any features but is used to preserve information that may not be extracted by the other layers. Within this residual block, there are  $1328N$  parameters to be estimated, compared to  $18N$  in the residual block used by Bekendam [1]. This combined with an increase of  $N$  by a factor of 4 makes this architecture slower to train and evaluate, but also makes it potentially better at recognizing features within the SH-PSF.

It must be noted that Bekendam proposed another architecture with similar features to the *U-net* implementation of Hu et al. that was trained to estimate Zernike coefficients from a SH-PSF. This architecture is not discussed in this Thesis because of the limitations imposed



**Figure 3-5:** The residual block used in combination with the architecture as shown in Figure 3-3 as used by Hu et al.. The different sizes of kernel allow the network to recognize features spanning across multiple pixels in earlier layers, but comes at the cost of an increase in parameters to be trained. The additional single pixels convolution helps preserve information that is not extracted by the other convolution layers.

by the modal wavefront representation. Bekendam did note that the  $1 \times 1$  convolution layer bypass may decrease the CNN's ability to filter out noise.

# Numerical Simulations of Optics

In this Chapter, the numerical simulation of the optical system developed as part of this Thesis is discussed. The simulation environment is a crucial aspect of this Thesis because it will be used to generate training data for the deep learning approach to wavefront sensing. The simulation must be very accurate and time efficient. Discrepancies between the simulation and the actual optical setup will result in poor real world performance of the developed methods, because neural networks used for deep learning wavefront sensing which are trained using data from the simulation must be able to find the exact same patterns in the data from the optical setup. The need for a fast simulation comes from a practical point of view. The amount of data points needed for the training of the neural networks is in the order of  $10^5$ , which means that every second in simulation time will result in several hours in the process of generating the training data.

This Chapter will discuss the numerical simulation in 3 separate parts: Firstly, Section 4-1 examines the generation of realistic turbulent wavefronts. Secondly, 4-2 explains the propagation of the wavefront into the SHWFS. Lastly, 4-3 illustrates shortly how the wavefront influences the image of the main imaging sensor.

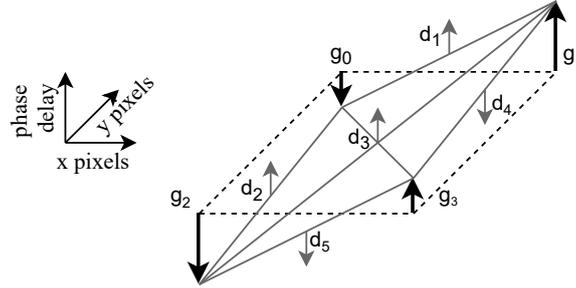
## 4-1 Wavefront Simulation

In the simulation environment, turbulent wavefronts are generated as described by Lane et al. [14]. A number of assumptions about the phase screen are made. The atmospheric turbulence follows a Kolmogorov spectrum, its phase is statistically uniform over the interval  $[-\pi, \pi]$  and it can be approximated as a single phase screen located at the entrance pupil. The Kolmogorov phase screens are self-similar, meaning that they appear similar regardless the scale they are looked at. On smaller scales, only the magnitude of the phase changes, not the fluctuation. Because of this property, a turbulent wavefront can be generated independent on  $D$  or  $r_0$  and scaled by  $(D/r_0)^{\frac{5}{6}}$ .

The phase screen is generated by initializing 4 points on a square grid, with a random value and variance equal to  $\sigma^2$  (see Equation 1-1). The resolution of the phase screen is increased by filling in the midpoint between existing data points by linear interpolation with the addition of a random variable. The value of the random variable  $d_1$  is chosen such that

$$\left\langle \left( g_1 - \frac{g_1 + g_2}{2} - d_1 \right)^2 \right\rangle = \sigma^2. \quad (4-1)$$

Here,  $g_1$  and  $g_2$  are the left and right data point, respectively. This process is continued until a sufficient resolution is reached. Figure 4-1 shows a visualization of this process. Note that this method can only create square phase screens with a resolution equal  $2^n + 1$ , with  $n \in \mathbb{N}$ . Hence, if a circular phase screen with a diameter of 680 pixels is needed (as used in this Thesis), a phase screen with a size of  $1025 \times 1025$  is generated and then cropped to the desired shape.



**Figure 4-1:** A visualization of the generation of a turbulent phase screen as described by Lane et al. [14]. 4 points are randomly generated on a grid ( $g_i$ ). New data points ( $d_i$ ) are generated in between the existing ones by interpolating between the data points and adding a random value.

For this Thesis, an efficient way to up-sample the turbulent phase screen was developed, which performs the interpolation using a dot product between the low resolution wavefront and an interpolation matrix. In Figure 4-1 the black arrows represent the low resolution wavefront and the collection of gray and black arrows represent the interpolated wavefront. The relation between the two can be defined by

$$\begin{bmatrix} g_1 \\ d_1 \\ g_2 \\ d_2 \\ d_3 \\ d_4 \\ g_3 \\ d_5 \\ g_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1/2 & 0 & 1/2 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/2 & 0 & 1/2 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} + \begin{bmatrix} 0 \\ r_1 \\ 0 \\ r_2 \\ r_3 \\ r_4 \\ 0 \\ r_5 \\ 0 \end{bmatrix}. \quad (4-2)$$

Here  $r_1$  to  $r_5$  are random values that adhere to Equation 4-1. The interpolation matrix depends on the size of the wavefront, but can be calculated in advance for each step in resolution and imported from a file. Since the interpolation matrix is a sparse matrix, the dot product can be calculated reasonably quickly.

At larger resolutions, the amount of memory required for the calculation of the dot product is in the order of Terabytes. To avoid this, the size of the interpolation matrix can be capped (e.g. at a resolution of  $65^2 \times 33^2$  pixels depending on the available memory) and the up-sampling can continue in blocks. For example, if the wavefront is interpolated from

513 to 1025 pixels, the wavefront is divided into 16 blocks of size 33 which are individually up-sampled to 65. Putting these blocks together then forms the wavefront of a  $1025 \times 1025$  resolution. As the top row and left column of the blocks overlap with the adjacent blocks, the continuity of the turbulent wavefront is maintained.

The generation of a phase screen of aforementioned resolution takes roughly  $0.5\text{sec}$  per phase screen. In order to increase computational speed, it is chosen to not generate a new phase screen for every data set, but instead to save the  $1025 \times 1025$  phase screen and crop 10 phase screens out of it at random locations. The cropped phase screens are then also randomly mirrored and rotated.

Other literature which looks into deep learning wavefront sensing used wavefronts generated from Zernike polynomials instead of Kolmogorov statistics (e.g. [1] [11] [12] [20]). Depending on the application of the imaging system, phase screens generated from Zernike polynomials can be a valid approximation of random phase screens. In the context of approximating turbulent wavefronts, however, this approach tends to ignore the high order (pixel-to-pixel) variations in the phase. It does not take the self-similarity of the turbulent phase into account. Wavefront sensing methods tested on phase screens generated from Zernike polynomials tend to show better performance than when tested on Kolmogorov phase screens as they generally lack the ability to retrieve these higher order wavefront fluctuations.

## 4-2 Point Spread Function Simulation

In the simulation, two PSFs must be constructed from the wavefront: the SH-pattern (Section 4-2-1) and the PSF of the main sensor (Section 4-2-2).

### 4-2-1 Shack-Hartmann Pattern Simulation

The simulation of the SHWFS is done in accordance to Soloviev et al. [29]. Recall the equation that relates the wavefront phase  $\phi$  to the PSF:

$$k(\mathbf{u}) \propto \left| \mathcal{F} \left\{ P(\mathbf{x}) e^{i\phi(\mathbf{x})} \right\} (\mathbf{u}) \right|^2. \quad (4-3)$$

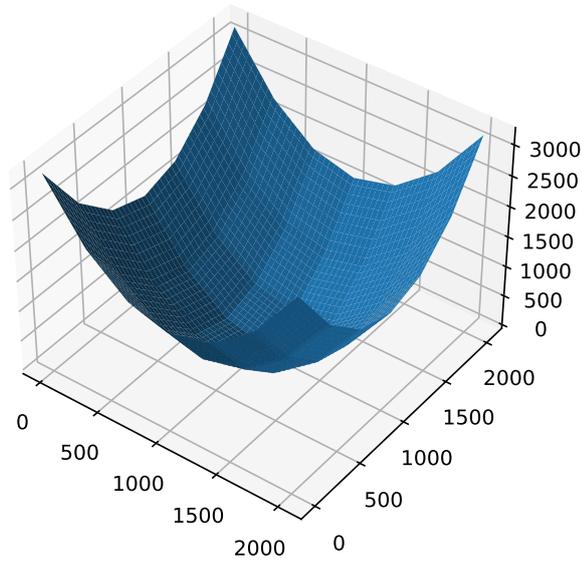
The addition of the MLA makes this formula not directly applicable to SHWFS. The influence of the MLA can be reproduced by dividing the wavefront phase into a grid representing the MLA (see Figure 1-4), calculating the PSF of each individual grid point and shifting the resulting PSFs in accordance with the location of the cell. This process can, however, be computationally expensive, especially with an high number of microlenses.

As shown by Soloviev et al., the influence of the MLA can also be modeled by adding the influence of the MLA on the wavefront phase to the earlier generated phase screen, denoted by  $\phi_{\text{MLA}}$ . In order to not add additional distortions to the turbulent phase  $\phi$ ,  $\phi_{\text{MLA}}$  has to be limited to tip and tilt modes within the grid cells as seen in Figure 1-4. The tip and tilt modes need to be chosen such that the subaperture PSFs end up in the exact right location. A piece-wise linear defocused wavefront phase achieves both these criteria. The shape of  $\phi_{\text{MLA}}$  can be seen in Figure 4-2.

In the software implementation developed in this Thesis, the piece-wise linear defocused wavefront was constructed by first sampling a continuous defocused wavefront on a coarse grid of  $(N + 1)^2$  points, where  $N$  is the number of subapertures in the MLA, using

$$\phi_{\text{defocus}} = a \left( 2 \left( (x - \bar{x})^2 + (y - \bar{y})^2 \right) - 1 \right). \quad (4-4)$$

Here,  $\bar{x}$  and  $\bar{y}$  represent the value at the center of the  $x$  and  $y$  axis, respectively. The values within this grid can now be found through a 2 dimensional linear interpolation of the course defocus. The exact strength of the defocus ( $a$ ), represents a scaling along the  $z$  axis and must be chosen such that the subaperture spacing appears to correspond to the SH-patterns found from the optical setup. Figure 4-2 shows a visualization of the piece-wise linear defocused wavefront.



**Figure 4-2:** Visualization of  $\phi_{\text{MLA}}$ , which consists of a piece-wise linear defocus (after [29]). The  $x$  and  $y$  axis represent the pixels of the wavefront matrix and the  $z$  axis represents the phase delay [rad], and needs to be scaled in order to achieve the right subaperture spacing.

From the Nyquist-Shannon principle, it follows that the maximum wavelength that the CCD can sample without aliasing is  $2s$ , where  $s = 5.2\mu\text{m}$  is the pixel spacing of the CCD. This corresponds to a maximum angular spectrum  $\mathbf{k}$  of

$$|\mathbf{k}| \leq \frac{\pi}{s}. \quad (4-5)$$

The angular spectrum is also limited by the aperture of the MLA by

$$|\mathbf{k}| \leq \text{NA} \frac{2\pi}{\lambda}. \quad (4-6)$$

For the simulation of the SHWFS, the resolution should be high enough such that it holds that

$$\text{NA} \frac{2\pi}{\lambda} \leq \frac{\pi}{s}. \quad (4-7)$$

In the test setup, an square aperture of  $3mm$  is used with a light source of  $\lambda = 470nm$ . It follows that  $NA \leq \frac{\lambda}{2s} \approx 0.043$ , corresponding to an aperture of a maximum of  $1mm$ . Hence, the resolution of the simulation should be 3 times larger than the effective resolution of the CCD ( $680 \times 3 = 2040$ ). After the SH-pattern has been generated it can be reduced to the desired resolution by averaging.

#### 4-2-2 Main Point Spread Function Simulation

The PSF of the main imaging sensor is necessary as input for the image deconvolution step. Discrepancies between the simulated PSF and the true PSF of the optical setup will not become apparent when testing the performance of the system within the simulation environment, but will decrease the performance only when the system is applied on the true optical system. There is, however, a trade-off to be made between the speed and accuracy of the simulation. Following the procedure of selecting the resolution of the simulated SH-PSF, the simulated resolution of the main CCD must also be in the order of  $10^3$  pixels. This high number of pixels, however, comes at the cost of more memory and computational time needed for the Fourier transform, which is unfavorable for real-time operation. Moreover, the errors that will be introduced in the wavefront reconstruction step have a much larger influence in the discrepancies between the simulated and actual PSF than the errors resulting from inadequate simulation.

For these reasons, it is chosen to simulate the PSF using wavefront screen of  $128 \times 128$  pixels (equal to the output of the deep learning wavefront sensing step). The size of the pupil function is then chosen such that the resulting size (in pixels) of the PSF corresponds to the size found in the actual optical system.

### 4-3 Imaging Simulation

Now that the PSFs of both sensors have been constructed from the wavefront, the image can be made through a convolution of an object and the PSF. Since the resolution and magnification of the subapertures and the main CCD are different, the resolution of the object needs to be adjusted. The object used for the main image is loaded from a file, and the object used for the SH-image is a down-sampled version of the same object.

The images can now be found through a convolution of the object and PSF

$$\begin{aligned} i &= o * k \\ i_{sh} &= o * k_{sh}. \end{aligned} \tag{4-8}$$

Convolution in the image domain is a computationally expensive operation. A faster alternative is to take the Fourier transform of  $o$  and  $k$  and to multiply them in an element-wise manner. Note that in the image domain convolution, there is no requirement for  $o$  and  $k$  to be of equal size. The resulting shape of  $i$ , denoted by  $N_i$ , will be  $N_i = N_o + N_k - 1$ . To achieve the same resolution,  $o$  and  $k$  must be padded with zeros before transforming to the frequency domain. After  $O$  and  $K$  are multiplied and transformed back to the image domain to form  $i$ ,  $i$  must be cropped again to the right size. The code snippet below shows how this process is implemented in a Python function. Here, `np` points towards the math library `numpy` or, if supported by the hardware, to its GPU accelerated version `cupy`.

```

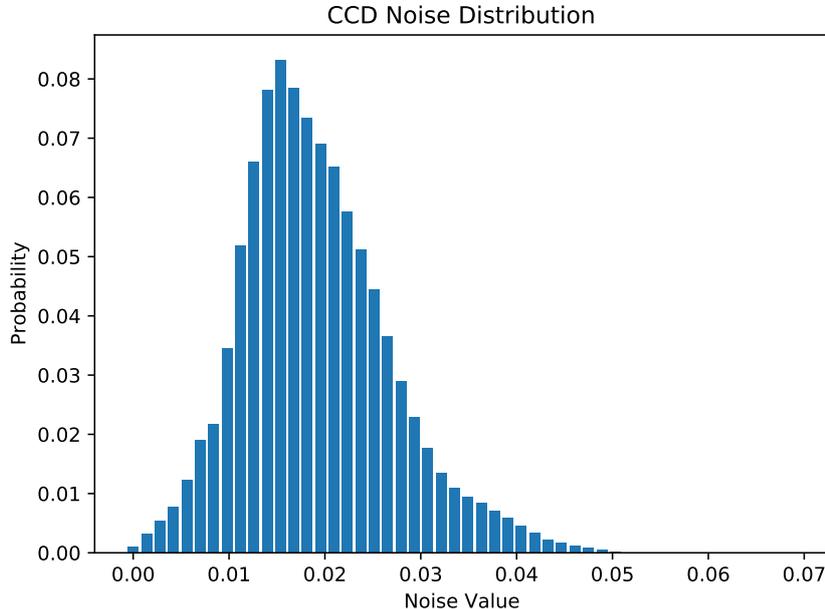
1 def convolve2(o, k):
2     # Add padding to the matrices
3     o_pad = np.pad(np.array(o), [0, k.shape[0]-1], mode='constant')
4     k_pad = np.pad(np.array(k), [0, o.shape[0]-1], mode='constant')
5
6     # Convolve the image and crop the edges
7     edge = np.minimum(o.shape[0], k.shape[0])/2
8     i_pad = np.real(np.fft.ifft2(np.fft.fft2(o_pad)*np.fft.fft2(k_pad)))
9     i = i_pad[edge:-edge+1, edge:-edge+1]
10    return i

```

## Sensor Noise

The final step is to add sensor noise to the generated images. The exact simulation of the noise distribution of a CCD can be complicated to reproduce and will depend on many variables, e.g. exposure time, scene lighting, background radiation, etc. Since there is access to the optical test setup, the noise distribution can be measured directly and is shown in Figure 4-3. The noise that is added to the simulated images is generated according to this distribution.

For the determination of the noise distribution, the CCD settings were adjusted such that the object was sufficiently bright. An object was chosen that did not cover the entire field of view, leaving black borders along the image. These borders were extracted and their pixel values were transformed to the distribution as seen in Figure 4-3.



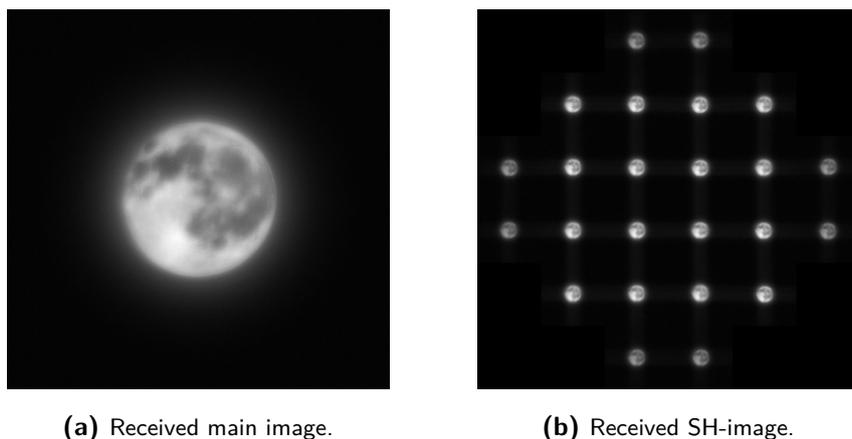
**Figure 4-3:** Visualization of the measured CCD noise distribution. Noise generated from this distribution is added to the simulated image, after the image is normalized between 0 and 1. The exact noise distribution depends on many parameters, e.g. frame rate, exposure time, background light, etc. In the simulation, the noise pattern is copied from the noise found in one particular camera setting.

# Deconvolution From Wavefront Sensing: System Design

This Chapter will walk through the design of the different subsystems of the DFWS system proposed in this Thesis. This starts with the preprocessing of the SH-image in Section 5-1, followed by DLWS step discussed in Section 5-2 and finally a brief note on the image deconvolution in Section 5-3.

## 5-1 Pre-Processing Shack-Hartmann Images

The pre-processing of the SH-image is done using a modified version of the MFBD algorithm TIP, which is discussed in Section 2-2-1. Usually in MFBD, the objective is to retrieve the object. In this case, however, the goal is to retrieve the SH-PSF. Figure 5-1 shows the image from the main CCD and the SH-image. Given these two images, the goal is to retrieve the SH-PSF such that it can be used as input for the deep learning wavefront sensing method.



**Figure 5-1:** An example of the information available from which the SH-pattern needs to be estimated. These images were generated from the optical simulation.

TIP begins with an initial estimate of either the object or PSF. In this Thesis, the image

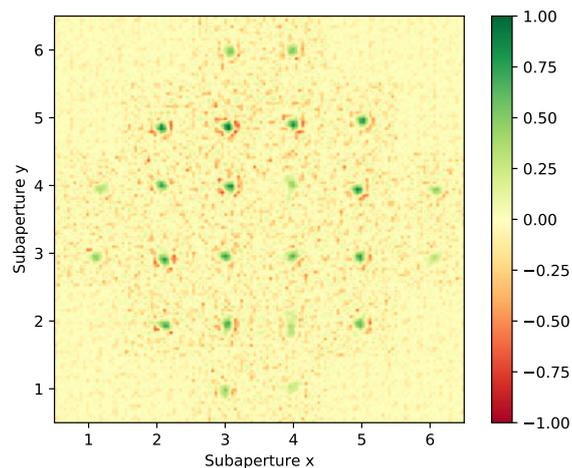
from the main CCD can be used as an initial estimate of the object. For this, the main image needs to be down-sampled in order to match the resolution of the smaller subaperture images. Distortions present in the main image have an influence on the resulting estimate of the SH-PSF. A proper choice of the projection operator can help to minimize this influence.

For the construction of the projection operator, consider the 3 possibilities describing the relation between the latest estimate of the object and the subaperture images:

1) The subaperture image is more blurred than the estimated object. This will result in a large PSF (e.g. subaperture (4, 2) in Figure 5-2).

2) The subaperture image is similarly blurred to the estimated object. This results in a PSF that is close to a delta function (e.g. subaperture (3, 3) in Figure 5-2).

3) The subaperture image is sharper than the estimated object. In this latter case, the PSF does not represent a blurring function, but instead a sharpening function. In order for the PSF to perform a sharpening action, it must contain negative values as well as large positive values. This effect is highlighted in Figure 5-2. Subaperture (3, 5) is sharper than the estimated object and contains negative values.



**Figure 5-2:** The estimated SH-pattern in the first TIP iteration, before the projection step. Three types of PSFs can be identified in this image: 1) The subaperture image is more blurred than the estimated object, resulting in a blurred PSF, e.g. (4, 2). 2) The subaperture image is similar to the estimated object, resulting in a sharp PSF, e.g. (3, 3). 3) The subaperture image is sharper than the estimated object, resulting in a PSF with a large positive values as well as negative values, e.g. (3, 5). Filtering out the subaperture images corresponding to 1) and discarding the negative values will direct the TIP algorithm to an accurately estimated object.

As the low values in the PSF point towards subaperture images that are more blurred than the estimated object, the TIP algorithm can be nudged into the right direction by decreasing the contribution of these subaperture images for the estimation. This can be done by increasing the contrast of the estimated PSF by normalizing it between 0 and 1 and raising it to a certain power.

The Python code below shows a snippet of the exact implementation of the TIP algorithm.

```
1 for n in range(iterations):
```

```

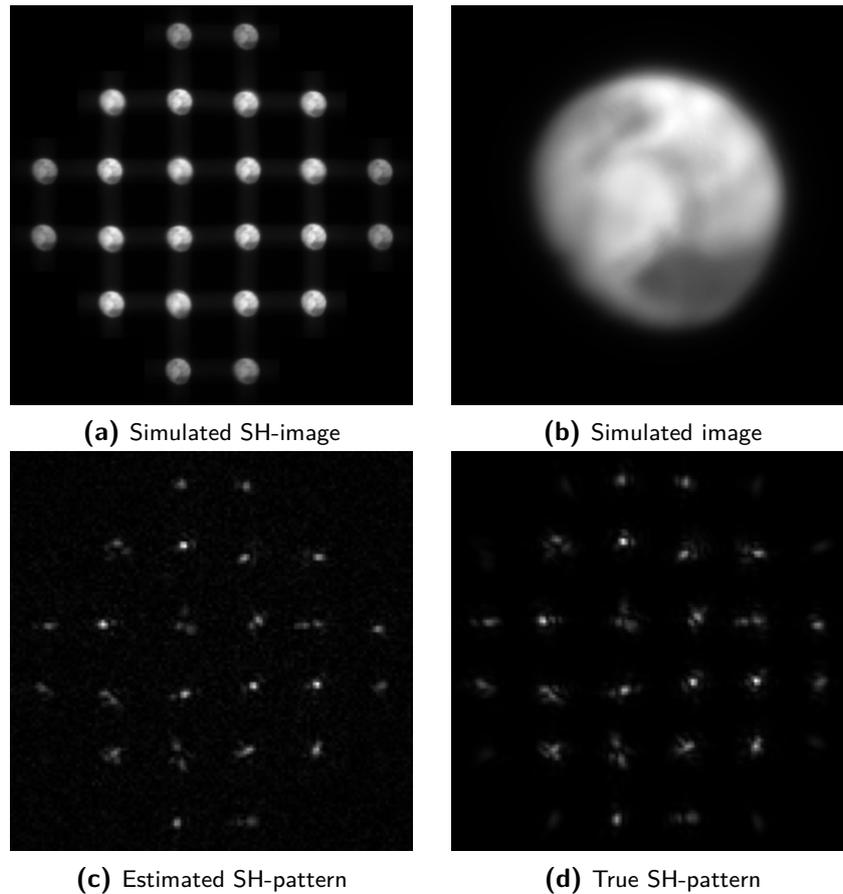
2     psf_est_F = i_F / (o_F + 1 * (np.abs(o_F) < 1))
3     psf_est = np.real(fft.ifft2(psf_est_F))
4     psf_est *= (psf_est > 0)
5
6     psf_est -= np.min(psf_est)
7     psf_est /= np.max(psf_est)
8     psf_est **= iterations - n
9
10    psf_est -= np.min(psf_est)
11    psf_est /= np.sum(psf_est)
12    psf_est_F = fft.fft2(psf_est)
13
14    conj = np.conj(psf_est_F)
15    o_F = (conj * i_F) / (conj * psf_est_F + 1e-9)
16    o = np.abs(fft.ifft2(o_F))
17    o *= setup.tip_pupil
18    o -= np.min(o)
19    o /= np.max(o)
20    o_F = fft.fft2(o)

```

Lines 2-4 perform a least-squares deconvolution and extract the positive real values from the estimated PSF. Note that the regularization parameter is only added to the frequencies with a lower absolute value than the regularization parameter in order to not distort valid frequencies. Lines 6-8 normalize the estimated PSF between 0 and 1 and increase the contrast. The exact power to which the image is raised is dependent on the iteration step and decreases to 1 in the final step (note that in Python  $0 \leq n < \textit{iterations}$ ). Lines 10-12 normalize the PSF such that the sum is equal to 1 and calculate the Fourier transform of the PSF. Lines 14-20 calculate the estimate of the object and normalizes it.

Unlike in the implementation of Wilding et al., the estimated object is the absolute value of the inverse Fourier transform. Wilding et al. calculates the object similar to the PSF (lines 2-4) by taking the real and positive entities of the inverse Fourier transform. In the case where one subaperture image is significantly sharper than the others, the increase in contrast will filter out only one subaperture image. In the next iteration, the estimated object will be equal to that subaperture image. After a number of iterations, the TIP algorithm will converge to a SH-PSF that consists of one single delta function at the location of that subaperture. By taking the absolute value in line 16, slight discrepancies will be introduced in the object, which helps to prevent the algorithm from converging to the aforementioned trivial solution. These discrepancies do not seem to appear in the estimated PSF.

Figure 5-3 shows how the implementation of the TIP algorithm performs on a simulation of the optical system. It can be seen that the estimated SH-pattern is nearly identical to the true SH-pattern and that the noise in the image is limited.



**Figure 5-3:** An example of the performance of the TIP algorithm based on a simulation of the optical system with a turbulence strength of  $D/r_0 = 20$ . In (c), some subaperture PSFs are bright and cover little pixels (i.e. are less aberrated), while others are dim and cover many pixels. By increasing the contrast of the SH-patterns (e.g. by raising it by a power larger than 1), the highly aberrated subapertures are filtered out. This helps the TIP algorithm find the correct object.

## 5-2 Deep Learning Wavefront Sensing

Now that the influence of the object is removed from the SH-image, DLWS can be used in order to retrieve a reconstruction of the wavefront. The selection of the CNN architecture has a large influence on the quality of the wavefront reconstruction.

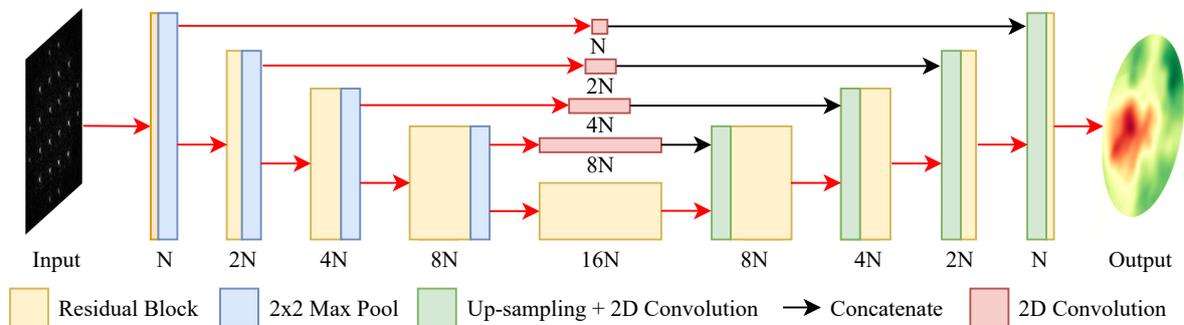
### 5-2-1 Selection of Architecture

The CNN architecture to be used for this Thesis' specific application is not necessarily the same CNN architecture found to be optimal in literature for similar applications (which are discussed in Section 3-2). There are several differences in SH-patterns used as input for the CNN between literature and this Thesis. Most notably, previous research has used the true SH-pattern, whereas the SH-pattern used in this Thesis is an estimate of the true SH-pattern found through a blind-deconvolution step. The blind deconvolution step not only introduces discrepancies in the shape of the individual PSFs, but also results in a lower signal-to-noise ratio.

From a practical point of view, especially in early research like this Thesis, a smaller network architecture is preferred, as it allows for faster iterations and more convenient tuning of specific parameters. Also, for real time application of the network, a smaller network is preferable both in terms of memory usage as well as evaluation time. The ideal architecture for this application is the network with the least parameters to be trained that is still able to extract all available information from the SH-pattern.

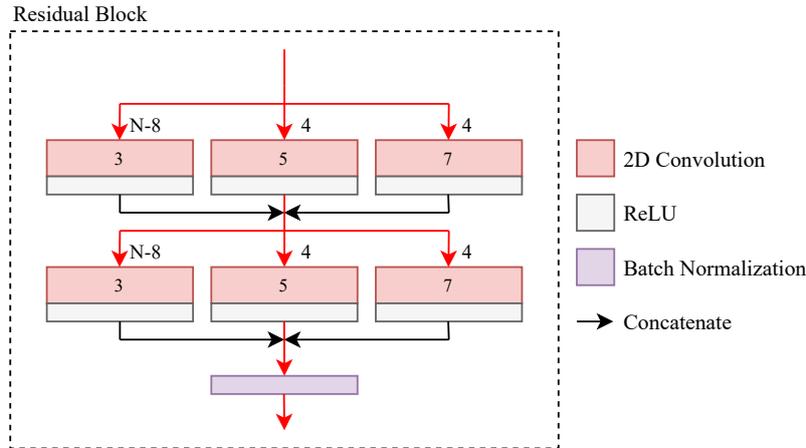
Recall the residual block of the architecture used by Hu et al. [12], shown in Figure 3-5, which consists of different sizes of convolutional kernels as well as a 'bypass' layer. The high amount of noise in the SH-pattern can be an issue if a 'bypass' of the convolutional layers is used. This method maintains features in the image without being filtered, and thus can preserve noise further in the network which results in a poorer performance on high noise images. Along the same line of thinking, a convolution layer between the concatenation from the different layers was added, resulting in a network architecture as displayed in Figure 5-4. The addition of these convolutional layers can filter out noise that is not removed by the earlier layers and prevents this noise from being concatenated in later layers.

Figure 5-5 shows the residual block used for this proposed architecture. This residual block has features based on the architecture of Hu et al. [12], where different sizes of convolutional kernels are used. The amount of larger size kernels is, however, significantly reduced and does not depend on  $N$ . As a result of this reduction, the size of the network is much lower than the architecture of Hu et al. because the amount of parameters in the convolutional layer scales with the kernel size squared. Additionally, there is another concatenation action introduced in the new residual block, located between the two convolutional layers. This concatenation action mixes the information of the differently sized convolutional layers and allows information to pass through two different sizes of kernels within the residual block.



**Figure 5-4:** A visualization of the general *U-net* architecture modified for the application in this Thesis, with  $N = 12$ . The convolution between the concatenation prevents noise present in the input image from propagating into deeper layers. The residual block used is displayed in Figure 5-5.

Chapter 6 will present a detailed comparison of the performance of the different architectures.



**Figure 5-5:** Residual block used in the proposed CNN architecture. Similarly to Hu et al. [12], different sizes of convolutional layers are used. The amount of larger convolutional layers is limited and the concatenation of the layers happens twice in the residual block.

## 5-2-2 Data Normalization

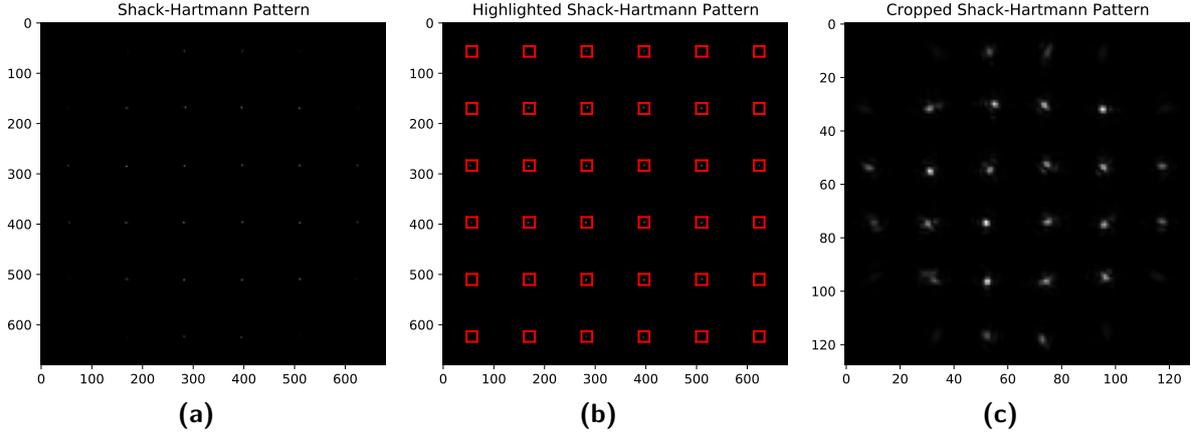
For the training of neural networks, it is important that both the input and output data are as constant as possible. The normalization of input and output data and the removal of data that cannot be reconstructed or does not contain information is therefore very important.

The SH-pattern is normalized such that the lowest pixel value is equal to 0 and the highest pixel value is equal to 1. Additionally, the shape of the SH-pattern is reduced to  $128 \times 128$  pixels from its original size of  $680 \times 680$  pixels.

Figure 5-6a shows an example of a SH-pattern as generated by the simulation environment. Due to the relatively low amount of subapertures, there are large spaces within the SH-pattern that do not contain any information. In Figure 5-6b, the areas where the subaperture PSFs are present are highlighted. The size of these areas is chosen such that they form an image of  $128 \times 128$  pixels when put together. This image is shown in Figure 5-6c and is used as the input for the neural network. Hu et al. [12] and Bekendam [1] both use an input image of  $256 \times 256$ , which is needed for the larger amount of subapertures in the SHWFS. The reduced image size significantly speeds up the training and evaluation process, as all element-wise and frequency domain transformation operations are sped up by a factor of  $2^2$  and  $2 \log(2)$ , respectively.

The output data, i.e. the wavefront, cannot be normalized between 0 and 1, as this will disrupt the strength of the wavefront aberrations. It is, however, important to remove the data of the wavefront that cannot be estimated for the SH-pattern.

The SH-pattern is not influenced by the first Zernike mode (piston). Additionally, the preprocessing step of the extended-scene SH-image has been observed to introduce uniform shifts in the estimated SH-patterns. As seen from the TIP-algorithm, the SH-pattern and estimated object can both have an equal but arbitrary shift introduced, as long as the SH-pattern stays in the image frame, while still being a valid solution to the blind deconvolution problem. Since the global shift in the SH-pattern corresponds to the tip and tilt modes in the wavefront, these modes can be corrupted by the preprocessing step. Fortunately, the piston, tip and tilt modes (i.e. the first 3 Zernike polynomials) do not influence the resulting shape



**Figure 5-6:** (a): The original SH-pattern, (b): the regions of interest within the SH-pattern highlighted, (c): the cropped SH-pattern, consisting of the collection of the highlighted areas in (b). This conversion loses no data from the original wavefront and allows the neural network to speed up significantly as it does not have to process the empty space within the subapertures.

of the PSF and are therefore not necessary in the DFWS setting.

It is important for the neural network that the 'invisible' wavefront modes are removed in the training process. To remove these modes, the first step is to estimate the first 3 Zernike modes of the wavefront using

$$\begin{aligned}\phi &= cZ. \\ c &= \phi Z^T (Z Z^T)^{-1}.\end{aligned}\tag{5-1}$$

Here,  $\phi$  is the vectorized wavefront of shape  $[128^2]$ ,  $c$  is the array of Zernike coefficients of size  $[3]$  and  $Z$  is the vectorized array of Zernike polynomials of size  $[128^2, 3]$ . The influence of these wavefront modes can now be removed by

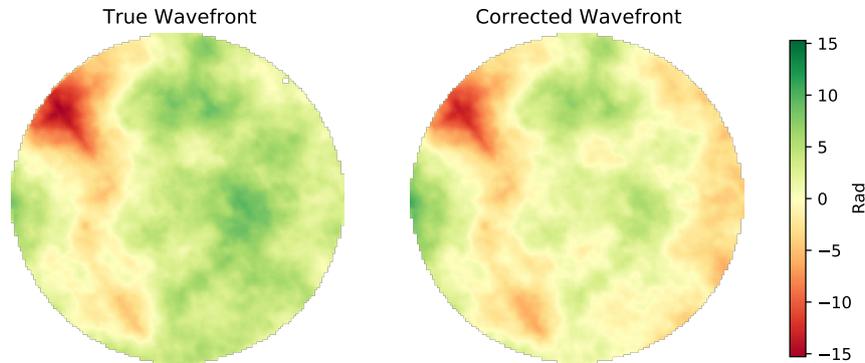
$$\phi_{\text{corr}} = \phi - cZ.\tag{5-2}$$

Where  $\phi_{\text{corr}}$  is the corrected wavefront with removed piston, tip and tilt modes. Figure 5-7 shows an example of the wavefront before and after its correction.

## 5-3 Image Deconvolution

Chapter 2 already concluded that iterative methods are generally better at handling the ill-conditioned properties of the image deconvolution step. Now, it must be decided which iterative deconvolution method is best to use in this Thesis and for how many iterations. Two methods are considered, the Landweber method with  $\tau = 1.5$  and steepest descent.

The amount of iterations to use can be either a fixed number, or determined by a stopping criterion. In the context of real-time image deconvolution, a variable amount of iterations resulting from the use of a stopping criterion can be disadvantageous as it can result in a variable frame rate. It is discussed in Section 6-3 that the bottleneck in terms of real-time performance is the DLWS step, which takes roughly 0.1sec to calculate. A calculation time



**Figure 5-7:** On the left, the original wavefront. On the right, the wavefront with corrected piston, tip and tilt modes. The piston mode of the wavefront cannot be reconstructed from the SH-pattern. The information about tip and tilt modes can be corrupted by the extended scene preprocessing step of the SH-PSF. Hence, these modes must be removed from the wavefront to improve the training.

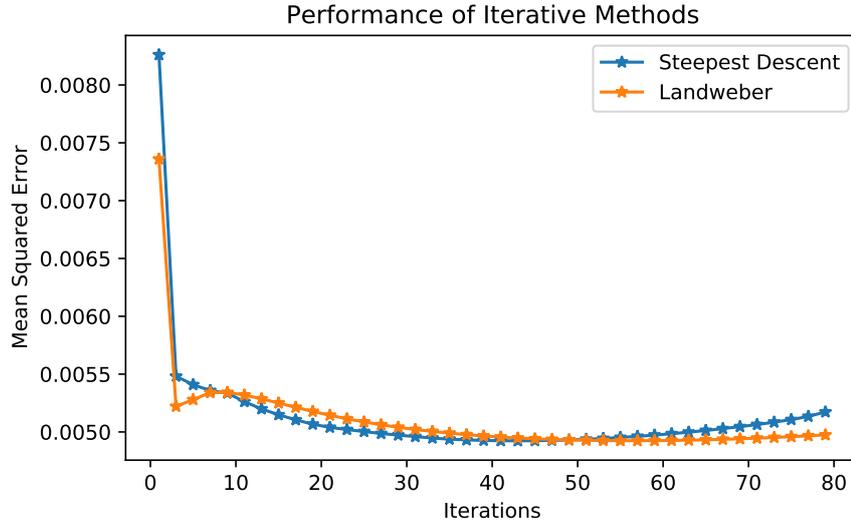
of  $0.1\text{sec}$  corresponds to about 80 iterations for the Landweber method and 40 iterations for steepest descent. If a stopping criterion is used, it must be limited to these amounts of iterations to maintain a constant frame rate in the system's integration.

Figure 5-8 shows the mean square error (MSE) between the deconvoluted estimated object and the true object plotted against the amount of iterations in the deconvolution method. These curves vary greatly case-to-case, but can still help to illustrate the considerations that must be taken in account for the selection of stopping criterion or number of iterations.

Firstly, notice that both curves have an optimum somewhere around 50 iterations. The steepest descent method converges faster to this optimum, but also diverges faster when too many iterations are performed. Secondly, the convergence of the Landweber method is not convex because the MSE increases between 2 and 10 iterations. Lastly, it is clear that the increase in convergence speed of the steepest descent method does not weigh up against the increased time per iterations. The steepest descent method converges to the optimal 20% quicker, but the computation time per step is almost 2 times longer.

As the optimal amount of iterations is variable case-to-case, it is chosen in this Thesis to use a stopping criterion to determine the amount of iterations to use. The criterion is constrained by a minimum and maximum allowed number of iterations. The minimum amount of iterations is set to 20 and is used to force the deconvolution method past the non-convex part, which is generally located below  $n = 20$ . The upper limit to the number of iterations is set to 80, to ensure a constant frame-rate.

The stopping criterion must not contain computationally expensive math operators in order to avoid slowing down the deconvolution time. Recall the deconvolution iteration equation



**Figure 5-8:** The convergence of the Landweber and steepest descent deconvolution methods.

$$\tilde{O}^{n+1} = \tilde{O}^n + \tau K^T (I - K^* \tilde{O}^n). \quad (5-3)$$

For a converging iteration, it must hold that

$$\lim_{n \rightarrow \infty} \tilde{O}^{n+1} - \tilde{O}^n = 0. \quad (5-4)$$

Therefore,

$$\lim_{n \rightarrow \infty} \tau K^T (I - K^* \tilde{O}^n) = 0. \quad (5-5)$$

As long as the increments in the object estimate ( $\tilde{O}^n$ ) are decreasing, it is assumed that the iteration is converging. If the increments start increasing, the object estimate is diverging. Hence, the iteration is stopped when

$$\sum \left( \tau K^T (I - K^* \tilde{O}^{n+1}) \right)^2 > \sum \left( \tau K^T (I - K^* \tilde{O}^n) \right)^2. \quad (5-6)$$

I.e. the iteration is stopped when the MSE of the increment increases (or when  $n > 80$ ). Since the increment of the object estimate is calculated for updating the estimated object, calculating its mean squared error decreases the computational speed only marginally.

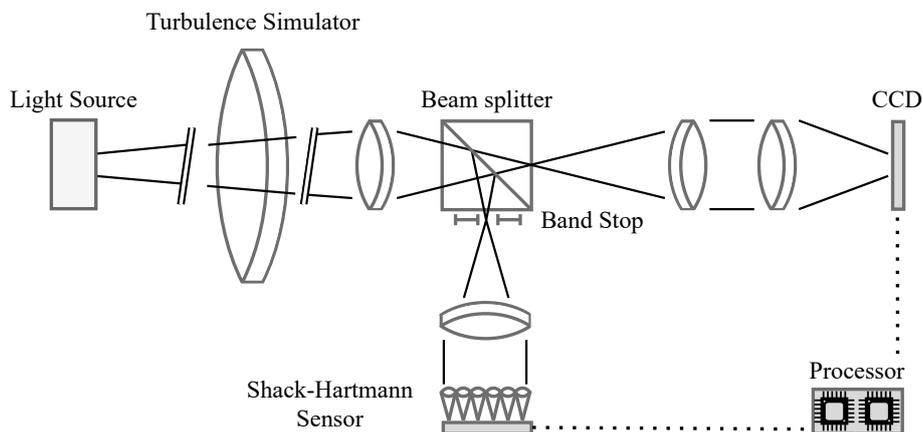


## Experimental Results

In this Chapter, the experimental results of the methods implemented in this Thesis are discussed. Firstly, Section 6-1 explains the optical setup that is used for testing the system. Section 6-2 will show the performance of the developed deep learning wavefront sensing methods and compare their performance to conventional methods for interpreting extended scene SHWFS images.

### 6-1 Experimental Setup

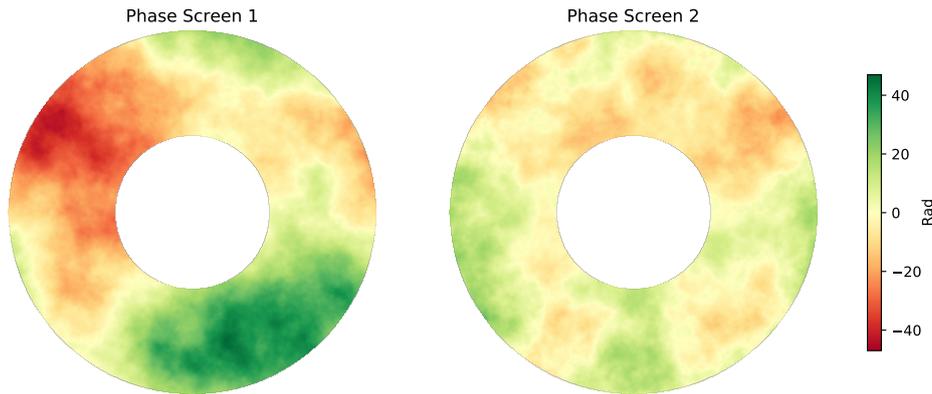
The system will be implemented on an optical test setup which is made to replicate a small diameter telescope. A test setup is preferred over a real telescope because it allows for controlled experiments. An overview of the test setup is shown in Figure 6-1.



**Figure 6-1:** A schematic of the optical test setup used. The lens left of the beam splitter is chosen such that its numerical aperture corresponds to that of a small telescope. The light source consists of an LED with a wavelength of  $470nm$  with a object placed in front of it. The light source is placed sufficiently far from the lens such that the wavefront is approximately flat when it reaches the lens. As the light source takes up only a small section of the field of view, the subaperture of the SHWFS does not overlap, leaving the band stop unused.

Except for the light source and turbulence simulator, the setup is nearly identical to a

DFWS system as discussed in Chapter 1. The lens left of the beam splitter is chosen such that its numerical aperture corresponds roughly to that of a 130mm diameter telescope with a focal point of  $10^3\text{mm}$ . The light source is an LED light ( $\lambda = 470\text{nm}$ ) behind which either a pinhole can be placed in order to simulate a point source, or a piece of paper with a figure to simulate an extended scene. From the source, the light passes through a turbulence simulator. This is a disk that introduces known wavefront aberrations and can be rotated using a motor in order to simulate a regressing wavefront. Figure 6-2 displays the wavefronts on the available turbulence simulators.



**Figure 6-2:** A visualization of the wavefronts introduced by the 2 turbulence simulators. The strength of the turbulence can be changed not only by moving the turbulence simulator closer or farther from the light source, but also by switching turbulence simulator.

It is chosen to not collimate the beam after the light source, even though this would better simulate the light that comes into a telescope from a star at infinity. Instead, the light source is placed at a distance of roughly 12 times the focal point of the lens before the beam splitter. This distance between the lens and the source results in an approximately flat wavefront at the lens and has the benefit that the strength of the turbulence can be adjusted continuously by either placing the turbulence simulator closer to the light source or closer to the lens. If the turbulence simulator is placed closer to the source, the light will pass through a small section of it. If the turbulence simulator is closer to the lens, the light will pass through a larger section of turbulence. There is a limit to how low the strength of the simulated turbulence can be, as anisoplanatic effects will arise when the turbulence simulator is placed too close to the light source.

The processor in Figure 6-1 is a desktop PC with a 12 core Intel Xeon E5-2630 DUAL CPU with 64 GB of memory and a NVIDIA GeForce GTX 970 GPU (CUDA enabled), with 4 GB of video memory. The same system is used for the training of the CNNs.

## 6-2 Wavefront Sensing

For the testing of the wavefront sensing capabilities of the newly developed method, the 3 different CNNs used for deep learning wavefront sensing are evaluated on a turbulence strength ranging from  $D/r_0 = 0$  to  $D/r_0 = 18$  in 19 discrete steps. For each step in turbulence strength, 100 unique wavefronts were generated which were not included in the training data set. The tests were performed using the software simulation of the optical setup for easier calculation

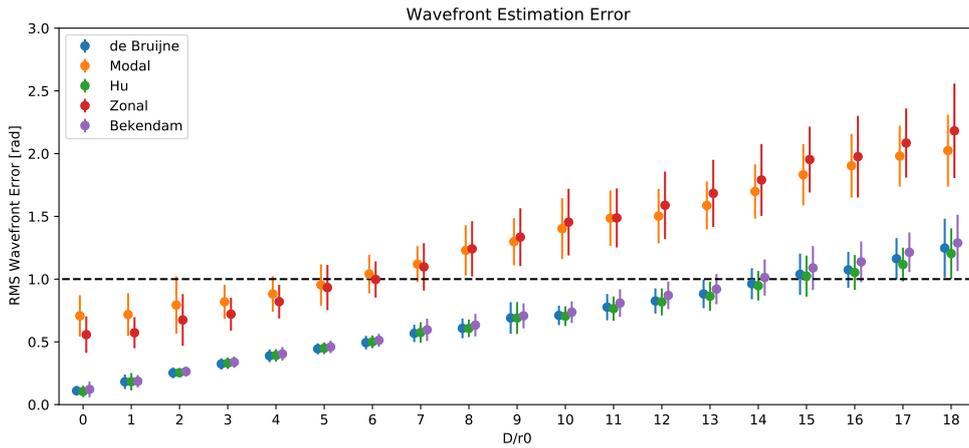
of the wavefront reconstruction performance.

To compare the developed methods to the conventional way of extended scene wavefront sensing, a zonal and modal method were implemented, too. For the conventional methods, the slope detection was done in accordance to Zhou et al. [40], using the absolute difference function as correlation function combined with sub-pixel interpolation in order to retrieve subaperture shifts on an accuracy higher than 1 pixel. The modal method represents the estimated wavefront using the first 24 Zernike modes, equal to the number of effective subapertures.

Figure 6-3 shows the results of the comparison. In this Figure, the 1rad RMS wavefront estimation error is highlighted, indicating the minimum required performance. The RMS wavefront error is calculated by

$$e_{\text{RMS}} = \sqrt{\sum_{x=1}^X \sum_{y=1}^Y (\phi(x, y) - \tilde{\phi}(x, y))^2}. \quad (6-1)$$

Where  $\tilde{\phi}$  is the reconstructed wavefront and  $\phi$  is the true wavefront of size  $[X, Y]$ .



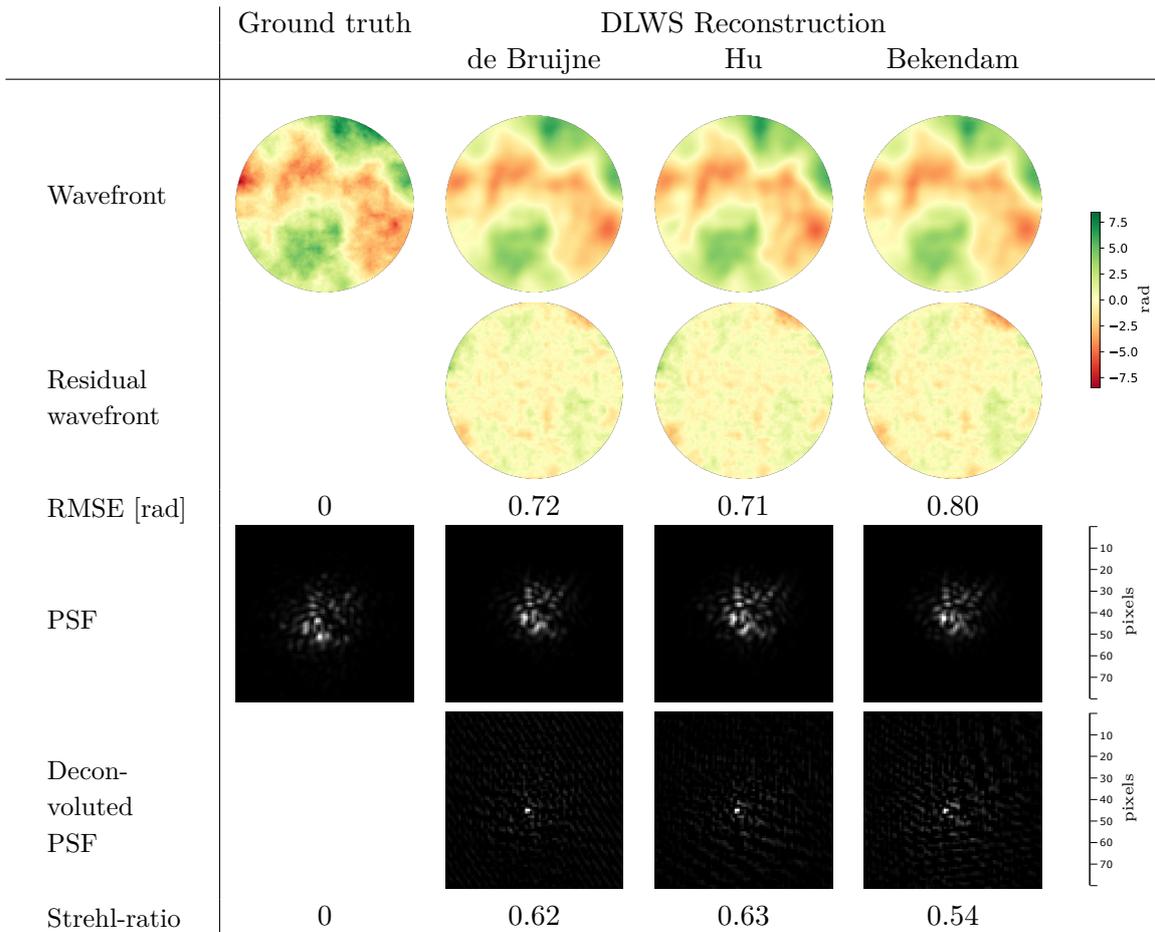
**Figure 6-3:** An overview of the wavefront estimation performance of the different developed deep learning wavefront sensing methods, compared to the traditional zonal and modal wavefront sensing methods. The wavefronts used for evaluation are 1900 unique wavefronts that were not included in the training data for the CNNs. For each discrete step in turbulence between  $D/r_0 = 0$  and  $D/r_0 = 17$ , 100 wavefronts were generated. The dot represents the mean value over the 100 tested wavefronts and the error line represents the standard deviation of the error. For readability, some data points are placed slightly before or after the integers on the  $x$ -axis. This does not reflect a difference in tested wavefront strength between the different methods.

As expected, the zonal method manages to sufficiently estimate the wavefront up to a turbulence strength of roughly  $D/r_0 = 6$ , which is equal to the amount of subapertures in the SHWFS. Since the wavefront is generated using Kolmogorov statistics rather than Zernike polynomials, the performance of the modal method is lower than that of the zonal method. The performance of the tested CNNs is very similar, all estimating the wavefront sufficiently up to a turbulence strength of  $D/r_0 = 14$ , which is a very significant improvement compared to the modal and zonal methods.

Up to  $\frac{D}{r_0} = 12$ , the newly proposed architecture performs slightly, but not significantly

better than Hu's architecture, while both perform roughly 4% better than Bekendams architecture. At turbulence strengths of  $\frac{D}{r_0} > 12$  Hu's architecture performs roughly 2% better than the proposed architecture. It must be noted that earlier in the development of the SH-image preprocessing algorithm, the images used as input for the CNNs contained more noise and the proposed network performed better with the lower signal to noise ratio.

Figure 6-4 shows an example of the wavefront reconstruction performance of the tested CNN architectures. It can be seen that the DLWS method is able to reconstruct the global shape of the wavefront accurately, but fails to retrieve the pixel-to-pixel fluctuations in the turbulent wavefront.

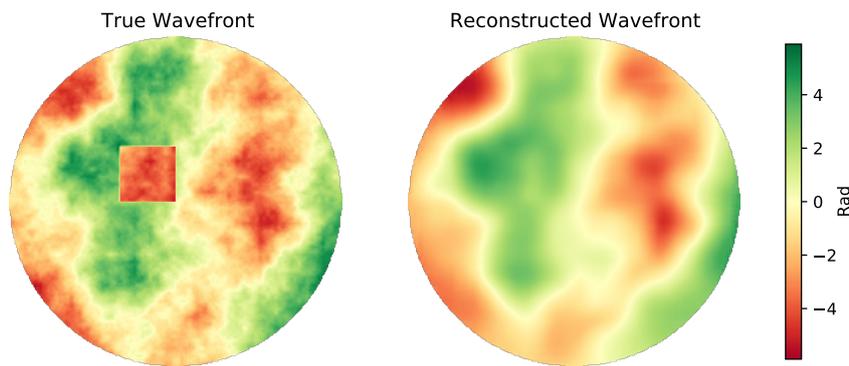


**Figure 6-4:** An example of the wavefront sensing performance of the 3 tested DLWS architectures. The tested wavefront has a turbulence strength of  $D/r_0 \approx 12$  and all networks reconstruct the wavefront sufficiently. The DLWS architectures perform overall very similar and it does not appear that one particular architecture fails or succeeds to recognize wavefront features that the others do not.

Figure 6-4 also shows how the PSFs of the reconstructed wavefronts, as well as the deconvolution of the true PSF with that of the reconstructed wavefronts. This deconvoluted PSF depends on the used deconvolution technique but gives an idea of the residual PSF after the image has been deconvoluted. The noise in these PSFs makes the calculation of the Strehl-ratio unreliable. Instead, the Strehl-ratios of the PSFs resulting from the residual

wavefronts is shown.

Interestingly, the DLWS methods show perfect phase unwrapping behavior. Areas with added phase delays of multiples of  $2\pi$  rad do not influence the SH-pattern and are therefore invisible to the DLWS methods. The training data for the CNNs consisted of continuous wavefronts (i.e. no large steps in phase delay), which has led to the DLWS only reconstructing continuous wavefronts. This effect is highlighted in Figure 6-5, where a jump in phase is manually added to the true wavefront, but the DLWS method reconstructs a wavefront with a phase wrap that corresponds to the turbulent wavefront without the added jump in phase.



**Figure 6-5:** A visualization of the phase unwrapping behavior of the DLWS methods. The DLWS methods have been trained to reconstruct a wavefront with a continuous phase, if a discontinuous jump in phase of  $2\pi$  rad is added to the wavefront, this will not be present in the reconstructed wavefront.

## 6-3 Real Time System Integration

Especially for real-time operations, it is important to not only compare the methods by residual wavefront error, but also computational speed. Table 6-1 shows an overview of the amount of parameters in the network, the evaluation time and the networks memory requirement.

**Table 6-1:** Computational efficiency evaluation of the 3 tested DLWS CNNs. The evaluation times depend greatly on the amount of background processes active. These tests were performed while the other parts of the DFWS system were also running.

CNN	Number of parameters	Evaluation time [s]	Memory [MB]
Hu	$2.14 \cdot 10^7$	0.145	252
Bekendam	$4.86 \cdot 10^5$	0.092	6.29
de Bruijne	$1.45 \cdot 10^6$	0.099	18.4

It can be seen that the newly proposed network is almost triple the size of Bekendam's network, both in terms of parameters and memory required. Hu's architecture is roughly another 15 times larger.

In the software implementation developed in this Thesis, the process is divided into 7 parts as shown in Table 6-2. Depending on the systems resources, a number of these steps can be performed in parallel. In a parallel setting, different functions of the program can be performed simultaneously. For example, one image can be deconvoluted while the next image is still in the wavefront sensing stage. In this parallel setting, the steps with shorter evaluation times must either be combined or have an added delay in order to synchronize the various processes. In this particular implementation no steps are combined, but instead a delay scheduling is used. The delay scheduling can have a delay introduced either before or after the step synchronization in order to spread the computational load evenly along the step's length. A disadvantage of this approach is that the lag between receiving the input is longer, defined by

$$lag = N||t_n||_{\infty}. \quad (6-2)$$

With  $N$  the amount of steps and  $t_n$  the evaluation time of step  $n \leq N$ . If all the steps are implemented in series, it is not necessary to add delay and the output lag is reduced to

$$lag = \sum_{n=0}^N t_n. \quad (6-3)$$

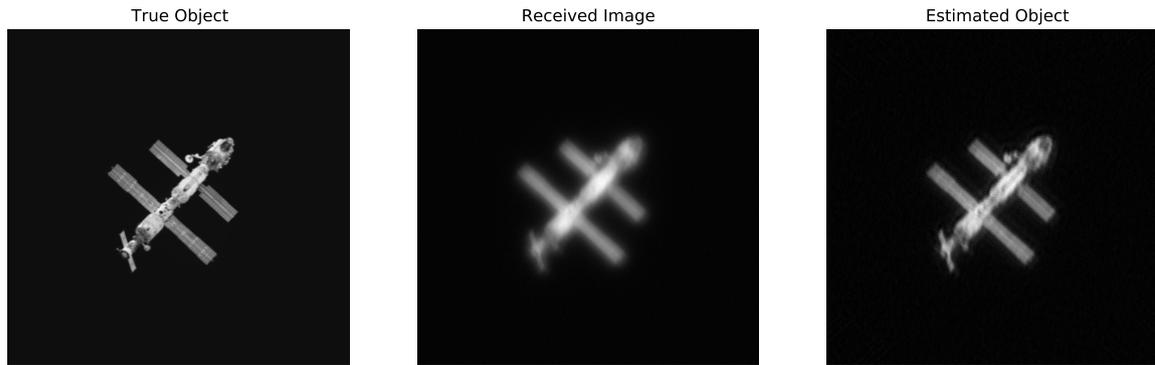
**Table 6-2:** An overview of the different sections that the software implementation is divided in, combined with their respective average computation time. Dividing the system up into these discrete parts allows for efficient parallel computation. This increases the time between receiving the image and estimating the object slightly, but allows for the calculation of multiple objects from sequential images simultaneously.

Section	Average Run Time ( $t_n$ ) [ms]
1) Retrieving the image from the setup	53
2) Performing the TIP pre-processing step	42
3) Cropping and normalizing the estimated SH-PSF	4
4) Evaluating the DLWS CNN	see Table 6-1
5) Constructing the PSF from the estimated wavefront	25
6) Performing the image deconvolution	< 99
7) Rendering the estimated object	40

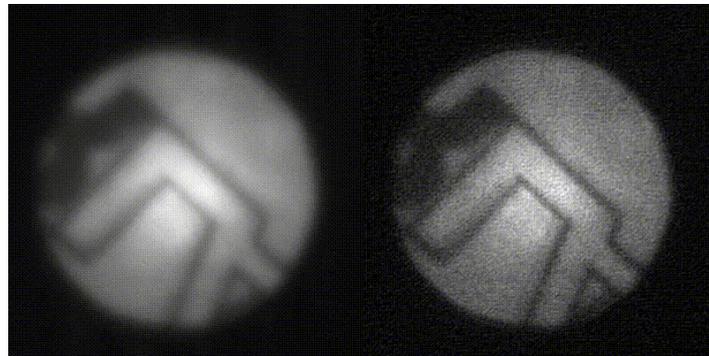
Tables 6-2 and 6-1 show that the bottleneck of the system is the evaluation of the CNN. Hence, Hu's architecture is unfavorable for real-time operation, as it limits the system to roughly  $7Hz$ . The newly proposed architecture limits the system to  $10Hz$  and Bekendams architecture to  $11Hz$ .

## 6-4 Examples

Figure 6-6 shows the performance of the DFWS on a simulated scene. The system is able to correct the observation to the diffraction limited image. Figure 6-7 also shows the system performing on the real world optical setup. It is clear that the system is able to increase the sharpness of the image significantly.



**Figure 6-6:** A display of the performance of the system on in a simulated situation. The turbulence strength is  $D/r_0 \approx 15$ .



**Figure 6-7:** Figure displaying the real time, real world correction capabilities of the system developed in this Thesis. The left picture shows the received image and the right picture shows the estimated object. This figure shows the inability of the developed system to correct tip and tilt modes. The estimated object is sharp but not centered.

## 6-5 Conclusions

In this Chapter, the performance of the DFWS system developed in this Thesis was evaluated. 3 different CNN architectures were compared and the real-time image correction capabilities of the system were showcased.

It can be concluded that the novel extended scene deep learning approach to wavefront sensing is both more reliable and can be used in stronger turbulence than the conventional approach to extended scene wavefront sensing. With this method, a  $15 \times 15$  SHWFS can reliably be replaced by a  $6 \times 6$  SHWFS, which increases the amount of light per subaperture more than five fold.

The selection of CNN architecture to be used is a trade-off between computation time and wavefront reconstruction accuracy. As stated in the Introduction, the goal is to have a refresh rate of  $10Hz$ . The best performing architecture that adheres to this goal is the newly proposed architecture.

It is shown that the system is able to correct incoming images at a rate of  $10Hz$ . It must be noted that these results are from a PC with a CPU that is 9 years old and a GPU of 7 years old at the time of writing. The refresh rate is expected to be significantly higher on

modern day consumer grade PCs and with more optimized implementation.

# Conclusion and Discussion

In this Thesis, a novel approach to extended scene wavefront sensing was proposed. The potential of this method was showcased in a real-time deconvolution from wavefront sensing system.

It was shown that a modified Tangential Iterative Projections algorithm can be used in order to reduce an extended scene Shack-Hartmann image to its point-source-equivalent. This eliminates the dependency of the scene content on the Shack-Hartmann image. Deep learning wavefront sensing techniques can then be used to reconstruct the wavefront from the point-source Shack-Hartmann image. For this approach, it is not necessary for the subaperture images to be diffraction limited because the deep learning method can (to a certain extent) retrieve wavefront information from the shape of the point-spread function.

Using this approach, a  $6 \times 6$  microlens Shack-Hartmann sensor was shown to be able to reconstruct the wavefront up to a turbulence strength of  $D/r_0 = 15$ . Using a conventional wavefront sensing method, a  $15 \times 15$  microlens Shack-Hartmann sensor would be necessary to achieve this accuracy. As a result of this reduction in microlenses, more than 4 times the amount of light becomes available per subaperture, allowing for the wavefront correction in low-light scenes.

This novel wavefront sensing approach was implemented in a deconvolution from wavefront sensing system. This type of adaptive optics system uses the information about the reconstructed wavefront in order to form an estimation on how the image from a secondary imaging sensor is corrupted. This corruption was reversed using a process called image deconvolution. This system was applied to a real world optical setup, showing impressive image correction capabilities and a refresh rate of roughly  $10Hz$  with an output lag of roughly  $0.7sec$ .

## 7-1 Future work

The proposed wavefront sensing method is still in early development and currently has a number of limitations and questions.

It was observed that the Tangential Iterative Projections algorithm used for preprocessing the Shack-Hartmann image occasionally introduces shifts in the estimated Shack-Hartmann pattern. As a result of this, the proposed deconvolution from wavefront sensing system was

not able to correct the tip and tilt modes of the wavefront. Additional constraints on the PSF and object could be added in the TIP implementation in order to eliminate the occurrence of these shifts.

The developed system is currently limited to extended objects surrounded by a dark background, which limits its employment in, for example, satellite or surveillance imaging. It is expected that using a band stop to introduce a dark background into the subaperture images can elevate this limitation, but this needs to be verified in future research.

This Thesis investigated the performance of the system using a  $6 \times 6$  microlens array. Bekendam [1] showed promising wavefront reconstruction capabilities for larger microlens array sizes using point source deep learning wavefront sensing techniques. This trend is believed to be true for the proposed system as well, but future research will have to verify this assumption.

More research is needed to explore the potential of the proposed wavefront sensing method in other areas of adaptive optics. For example, extended scene deep learning wavefront sensing can be extended to anisoplanatic imaging or combined with the Frozen Flow Hypothesis in order to utilize temporal correlations of turbulence for more accurate wavefront reconstruction.

It is expected that further improvements to the software implementation of the proposed system will increase its real world reliability and speed. For commercial deployment of the system, the code will likely need to be rewritten from Python to C++.

---

# Appendix A

---

## Appendix

### A-1 Python Code

The Python code developed in this Thesis can be found under <https://github.com/Basdbuijne/MscThesis>.

#### A-1-1 Implementation of the convolutional neural network used for deep learning wavefront sensing

```
1 inputs = keras.Input((128, 128, 1))
2 filter_size = [12, 24, 48, 96, 192]
3 activation = 'relu'
4
5 def residual_block(N, inp):
6     for ii in range(2):
7         c = []
8         c.append(Conv2D(4, (7, 7), activation=activation, padding='same')(inp))
9         c.append(Conv2D(4, (5, 5), activation=activation, padding='same')(inp))
10        c.append(Conv2D(N-8, (3, 3), activation=activation, padding='same')(inp))
11        inp = concatenate(c)
12    return inp
13
14 # Setup the input layers
15 p = [inputs]
16 cin = [inputs]
17 for i in range(len(filter_size)):
18     cin.append(p[-1])
19     cin[-1] = residual_block(filter_size[i], cin[-1])
20     # The last input layer does not have a pooling layer
21     if i < len(filter_size)-1:
22         p.append(MaxPooling2D((2, 2))(cin[-1]))
23         p[-1] = BatchNormalization()(p[-1])
24         cin[-1] = Conv2D(filter_size[i], (3,3), activation=activation, padding='same'
25                             )(cin[-1])
26
27 # Setup the output layers
28 u = [cin[-1]]
29 for i in range(len(filter_size)-1):
```

```
29 u.append(Conv2DTranspose(filter_size[len(filter_size)-2-i], (2, 2), strides=(2,
2), padding='same')(u[-1]))
30 u[-1] = concatenate([u[-1], cin[len(filter_size)-1-i]])
31 u[-1] = residual_block(filter_size[len(filter_size)-2-i], u[-1])
32
33 outputs = keras.layers.Conv2D(1, (1, 1), activation='linear')(u[-1])
34
35 model = keras.Model(inputs=inputs, outputs=outputs)
```

---

# Bibliography

- [1] Bekendam, J. (2020). Deep learning wavefront sensing via raw shack-hartmann images. Master's thesis, Delft University of Technology.
- [2] Berisha, S. and Nagy, J. (2014). Iterative methods for image restoration. *Academic Press Library in Signal Processing*, 4.
- [3] Biggs, D. and Andrews, M. (1997). Acceleration of iterative image restoration algorithms. *Applied optics*, 36:1766–75.
- [4] Chen, G., Gao, Z., Wang, Q., and Luo, Q. (2019). U-net like deep autoencoders for deblurring atmospheric turbulence. *Journal of Electronic Imaging*, 28(5):1 – 14.
- [5] Costello, T. and Mikhael, W. (2003). Efficient restoration of space-variant blurs from physical optics by sectioning with modified wiener filtering. *Digital Signal Processing*, 13:1–22.
- [6] Díaz-Baso, C. and Asensio-Ramos, A. (2017). Enhancing sdo/hmi images using deep learning. *Astronomy and Astrophysics*, 614.
- [7] Ford, S., Welsh, B., and Roggemann, M. (1998). Constrained least-squares estimation in deconvolution from wave-front sensing. *Optics Communications*, 151(1):93 – 100.
- [8] Fried, D. L. (1966). Optical resolution through a randomly inhomogeneous medium for very long and very short exposures. *J. Opt. Soc. Am.*, 56(10):1372–1379.
- [9] Guo, H., Korablinova, N., Ren, Q., and Bille, J. (2006). Wavefront reconstruction with artificial neural networks. *Opt. Express*, 14(14):6456–6462.
- [10] Hsieh, Y. H., Yu, Y. T., Lai, Y. H., Hsieh, M. X., and Chen, Y. F. (2020). Integral-based parallel algorithm for the fast generation of the zernike polynomials. *Opt. Express*, 28(2):936–947.
- [11] Hu, L., Hu, S., Gong, W., and Si, K. (2019). Learning-based shack-hartmann wavefront sensor for high-order aberration detection. *Opt. Express*, 27(23):33504–33517.
- [12] Hu, L., Hu, S., Gong, W., and Si, K. (2020). Deep learning assisted shack–hartmann wavefront sensor for direct wavefront detection. *Opt. Lett.*, 45(13):3741–3744.

- [13] Jefferies, S. and Hart, M. (2011). Deconvolution from wave front sensing using the frozen flow hypothesis. *Optics express*, 19:1975–84.
- [14] Lane, R., Glindemann, A., and Dainty, J. (1992). Simulation of a kolmogorov phase screen. *Waves in Random Media*, 2(3):209–224.
- [15] Lofdahl, Mats (2010). Evaluation of image-shift measurement algorithms for solar shack-hartmann wavefront sensors. *Astronomy and Astrophysics*, 524.
- [16] Lucy, L. (1974). An iterative technique for the rectification of observed distributions. *aj*, 79:745.
- [17] Meinel, E. S. (1986). Origins of linear and nonlinear recursive restoration algorithms. *J. Opt. Soc. Am. A*, 3(6):787–799.
- [18] Mugnier, L., Robert, C., Conan, J., Michau, V., and Salem, S. (2001). Myopic deconvolution from wave-front sensing. *Journal of the Optical Society of America. A, Optics and image science*, 010735030203190.
- [19] Neal, D. R., Copland, J., and Neal, D. A. (2002). Shack-Hartmann wavefront sensor precision and accuracy. In Duparré, A. and Singh, B., editors, *Advanced Characterization Techniques for Optical, Semiconductor, and Data Storage Components*, volume 4779, pages 148 – 160. International Society for Optics and Photonics, SPIE.
- [20] Nishizaki, Y., Valdivia, M., Horisaki, R., Kitaguchi, K., Saito, M., Tanida, J., and Vera, E. (2019). Deep learning wavefront sensing. *Opt. Express*, 27(1):240–251.
- [21] Noort, M., Voort, L., and Löfdahl, M. (2005). Solar image restoration by use of multi-frame blind de-convolution with multiple objects and phase diversity. *Solar Physics*, 228:191–215.
- [22] Primot, J., Rousset, G., and Fontanella, J. C. (1990). Deconvolution from wave-front sensing: a new technique for compensating turbulence-degraded images. *J. Opt. Soc. Am. A*, 7(9):1598–1608.
- [23] Richardson, W. H. (1972). Bayesian-based iterative method of image restoration\*. *J. Opt. Soc. Am.*, 62(1):55–59.
- [24] Rimmele, T. and Radick, R. (1998). Deconvolving solar images using a Shack-Hartmann wavefront sensor. In Bonaccini, D. and Tyson, R. K., editors, *Adaptive Optical System Technologies*, volume 3353, pages 1014 – 1021. International Society for Optics and Photonics, SPIE.
- [25] Roggemann, M., Welsh, B., and Hunt, B. (1996). *Imaging Through Turbulence*. Laser & Optical Science & Technology. Taylor & Francis.
- [26] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.
- [27] Singh, M., Tiwary, U. S., and Young-Hoon, K. (2008). An adaptively accelerated lucy-richardson method for image deblurring. *EURASIP Journal on Advances in Signal Processing*, 2008.

- [28] Sánchez-Lasheras, F., Ordóñez, C., Roca, J., and de Cos Juez, F. (2019). Real-time tomographic reconstructor based on convolutional neural networks for solar observation. *Mathematical Methods in the Applied Sciences*, 43.
- [29] Soloviev, O., Thao Nguyen, H., Bezzubik, V., Belashenkov, N., Vdovin, G., and Verhaegen, M. (2020). Shack-Hartmann sensor as an imaging system with a phase diversity. *arXiv e-prints*, page arXiv:2011.00891.
- [30] Suárez Gómez, S., Gonzalez, C., Alonso, E., Santos, J., Rodríguez, M., Carballido-Landeira, J., Basden, A., and Osborn, J. (2018). *Improving Adaptive Optics Reconstructions with a Deep Learning Approach*, pages 74–83.
- [31] Swanson, R., Lamb, M., Correia, C., Sivanandam, S., and Kutulakos, K. (2018). Wavefront reconstruction and prediction with convolutional neural networks. In Close, L. M., Schreiber, L., and Schmidt, D., editors, *Adaptive Optics Systems VI*, volume 10703, pages 481 – 490. International Society for Optics and Photonics, SPIE.
- [32] Taylor, G. I. (1938). The spectrum of turbulence. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 164(919):476–490.
- [33] Varghese, J., Subhash, S., Subramaniam, K., and Sridhar, K. P. (2020). Adaptive gaussian notch filter for removing periodic noise from digital images. *IET Image Processing*, 14(8):1529–1538.
- [34] Verhaegen, M., Pozzi, P., Soloviev, O., Vdovin, G., and Wilding, D. (2017). *Control for High Resolution Imaging*. Delft University of Technology.
- [35] Welsh, B. M. and Gardner, C. S. (1989). Performance analysis of adaptive-optics systems using laser guide stars and sensors. *J. Opt. Soc. Am. A*, 6(12):1913–1923.
- [36] Wilding, D., Soloviev, O., Pozzi, P., Vdovin, G., and Verhaegen, M. (2017). Blind multi-frame deconvolution by tangential iterative projections (tip). *Opt. Express*, 25(26):32305–32322.
- [37] Xiong, Z., Chen, S., Dong, L., Yang, P., Xu, B., Zhao, W., Yu, X., Yang, K., Wang, X., and He, X. (2019). Analysis of correlation algorithms for Shack-Hartmann wavefront sensors. In Ma, X., Fan, B., Wan, Y., Russell, A., and Luo, X., editors, *9th International Symposium on Advanced Optical Manufacturing and Testing Technologies: Large Mirrors and Telescopes*, volume 10837, pages 184 – 191. International Society for Optics and Photonics, SPIE.
- [38] Zernike, F. (1934). Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica*, 1(7):689–704.
- [39] Zhao, H., Xia, J., Zhang, L., Yu, C., and Fan, X. (2019). Improved vector extrapolation based richardson-lucy algorithm used for wavefront coded imaging and experimental demonstration. *Appl. Opt.*, page 45.
- [40] Zhou, H., Zhang, L., Zhu, L., Bao, H., Guo, Y., Rao, X., Zhong, L., and Rao, C. (2016). Comparison of correlation algorithms with correlating Shack-Hartmann wavefront images. In Li, M., Jalali, B., Goda, K., and Tsia, K. K., editors, *Real-time Photonic Measurements, Data Management, and Processing II*, volume 10026, pages 196 – 207. International Society for Optics and Photonics, SPIE.

