Design and Simulation of a High Spatial Resolution Hartmann-Shack Wavefront Sensor

Otto Manneberg
Master’s Degree Project
Stockholm, Sweden 2005
Design and Simulation of a High Spatial Resolution Hartmann-Shack Wavefront Sensor

Otto Manneberg

Biomedical and X-ray Physics
Department of Physics
AlbaNova
Royal Institute of Technology
Stockholm, Sweden
2005
This Thesis summarizes the Diploma work by Otto Manneberg for the Master of Science degree in Engineering Physics. The work was performed during the fall and winter of 2004 at the Department of Physics, Royal Institute of Technology in Stockholm, Sweden. The tutor was Peter Unsbo and the examiner was Hans Hertz.

TRITA-FYS 2005:11
ISSN 0280-316X
ISRN KTH/FYS/--05:11--SE
© Otto Manneberg 2005
Abstract

This Diploma Thesis concerns the design and simulation of a Hartmann-Shack wavefront sensor with high spatial resolution, intended to be used in measuring wavefront aberrations in the eye of the gecko lizard *Tarentola chezaliae*. The project was carried out in collaboration with the Vision Group at Lund University as a part of their ongoing work on eye designs in nocturnal and diurnal geckos.

A Hartmann-Shack wavefront sensor is designed, using a superluminescent diode as light source. The emergent wavefront is corrected for the larger part of refractive errors using a Badal system and magnified. Light economy is also considered.

Several different diffraction-limited lenslet arrays for the Hartmann-Shack sensor are simulated using MATLAB, with the effects of photon quantum noise and ADC noise in the 12-bit, four megapixel digital camera accounted for in the simulation. The simulated sensors give between 2500 and 11500 measurement points over the pupil area, which is assumed to be circular with a diameter of 3 mm. It is found that it is feasible to build the sensor, and recommendations are given for the crucial and simulated components.
Acknowledgements

Warmest thanks to my tutor Peter Unsbo, without whom I would still be working on the code, and to Linda Lundström for persisting in making me explain things and helping me. Couldn’t have done it without you.

Many thanks also to everyone at BIOX; Hans Hertz for letting me do this, Per and Jessica for putting up with me, Per for help with images, Heide for the translations from German, everyone else for being great people to work with and Göran for knowing what light really is.

Thanks to the incredibly patient people around me: Mia, Peter, Jocke, mom, dad, Sarah and the e-hacked/idiotopia gang, and to all the people I’ve forgotten: Thank you!

Thanks also to Dr Geunyoung Yoon for the images and inspiration to images.

Caveat lector

A beam aimed straight into an unshielded eye might be dangerous and painful. Especially if it is a 2-by-4.
# Table of Contents

1. **Introduction** .................................................................................. 1
   1.1. **Measuring Wavefronts** ......................................................... 2
       1.1.1. Laser Ray Tracing ......................................................... 3
       1.1.2. The Spatially Resolved Refractometer .......................... 3
       1.1.3. The History of the Hartmann-Shack Sensor ................. 4
       1.1.4. The Function of the Hartmann-Shack Sensor .......... 5
   1.2. **The Gecko Eye and Longitudinal Chromatic Aberration (LCA)** 7
       1.2.1. Measuring Multifocality by Wavefront Sensing .......... 8

2. **Theory** ........................................................................................... 9
   2.1. **The Eye** ................................................................................. 9
       2.1.1. Basic Design .................................................................. 9
       2.1.2. Refractive Errors ......................................................... 11
       2.1.3. Seidel Aberrations ....................................................... 13
       2.1.4. Chromatic Aberration ................................................. 17
   2.2. **Fourier Optics** ..................................................................... 18
       2.2.1. The Fourier Transform ............................................... 18
       2.2.2. The Scalar Wave Equation and Fresnel-Huygens’ Principle 20
       2.2.3. The Fresnel Approximation ....................................... 23
       2.2.4. Phase Transform in a Perfect Lens ............................. 24
       2.2.5. Convolutions and the Optical Transfer Function .... 25
   2.3. **Data Analysis** ....................................................................... 28
       2.3.1. Fast Fourier Transforms ........................................... 28
       2.3.2. Zernike Polynomials .................................................. 30

3. **Designing the HS Sensor** .............................................................. 31
   3.1. **Overview** ........................................................................... 31
   3.2. **Components** ........................................................................ 33
       3.2.1. Light Source ................................................................ 33
       3.2.2. The Eye ..................................................................... 34
       3.2.3. The Optics and Pupil Camera ................................. 34
       3.2.4. The Camera .............................................................. 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.5. The Lenslet Array</td>
<td>36</td>
</tr>
<tr>
<td>3.3. Light Economy</td>
<td>37</td>
</tr>
<tr>
<td>4. Simulations in Matlab</td>
<td></td>
</tr>
<tr>
<td>4.1. The Simulated System</td>
<td>41</td>
</tr>
<tr>
<td>4.1.1. Simulating psfs and HS pattern</td>
<td>43</td>
</tr>
<tr>
<td>4.1.2. Creating and Re-creating Wavefronts</td>
<td>48</td>
</tr>
<tr>
<td>4.2. The Simulations</td>
<td>49</td>
</tr>
<tr>
<td>4.3. Results of the Simulations</td>
<td>52</td>
</tr>
<tr>
<td>5. Conclusions and Recommendations</td>
<td>55</td>
</tr>
<tr>
<td>6. Outlook</td>
<td>56</td>
</tr>
<tr>
<td>7. References</td>
<td></td>
</tr>
<tr>
<td>7.1. References</td>
<td>57</td>
</tr>
<tr>
<td>7.2. Image Sources</td>
<td>59</td>
</tr>
<tr>
<td>8. Appendix</td>
<td></td>
</tr>
<tr>
<td>8.1. Matlab Code</td>
<td>60</td>
</tr>
</tbody>
</table>
1. Introduction

The discovery of multifocal eyes during the late twentieth century has given rise to an interest in adapting existing methods as well as developing better methods for measuring wavefronts and wavefront aberrations. Multifocal eyes are eyes with different focal properties in different areas of the pupil, and since this could be seen as an aberration, wavefront sensing is a possible way of investigating the phenomenon. The Vision Group in Lund [1] is one prominent group in animal vision science, and they were the first to present measurements of multifocal properties in some species.[2] These measurements have included some 200 measurement points over the diameter of one eye. This project concerns the design and simulation of a measurement apparatus which would give several thousand measurement points in one eye; a very high spatial resolution Hartmann-Shack (HS) sensor.

Section 1 deals with the background of the project and describes briefly some methods currently used to measure wavefronts. It also presents the goal of the project – measurements on living gecko lizards.

Section 2 contains the background theory of the project; the workings of an eye as well as some basic theory of aberrations, a very brief introduction to Fourier Optics, and a short description of the numerical methods used in the simulation.

Section 3 outlines the proposed design and the components that could be used in a realization of the HS sensor.

Section 4 describes the simulations made, with emphasis on how the specific numerical problems were solved. It also contains the results of the simulations and discussion of these.

The conclusions and recommendations based on the findings described in Sect. 4 can be found in Sect. 5, and an outlook into the possible future of the project is given in Sect. 6.
1.1 Measuring Wavefronts

This work concerns investigating a possible method for measuring the different focal lengths of a multi-focal eye, and thus we begin with a short description of wavefront sensing and why it is the method of choice in this case.

The simplest way to find the focal length of a thin positive lens would be to simply let parallel light impinge on it, and measure very carefully how far behind the lens the focus is situated. To do the same kind of measurement on a thick lens, or a system of lenses, is slightly more complicated, since the principal planes of the lens (system) must somehow be located. To do the same measurement on a lens with different focal lengths at different locations in the lens is harder, and when the lens is in the eye of a living creature, it quickly becomes impossible.

Another way of measuring would be to place a point source of light at different distances from the lens and find the point-to-lens distance which makes the refracted light parallel. This method could also be used to find any aberrations, deviations from a theoretical “perfect system”, in the lens if the direction of the light could somehow be measured, since light passing this part of the lens would be refracted in a slightly different way. The information about in which direction light is traveling lies in the phase of the light-wave, and since a wavefront is a thought-up surface connecting points of constant phase, being able to measure (or “sense”) the shape of a wavefront would enable us to measure aberrations.

There are several different methods of wavefront sensing. Three common methods are Laser Ray Tracing, using a Hartmann-Shack sensor (or Shack-Hartmann, depending on personal preferences) and Spatially Resolved Refractometry. This thesis concerns the simulation and design of a Hartmann-Shack sensor, so the other two will only be described very briefly. Please note that all of these actually measure the slope, or gradient, of the wavefront. The reconstruction of the actual wavefront from this data is done by a computer in one of several ways. The interested reader can refer to [3] or [4].
1.1.1 Laser Ray Tracing

Laser Ray Tracing (LRT) is an objective technique based on direct measurements of ray deviations on the retina. Two rays of light are sent into the eye, as seen in Fig. 1.1. One of these, the dotted one in the figure, is sent through the center of the pupil and serves as a reference beam. Note that the mechanism for creating the reference beam is not shown in the figure. The entrance location of the other beam can be varied as a means to scan the eye. If the optics of the eye were free from all aberrations, these rays would always converge to a spot on the retina. As this is not the case, they will instead be refracted to different points on the retina. These points are imaged onto a CCD detector outside the eye, and the distance between their images then can be used to measure the local tilt of the wavefront after passing the optics of the eye at the entrance location of the measurement beam.

![Fig. 1.1. The Laser Ray Tracing arrangement](image)

1.1.2 The Spatially Resolved Refractometer

The Spatially Resolved Refractometer, SRR, is an instrument built on the same principle as LRT, but differs from it in one fundamental way as it is a subjective method; the patient must participate actively in the measurement process. As in LRT, two parallel, narrow beams of light are sent into the eye, one of them serving as a reference. The patient is given means to control the tilt of the measurement beam, and is told to do so until the she only sees one spot, meaning that the spots coincide on the retina (Fig. 1.2). The angle with which the patient has changed the tilt can be shown to be the local tilt angle of a plane wave passed through the optics of the eye, again at the entrance point of the measurement beam.
1.1.3 The History of the Hartmann-Shack Sensor

The Hartmann-Shack sensor [5] is a measurement instrument with a long history, at least if we acknowledge the Scheiner disc as its ancestor. In 1619, the Jesuit philosopher Christoph Scheiner published a work called *Oculus, sive Fundamentum Opticum* (roughly “The Eye, and hence a Ground for Optics”) in which he described a simple way to measure the refractive power of an eye. He let light from a distant point source reach the eye through a disc with two pinholes drilled in it, and presented the idea that an eye without any refractive errors would see only one spot, but an imperfect eye would see two. This, as might be guessed, was also the predecessor of LRT and SRR.

In 1900, Hartmann published an article in which he described a method of measuring wave aberrations by letting the light impinge on a screen with an array of pinhole openings. A flat wavefront would yield a bunch of parallel rays, and an aberrated wavefront would cause some rays to diverge from the bunch. This method was greatly improved by Shack and Platt, who instead of an array of pinholes used an array of small lenses. This was the invention of Shack’s Modified Hartmann Screen, or Shack-Hartmann for short.

The Shack-Hartmann wavefront sensor mainly found use in astronomical instruments, where measuring the shape of a wavefront from a celestial object and comparing it to a reference wavefront allowed astronomers to correct for atmospherically induced aberrations for the first time.
When the first article on using this method to measure aberrations in a human eye was published, the author referred to the method as the “Hartmann-Shack”, something which besides having led to numerous arguments about what the instrument is “really called” has led to astronomers referring to it as “Shack-Hartmann” and people in Vision Science referring to it as Hartmann-Shack. Since this work is much closer to Vision Science, The name Hartmann-Shack sensor will be used.

1.1.4 The Function of a Hartmann-Shack sensor

The Hartmann-Shack sensor (HS sensor) is, just like LRT, an objective method for determining a wavefront. In principle, the arrangement is fairly simple: An array of identical small positive lenses, often called lenslets because of their size, is placed at a distance equal to their focal length from a CCD detector.

When a wavefront reaches the lenslet array, the array will divide it into discrete sections, one section for each lenslet. If we now assume the lenslets to be small compared to the curvature of the wavefront, the wavefront can be assumed to be flat over each lenslet. Therefore, each lenslet will focus its part of the incoming wavefront to a spot on the CCD. If the entire incoming wavefront is flat, the spots on the CCD will be directly behind the lens centra, and make up an orderly pattern depending only on the geometrical packing of the lenslets, which normally are either square or hexagonal.

If the incoming wavefront is curved in some fashion, the light at a single lenslet will be impinging at an angle, but still be parallel over the lenslet according to our previous assumption of small lenslets. This means that it will still give rise to a spot on the CCD, but the spot will have moved relatively to its location with a flat wavefront. This displacement is measured, and used to find the local tilt of the wavefront (Fig. 1.3).

![Fig. 1.3. The Hartmann-Shack sensor arrangement](image)
All the local tilts are then pasted together, and different numerical methods are employed to interpolate between the measurement points to yield a smooth measured wavefront. Naturally, there are limitations to this technique. If the wavefront is too curved, the spots will become blurred due to defocus and the aberrations discussed in Sect. 2.1.3, and the spots might be displaced so much that they overlap or switch places with each other. A constant high curvature can be compensated for by placing a lens of known power in front of the lenslet array, thus canceling the curvature. If this method is used, it is of course vital that the lens does not introduce measurable aberrations to the wavefront.

A second problem is light economy; if a high number of very small lenslets are used, the amount of light that passes through each lenslet might become so small that the CCD has troubles detecting the spot or that noise becomes a serious problem.
1.2 The Gecko Eye and Longitudinal Chromatic Aberration (LCA)

To humans as well as many other animals, vision is of paramount importance in daily life. Total or partial loss of vision, or degradation in visual acuity, often has great implications for the individual, perhaps with the exception of human eyes and errors that can be corrected for by use of spectacles or contact lenses. Thus it is no wonder that the human eye is the sensory organ upon which the most research has been conducted. Biologists have also conducted extensive research on animal eyes and comparative studies have been made en masse.

A more detailed description of the eye is given in Sect. 2.1.1, but for now it will suffice to know that all eyes considered herein work on the principle of a lens imaging the object of scrutiny onto a light-sensitive surface called retina. If the lens is somehow defective, of the wrong power, or at the wrong distance from the retina, a blurred image results. The details of such refractive errors and aberrations are discussed in more detail in Sects. 2.1.2-4.

One of these aberrations, longitudinal chromatic aberration or LCA, has the effect that a given homogenous lens will focus light of different wavelengths to points at different distances from the lens. The image blurring effects of this phenomenon is larger the shorter the depth of field is. The depth of field, in turn, depends on the f-number – the quotient between the focal length and diameter of a lens – so that a low f-number gives a short depth of field. For a human eye, the f-number is about 6.5 in good lightning conditions. For certain other animals, this ratio approaches unity. One group of such animals are nocturnal geckos, for which the LCA should be of such proportions that the gecko would have very poor eyesight, due to colored blurring of objects – it would only be able to focus in one color at a time. However, an investigation of the retina of a nocturnal gecko reveals light receptors for three different wavelengths, and the small reptilians can be shown to have greater visual acuity than LCA would allow them. Similar phenomena occur in the eyes of a number of fish species and other animals.

Kröger et al have shown that these animals tend to have multifocal lenses – lenses with annular zones of different focal lengths.[6] The multifocality has been shown to correct for some (about 30% in a fish lens[2]) of the LCA predicted by theory. Such multifocal eyes are in fact found in many animals that have slit-shaped pupils in good lightning conditions – the slit shape is hypothesized to enable use of the entire range of zones even when the pupil contracts. Examples of such animals are cats and horses.
1.2.1 Measuring Multifocality by Wavefront Sensing

As mentioned above, studies have already been made of LCA in animal eyes and multifocal compensation thereof. For a more thorough discussion of the studies and their findings, please consult the extensive list of references in [2] and [6].

It can be shown that the amount of LCA present in an eye does not affect the longitudinal spherical aberration (LSA), an aberration that occurs in monochromatic light and means that the focal length of the lens is dependent on a radial coordinate.[2] Knowing this, the LCA for different wavelengths can be calculated from measurements made with monochromatic light if only the dependency of refractive index upon wavelength (the dispersive properties) of the lens medium is known. From this, it is also possible to deduce which areas of the lens that are used to focus a given wavelength onto the retina. Measurements of LCA have been carried out for, e.g., the cichlid fish *Haplochromis burtoni*.[2]

Measurements also show that multifocality is present in nocturnal geckos but not in diurnal geckos, something which is of great interest to evolutionary biologists: The fact that there nocturnal and diurnal gecko species which are very closely related makes the gecko eye important in understanding the evolution of different types of eyes. This Thesis describes a non-invasive method, useable on living geckos and fairly easily adapted to other eyes, to measure the multifocal properties of the gecko eye.
2. Theory

In this section, the theoretical fundament upon which this thesis rests will be outlined. More important parts will be more carefully explored. First, the basic principles of an eye are discussed, followed by a brief description of some basic aberrations. For a more complete description of Visual Optics, see [7]. This is followed by an introduction to Fourier Optics, with emphasis on the Fourier transforming properties of a lens and on the optical transfer function. A much more thorough theory of diffraction and Fourier Optics can also be found in references in these sections. Concluding this section is a short overview of the most important functions used in Hartmann-Shack pattern interpreting, the Zernike polynomials.

2.1 The Eye

This section describes the basic functions of an eye, and some theory of aberrations in optical systems.

2.1.1 Basic Design

All eyes, regardless of whether it is a many-faceted insectile eye, a camera obscura-like lensless pinhole, such as the infrared-sensitive pits of some snakes, or the more familiar arrangement with a lens giving a real image on a retina, basically serve the same purpose: To gather information from an animal’s surroundings and encode this information for further processing. The encoding as well as the signal relaying and transport and the processing of the information are all fascinating parts of Vision Science, but this thesis will be limited to the first step – the gathering of information in the form of light entering the eye.

The kind of eye that consists of a number of refracting surfaces which together form a real image on the retina is present in a great variety of animals, such as octopi, certain spiders, and most vertebrates.[8] Although the details may vary quite a bit, the basic framework stays the same and can be represented by the human eye (Fig. 2.1) in this section.

![Fig. 2.1. Schematic of the human eye](i2)
The light entering the eye is refracted each time it crosses a boundary between two different media. The amount with which the light is refracted depends on the curvature of the boundary, as well as the indices of refraction of the involved media. The optics of the eye - the cornea, aqueous humor, crystalline lens and vitreous humor in a human eye - work together to form a real, minified, image of the object of scrutiny.

Under normal circumstances, assuming an emmetropic (neither far- nor nearsighted) eye, this image will be formed on the retina, a thin layer of light sensitive cells where the light is absorbed via electrochemical processes and the information therein encoded for further processing. This processing takes place in several steps and involves sorting out the relevant bits of the information and reacting on it.

In many animals the distances between the different components of the eye is fixed, which means that in order to be able to see both distant and nearby objects clearly, at least one component of the eye has to able to change its refractive power. In humans, this component is the crystalline lens, which can be stretched and flattened by the relaxation of the ciliary muscles. This process of re-focusing the eye is known as accommodating; the unaccommodated eye has a focal length such that it images distant objects onto the retina, as seen in Fig. 2.2. Other methods of accommodation include changing the refractive power of both lens and cornea (birds) and moving the lens with respect to the retina, as many fish do.

![Relaxed eye, object at infinity](image1)

**Fig. 2.2. Accommodating in a human eye [i3]**
Naturally, the eye cannot accommodate to image arbitrarily close objects, something which is easily verified by simply covering one eye, focusing on an object held at arm’s length, and slowly bringing the object closer to the eye whilst keeping the object in focus. At a certain distance, the object will start to appear blurred no matter how hard one tries to focus on it. The closest point the eye is able to accommodate to is called the near point of the eye, and is most often taken to be situated 25 cm or 10 inches from the eye depending on the choice of unit system. Incidentally, the point at which an object gives an image on the retina of an unaccommodated eye is called the far point of the eye. For an emmetropic eye, the far point is situated at infinity.

2.1.2 Refractive Errors

Like any other part of a living organism, eyes are never completely “perfect”. Many different factors combine in the image forming process and therefore many things can occur which degrade the image quality. For humans, the most important of these are refractive errors, of which there are three: Hyperopia, myopia and astigmatism. All of these can be derived in paraxial or Gaussian theory, which is based on the assumption that all rays of light in the optical system at hand form small angles with the optical axis, thus justifying the approximation \( \sin x \approx x \) for these angles. The first two, hyperopia and myopia, are perhaps better known as far-sightedness and near-sightedness, respectively, and relate to the total power of the optics of the eye, while the latter is an error due to asymmetric properties of the eye’s optics. For certain other animals other aberrations seem to limit the image forming process, as we shall see further on. A more thorough description of both the refractive errors and the Seidel aberrations described in Sect. 2.1.3 can be found in, e.g., [8].

Hyperopia

Hyperopia, far-sightedness, is a condition most often caused by the eye being physically “too short”. When totally relaxed, the lens forms images of distant objects onto a plane behind the retina. Since the distance between the lens and the retina is fixed, this means that a hyperopic eye must accommodate to be able to see distant objects. Another consequence of hyperopia is that the near point of the eye is at a greater distance than the near point of an emmetropic eye, that is, the hyperope cannot see nearby objects clearly - hence the term “far-sightedness” (Fig. 2.3). The simplest way of correcting for this refractive error is to put a positive lens in front of the eye.

Fig. 2.3. The hyperopic eye [i3]
**Myopia**
More informally known as “near-sightedness”, myopia is in a way the opposite of hyperopia; the eye forms the image of a distant object in a plane between the lens and the retina, thus placing the far point of the eye at a distance less than infinity (where it is situated for an emmetropic eye). All objects beyond this point will appear blurred, as shown in Fig. 2.4. Note that the figure is in no way drawn to scale, neither with Fig. 2.3 or itself, with respect to the distances between objects and eyes. Myopia is corrected for by placing a negative lens in front of the eye so that the eye looks at the virtual image formed by the lens.

![Fig. 2.4. The myopic eye](image)

**Astigmatism**
Contrary to hyperopia and myopia, astigmatism is an error that is present to some degree in most human eyes. This error stems from asymmetries in the eye – and almost no eye is perfectly rotationally symmetric. These asymmetries give the eye different powers along different meridians (directions in the lens, perpendicular to the axis of the system), so that an object consisting of two sets of parallel lines intersecting at an angle will not necessarily have both sets in focus at the same time. If the directions of highest and lowest power are perpendicular to each other, the astigmatism is termed regular, and can be corrected by use of contact lenses or glasses. Note that astigmatism comes in a great deal of varieties – besides variations in direction of the axes of highest and lowest power, the eye can be hyperopic, emmetropic or myopic along the axes.
2.1.3 Seidel Aberrations

As was implied above, there are several aberrations other than the three derivable in paraxial theory; if we include one additional term in the series expansion of the sine function, and let \( \sin x \approx x - \frac{1}{6} x^3 \), we can derive five new aberrations, often known as the Seidel aberrations after Ludwig von Seidel, who was the first to publish a comprehensive study of them in 1857. Of course, we could continue from third order to fifth order in the sine - or even seventh order - to find what is even more aberrations, but the calculations quickly become very cumbersome, and we have passed far beyond the point where it is much easier to use one of the several computer programs commercially available for solving optical problems of this kind.

The Seidel aberrations are five, and they are all monochromatic aberrations; they occur even if the light passing through the optical system is monochromatic. Of the aberrations described below, the first three ones – spherical aberration, coma and third order astigmatism – degrade the image, making it unclear in such a way that information present in the object may be lost in the image. The last two, distortion and field curvature, deform the image without changing the information content.

**Spherical Aberration**

Spherical aberration differs from the other kinds of Seidel aberrations in one central aspect (no pun intended), as it is the only one that exists for an on-axis object point. While the other four exist only for object points that are off-axis, spherical aberration is simply always there, although sometimes in so small an amount that it is completely negligible. In effect, spherical aberration means that the refractive surface has a power dependent on the distance from the optical axis. Light rays hitting the central part of the lens will not experience the same focal length as rays hitting the periphery, which means that there is no focal point anymore (Fig. 2.5). Instead, there is a place where the light is less spread out than anywhere else – the circle of least confusion.

![Circle of least confusion](image)

Fig. 2.5. Spherical aberration in a lens [i1]
For an eye, spherical aberration is not much of a problem in good lighting conditions. The pupil contracts, effectively stopping down the pupil to a size where spherical aberration becomes a small problem compared to any refractive errors, irregularities in the cornea, diffraction and higher order aberrations.

**Coma**

Also known as comatic aberration (from Greek *kome*; hair, and *koman*, to wear long hair) coma is an aberration with effects akin to those of spherical aberration. Unlike spherical aberration, however, coma only occurs for off-axis image points. The light from an off-axis point source will be smeared out into a shape much like a stylized drop or a cone with the point either inwards or outwards, depending on the lens. Figure 2.6 depicts a lens with comatic aberration.

![Fig. 2.6. Comatic aberration in a lens](image)

Coma occurs when the magnification in a lens or optical system is dependant on location in the lens. Modern optical instruments are often well corrected for coma, but in the eye it is one of the reasons that objects in peripheral vision give blurred images on the retina.
Third Order Astigmatism
While giving almost the same effects as its first order namesake, third order astigmatism, or oblique astigmatism as it is also known, has a different cause. While the first order astigmatism is due to cylindrical asymmetry in the lens, oblique astigmatism is present also in a perfectly symmetric lens as long as the object point is off-axis. To better understand this effect, we envision two planes, as in Fig. 2.7: The meridional plane, which is the plane containing the optical axis and the ray through the center of the lens (the chief ray), and the sagittal plane which contains the same ray and is perpendicular to the meridional plane.

![Fig. 2.7. Third order astigmatism and the meridional and sagittal planes](image)

The rays in the different planes will experience different refractive powers, and thus be focused at different distances from the lens. In between these two foci, which will be lines rather than dots, is the circle of least confusion (Fig. 2.7).
Distortion
Unlike the other third-order aberrations, distortion is something that most people have seen the effects of in the “fish-eye” lenses mounted in doors. Distortion occurs if the magnification differs for different points in the object. Depending on the optical system, the distortion can be either “pinchusion” or “barrel”, as shown in Fig. 2.8 below. As has been stated before, distortion is an aberration that deforms the image without changing its information content – no “blurring” takes place.

Field Curvature
By now, the reader might have guessed that most things that in paraxial optics are called something ending in “plane” in fact aren’t – it is most often a curved surface that can be approximated with a plane in the region close to the optical axis. This is also the case with the image plane. The surface on which the image will be completely in focus is, in reality, curved towards the lens for positive lenses, as seen in Fig. 2.9 below, and away from the lens for negative lenses. This aberration does not present much of a problem in the eye, since the retina is also curved. Like distortion, field curvature does not mean loss of information in the image.
2.1.4 Chromatic Aberration

The aberrations discussed so far have all been monochromatic, meaning that they occur for light of a single wavelength. Now, the animal eye very seldom operates in monochromatic light but rather in the polychromatic light from the sun, or in the case of humans, different types of lamps. This brings us up to chromatic aberration – the fact that a lens has different powers for different wavelengths. This phenomenon is a result of dispersion, which means that the index of refraction \( n \) for a substance is not constant as the wavelength of the light is varied. The dependence of \( n \) on the frequency of the light can be summarized in a dispersion equation, often giving \( n \) as a function of \( \omega \), the angular frequency. Such an equation can be fairly complicated, but in frequency domains far from absorption wavelengths of the material, \( n \) is a strictly increasing function of frequency. This is what is responsible for a prism’s ability to split light into its constituent wavelengths as well as a fundamental part in the forming of a rainbow. Notwithstanding its crucial role in forming pretty patterns of color, dispersion is a problem for anyone who wants to design an optical system, for the very same reason. The fact that different wavelengths experience different indices of refraction also means that they experience different refractive powers in a lens – light of different colors are focused to points at different distances from the lens. This is known as longitudinal chromatic aberration, or LCA, and is depicted in Fig. 2.10 below.

![Fig. 2.10. Longitudinal chromatic aberration in a lens](image)

In modern optical systems, this is compensated for mainly by the use of achromatic lenses, lenses that are made up of two or more lens elements whose chromatic aberrations negate each other. In an animal eye this is obviously not an option, and chromatic aberration should logically pose a problem, more so for animals with short eyes and large pupils, since the depth of focus of an eye (if we disregard all effects of photoreceptor density on the retina) decreases as the quotient between the focal length and pupil diameter – the f-number – of the eye decreases.
2.2 Fourier Optics

This section deals with the propagation and diffraction of light, and the use of Fourier methods to describe an optical system. An expression for the field distribution in the focal plane of a diffraction-limited lens (one where aberrations are negligible) is derived, and some basic theory of optical transfer functions is outlined. For a much more thorough treatment of the subject, the reader is urged to refer to [9] or [10]. This section will demand some knowledge of mathematics from the reader, mainly calculus in two variables and some knowledge of the Fourier transform, the basic theorems of Fourier Analysis and the Dirac delta function. The latter can be gained from, e.g., [11].

2.2.1 The Fourier Transform

To refresh the reader’s memory we begin by defining the (two-dimensional) Fourier transform, or frequency spectrum, of a complex-valued function \( g(x,y) \) as

\[
G(f_x, f_y) = \mathcal{F}\{g(x,y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) e^{-2\pi i(f_x x + f_y y)} \, dx \, dy \tag{2.1}
\]

where the operator \( \mathcal{F} \) has been introduced for reasons of convenience. The transform \( G \) of \( g \) is thus itself a (generally complex-valued) function of the two variables \( f_x \) and \( f_y \), which will henceforth be referred to as spatial frequencies. In complete analogue, we define the inverse Fourier transform as

\[
\mathcal{F}^{-1}\{G(f_x, f_y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(f_x, f_y) e^{2\pi i(f_x x + f_y y)} \, df_x \, df_y . \tag{2.2}
\]

This means that \( \mathcal{F}^{-1}\mathcal{F}\{g\} = g \), which is what we would expect from something dubbed “inverse”. The mathematically inclined now perhaps wonders about the existence of the transform and its inverse. It has been pointed out that “physical possibility” is in fact a sufficient condition for the existence of the transform.[12] For more mathematically stringent existence conditions, consult a book on Fourier Analysis such as the previously mentioned.

Some basic properties of the Fourier transform are listed below (uppercase letters designate transform of lowercase) together with their most common names. Proof of these can be found in the literature mentioned above.
Linearity:
\[ \mathcal{F}\{\alpha f + \beta g\} = \alpha \mathcal{F}\{f\} + \beta \mathcal{F}\{g\} \quad \alpha, \beta \in \mathbb{C} \]  
(2.3)

The Similarity Theorem:
\[ \mathcal{F}\{g(ax, by)\} = |ab|^{-1} F\left(\frac{f_x}{a}, \frac{f_y}{b}\right) \quad a, b \in \mathbb{R} \setminus \{0\} \]  
(2.4)

The Shift Theorem:
\[ \mathcal{F}\{g(x - x_0, y - y_0)\} = e^{-2\pi i(f_x x_0 + f_y y_0)} G(f_x, f_y) \]  
(2.5)

The Convolution Theorem:
\[ \mathcal{F}\left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \cdot h(x - \xi, y - \eta) \, d\xi \, d\eta \right\} \equiv \]  
\[ \mathcal{F}\{g \ast h\} = G(f_x, f_y) \cdot H(f_x, f_y) \]  
(2.6)

The notation \( g, h \) from the last theorem will be used instead of the cumbersome convolution integral in the remainder of this text. Note that Eq. 2.6 together with the definitions of the transform and its inverse also gives that \( \mathcal{F}\{g(x, y) \cdot h(x, y)\} = G(f_x, f_y) \ast H(f_x, f_y) \), and that a convolution of any function \( h(x, y) \) with the Dirac delta function corresponds to multiplication by unity in the Fourier domain, and thus returns \( h(x, y) \).
2.2.2 The Scalar Wave Equation and Fresnel-Huygens’ Principle

The fact that light in most cases can successfully be described as an electromagnetic wave has been known for more than a century. A wave equation for the electric and magnetic field vectors \( E \) and \( B \) can be derived from Maxwell’s equations, which in SI units and in the absence of free charge and currents are

\[
\nabla \times E(r, t) = -\frac{\partial B(r, t)}{\partial t}
\]

\[
\nabla \times B(r, t) = -\mu \varepsilon \frac{\partial E(r, t)}{\partial t}
\]

\[
\nabla \cdot E = \nabla \cdot B = 0
\]

Now, applying the operator \( \nabla \times \) to the first equation and switching the order of time and space derivatives yields

\[
\nabla \times (\nabla \times E) = -\frac{\partial}{\partial t} (\nabla \times E) \Rightarrow \nabla (\nabla \cdot E) - \nabla^2 E = -\mu \varepsilon \frac{\partial^2 E}{\partial t^2}
\]

or, rearranging the terms to make it look familiar and letting

\[
\n n = \left( \frac{\varepsilon \mu}{\varepsilon_0 \mu_0} \right)^{1/2} \text{ and } c = \left( \frac{\varepsilon_0 \mu_0}{\varepsilon \mu} \right)^{1/2}, \text{ we arrive at the vector wave equation:}
\]

\[
\n\nabla^2 E - \frac{n^2}{c^2} \frac{\partial^2 E}{\partial t^2} = 0
\]

(2.7)

To end up with an equation for \( B \), simply go through the same steps starting with the equation for \( \nabla \times B \). Now, the interesting thing here is that all the components of both vectors satisfy the very same equation, and we can actually describe both fields by giving the scalar wave equation
\[ \nabla^2 u - \frac{n^2}{c^2} \frac{\partial^2 u}{\partial t^2} = 0 \]  
(2.8)

where \( u \) is one of \( E_x, E_y, E_z, B_x, B_y, \) and \( B_z \). This is of course not absolute truth (physics never is), as we have tacitly made a few assumptions regarding the medium in which the wave propagates. Assuming the medium to be nonmagnetic \( (\mu = \mu_0) \) as all media considered here will be, we have also assumed it to be

- linear – \( n \) is independent of the light intensity,
- isotropic – \( n \) is independent of the polarization of the light,
- homogenous – \( n \) is constant throughout each medium through which the light propagates,
- nondispersive – \( n \) is independent of the wavelength of the light, at least in the region of wavelengths occupied by the light in question.

Though it might seem like we are severely restricting our description by the numerous assumptions, our theory describes the observable reality very well. The only thing left to us now is to solve the wave equation, something which might prove no mean feat. The different methods of solving PDEs are numerous, and beyond the scope of this work. For our purposes, it will suffice to know that for a very interesting case (diffraction of light by an aperture in a plane screen) a solution using Green’s functions will work.\[9,13\] Before we proceed, we shall introduce complex notation for the field \( u \).

Let us assume monochromatic light, and let

\[ u(r,t) = \text{Re}\left\{ U(r) \cdot e^{-2\pi i \nu t} \right\} \]  
(2.9)

where \( U(r) = A(r) \cdot e^{-i\phi(r)} \) and \( \nu \) is the (temporal) frequency of the light. This means that the field \( u \) is fully characterized by \( U \) alone, since the time-dependence is given by Eq. 2.9. Using the scalar wave equation (Eq. 2.8), we now get the Helmholtz equation in \( U \):

\[ \left( \nabla^2 + k^2 \right) U = 0 \]  
(2.10)

where \( k = \frac{2\pi}{\lambda} = \frac{2\pi n\nu}{c} \) is the wave number of the light. This equation can now be attacked in different ways, but as previously stated, Green’s functions will yield manageable albeit not trivial expressions. It can be shown that the field at a point \( (u,v) \) a distance \( z \) (see fig 2.11) from the diffracting aperture can be written...
\[
U(u,v) = \frac{z}{i\lambda} \iiint_U \frac{e^{ikr}}{r^2} dxdy
\]  

(2.11)

Where \( r \), as seen in the figure below, is given by \( r = \sqrt{z^2 + (u - x)^2 + (v - y)^2} \). Of course, the assumptions made earlier must still hold, and for Eq. 2.11 to be valid, we also need the distance between the source point and the point of observation to be much greater than a wavelength, or \( r \gg \lambda \). Equation 2.11 is one way to state the Fresnel-Huygens principle, which in a shortened version says that each point on a wavefront acts a source of “secondary wavlets” of spherical shape, and that the envelope of these will constitute the wavefront at the next instant.

It would now seem that our problem has been solved, and all we need to do is to numerically evaluate the integral in Eq. 2.11. This is in a way the case, but actually evaluating the integral often proves to be very hard, or at least very time-consuming, indeed. The solution is to keep approximating.

Fig. 2.11. Diffraction geometry for Eq 2.11. The gray area represents an infinite opaque screen in the xy-plane.
2.2.3 The Fresnel Approximation

In our search for a more manageable expression, we shall now turn our interest to the distance \( r \) in Eq. 2.11 above. Factoring out a \( z \), and using the binomial expansion for \((1 + \varepsilon)^{1/2}\) when \( \varepsilon < 1 \) and keeping the first two terms of the expansion yields

\[
r \approx z \left( 1 + \frac{1}{2} \left( \frac{u - x}{z} \right)^2 + \frac{1}{2} \left( \frac{v - y}{z} \right)^2 \right).
\]

(2.12)

As we see, our integral in Eq. 2.11 would be much simplified if we could neglect the second term in the expansion, and simply let \( r \approx z \). This is a feasible plan for the \( r^2 \) in the denominator, but not for the \( r \) in the exponential. One way of explaining this is by considering the fact that the \( r \) in the exponential is multiplied by \( k \), which most often is a comparatively large number (around \( 10^7 \text{ m}^{-1} \) for visible light), so any error in \( r \) has a large impact on the resulting exponential. Thus, keeping the two first terms in the exponential, and only the first in the denominator, we get

\[
U(u,v) = \frac{e^{ikz}}{i\lambda z} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(x,y) e^{i(\frac{(u-x)^2}{2z} + (v-y)^2)} \, dx \, dy
\]

(2.13)

where we can let the integrals run over the entire xy-plane by letting \( U(x,y) \) be zero outside the aperture. By algebraically manipulating Eq. 2.13, we see that it can be rewritten as

\[
U(u,v) = \frac{e^{ikz}}{i\lambda z} e^{i\frac{k}{2z}(u^2 + v^2)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(x,y) e^{i\frac{k}{2z}(x^2 + y^2)} \cdot e^{\frac{2\pi i}{\lambda z} (ux + vy)} \, dx \, dy
\]

(2.14)

which is, apart from the exponentials outside the integral, the Fourier transform of a function consisting of the field in the aperture multiplied by a quadratic phase factor. The Eqs.s 2.13 and 2.14 are both referred to as Fresnel diffraction integrals, and Eq. 2.14 shall be our starting point when we approach our description of diffraction in a thin lens.

For a discussion of when the Fresnel diffraction integral gives an accurate description of reality, the reader is urged to consult [9].
2.2.4 Phase Transform in a Perfect Lens

The by far most common element in an optical system is the lens. From a wave-optic point of view, a lens is a component which alters the phase of the light, since it is made of a medium with a different index of refraction than its surrounding. Thus, it can be shown that the effect of a lens with the z-axis as its optical axis on an impinging (complex) field $U(x,y)$ can be described by multiplication by a complex function $t(x,y)$, which describes the thickness variations and medium of the lens.[9 pp96-99] The field directly after passage through a thin, aberration-free lens can then be written as

$$U_{after}(x, y) = e^{-\frac{ik}{2f}\left(x^2 + y^2\right)} U_{before}(x, y)$$ \hspace{1cm} (2.15)

where $f$ is the focal length of the lens. Note that for this expression to be valid, we must make the same small-angle approximation as we do in first-order theory in geometrical optics (cf. Sect. 2.1).

Let us now define a pupil shape function $p(x,y)$ as being zero outside the pupil and unity inside, and the pupil function $P(x,y)$ as

$$P(x, y) = p(x, y) U_{before}(x, y).$$ \hspace{1cm} (2.16)

To arrive at an expression for the field distribution in the focal plane of the lens, we shall use the Fresnel diffraction integral in the form given in Eq. 2.14. Denoting the field in the focal plane $U_f$ and letting $z=f$, we get

$$U_f(u, v) = e^{ikf} \frac{e^{\frac{ik}{2f}(u^2 + v^2)}}{i\lambda f} \times$$

$$\int\int_{-\infty}^{\infty} \int\int_{-\infty}^{\infty} U_{before}(x, y) p(x, y) e^{\frac{ik}{2f}\left(x^2 + y^2\right)} e^{\frac{ik}{\lambda f}\left(ux + vy\right)} dx dy$$ \hspace{1cm} (2.17)

The very first exponential in the expression is a constant phase factor, and can be dropped. The phase factor in $t(x,y)$ is seen to precisely cancel the one inherent in the Fresnel diffraction integral, leaving us with (noting that $p=0$ outside the lens)
Thus we have shown that the field distribution in the focal plane of the lens is in fact the Fourier transform of the field before the lens multiplied by a quadratic phase factor. It is important to note that the complex field at the coordinates \((u,v)\) in the focal plane are determined solely by the amplitude and phase of the input at the spatial frequencies \((\lambda_x, \lambda_y)\). The intensity, which is the measurable entity, in the focal plane is now proportional to \(|U_f(u,v)|^2\). We shall hereafter disregard the proportionality constant, as this will affect only the light economy and noise level, and are incorporated into the simulations described in Sect. 4 when these things are taken into consideration.

Having reached this goal, we shall have a brief look at how convolutions can be used to describe optical systems.

### 2.2.5 Convolutions and the Optical Transfer Function

In the previous section, it was shown that the field distribution in the focal plane of a diffraction-limited lens is closely related to the Fourier transform of the incident field. An interesting special case of input is a plane wave, in which case the size and geometry of the lens decides the field distribution in the focal plane. Carrying out the transforms will show that, for a circular lens, the intensity distribution will be proportional to the square of a Bessel function of the first degree, and for a square lens it would be the square of the sinc function in two variables, the sinc function being defined as

\[
\text{sinc}(u) = \begin{cases} 
\frac{\sin(\pi u)}{\pi u} & u \neq 0 \\
1 & u = 0 
\end{cases}
\]  

\[
\text{sinc}(u, v) = \text{sinc}(u) \text{sinc}(v) 
\]

The plane wave case corresponds to the image of a point source placed in infinity. For any system, the intensity distribution in the image of a perfect point source is termed the intensity point spread function. Thus if the incident light is a plane wave, the intensity distribution in the focal plane will be the point spread function, which we shall refer to as \(psf\). We can now use Eq. 2.18 to find an expression for the \(psf\) of a perfect lens without aberrations mathematically, disregarding proportionality constants:

\[
U_f(u, v) = \frac{1}{2\pi f} \iint_{\text{lens}} U_{\text{before}}(x, y) e^{\frac{2\pi i}{\lambda y}(ux+vy)} dx dy.
\]  

(2.18)
\[
psf = \left| \frac{\partial}{\partial \lambda \gamma} \int_{\text{lens}} e^{\frac{-2\pi i (\omega x + \nu y)}{\lambda f}} \, dxdy \right|^2 = \left| \frac{\partial}{\partial \lambda \gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) e^{\frac{-2\pi i (\omega x + \nu y)}{\lambda f}} \, dxdy \right|^2
\]

(2.20)

In the same way, it would be possible to calculate the \textit{psf} on an entire optical system, if all the components are known. Now, we turn our interest to how the \textit{psf} can be used to calculate the field distribution for an arbitrary input into a system.

Let us begin by assuming an intensity distribution in the object plane \(I_o(x, y)\) and a system with a given point spread function \textit{psf} which, if we assume 1:1 imaging, can also be taken to be a function of \(x\) and \(y\). If the object was an intensity distribution of very small size spatially with the constant intensity \(I\) located at the origin, the distribution could mathematically be described by a Dirac delta function in this point, and the intensity in the image plane would simply be \(I \cdot \text{psf} (x, y)\) – the point spread function scaled with the intensity of the source. If the object consisted of a number of point sources of different intensities and at different locations, we would have

\[
I_o (x, y) = \sum_k I_k \delta(x - x_k, y - y_k) ,
\]

and the intensity \(I_i\) in the image plane would be a sum looking like

\[
I_i (x, y) = \sum_k I_k \text{psf} (x - x_k, y - y_k).
\]

(2.21)

Now, we can approximate any physically feasible continuous object function by an infinite sum of infinitesimally close delta functions. If we do this, and let the \((x_k, y_k)\) above become the continuous variables \((\xi, \eta)\) and let \(I_k\) become \(I_o(\xi, \eta)\), we get the integral expression

\[
I_i (x, y) = \int_{-\infty}^{\infty} I_o (\xi, \eta) \cdot \text{psf} (x - \xi, y - \eta) d\xi d\eta = I_o \ast \text{psf} ,
\]

(2.22)

which tells us that given an object function and the point spread function of the system, we can find the image function by convolution. Note that an ideal point spread function, a delta function, thus returns a perfect copy of the object. Since convolution corresponds to multiplication in the Fourier plane, it is often simple to carry out the calculations there. Fourier transforming both sides of Eq. 2.22 gives

\[
\hat{I}_i = \hat{I}_o \cdot \hat{\text{psf}} \equiv \hat{I}_o \cdot \text{OTF}
\]

(2.23)
where the hats denote Fourier transforms, and we have termed the Fourier transform of the \textit{psf} $OTF$, Optical Transfer Function. It is common practice to normalize the $OTF$ so that $OTF(0,0) = 1$, since this will preserve the energy contained in the signal. The theory of transfer functions can be taken much further, for a very nice treatment see [14]. The “trick” of calculating an image function by multiplication of the transform of the object function and the $OTF$ will be central to solving problems at hand in this project.
2.3 Data Analysis

Whilst the thoughts behind, and some details of, the programs written in this project can be found in Sect. 4, a brief description of the most mathematical parts of the programming is given here. As was shown in Sect. 2.2, Fourier methods can be used to calculate field and intensity distributions from given lenses. This section describes how MATLAB handles Fourier transforms, and also very briefly the most import function family that is used in the programs, Zernike polynomials.

2.3.1 Fast Fourier Transforms

The main difference between the analytical expression for the Fourier transform and the result as calculated by MATLAB is that the computer program does not handle continuous functions. Any continuous function must be approximated by a discrete set of data points before any further calculations can be done. This also means that the integrals above must be given as sums instead. For the two-dimensional case, which is most interesting to us, MATLAB uses the one-dimensional discrete Fourier transform twice, so only the one-dimensional will be considered here.[15] There is one new existence condition; as in the case of Fourier series, the sequence representing our function must be periodic. This can be solved by simply assuming that outside the interval on which it is defined, it repeats itself *ad infinitum*. As we shall see, this might cause some problems, but all of these can be solved. Thus we can begin by assuming that our function \( g(x) \) is approximated by the set of points \( g(x_n) \), and that this sequence has a period of \( N \), that is \( g(x_n) = g(x_n + N) \) where \( n, N \in \mathbb{Z} \). We can now define the discrete Fourier transform, or DFT, as

\[
DFT\{ g(x_n) \} = G(f_n) = \sum_{n=0}^{N-1} g(x_n) e^{-2\pi i f_n x_n} . \tag{2.23}
\]

In MATLAB, the calculation of a DFT is speeded up by using a method known as fast Fourier transform (FFT), the point of which is to iteratively decompose the problem into smaller subproblems until one of MATLAB’s several fixed codelets can handle the subproblems.[15] The algorithms in FFT are based on a algorithm library called FFTW, more about which can be found on the library homepage.[16] What we need to note here is that the FFT is considerably faster than using the definition of the DFT, especially so when the number of data points is a power of two.
It can be shown that in MATLAB the command “\texttt{fftshift(fft(fftshift(g)))}”, where \( g \) is a function sampled with an inter-sample distance of \( \Delta x \), will produce a discrete approximation of the function \( \frac{1}{\Delta x} \mathcal{F}\{g\} \). The “\texttt{fftshift}” part of the command is needed in order to make MATLAB display the FFT with the zero-frequency component in the center of the transform instead of in the edges (or corners, in two dimensions).

If FFT is used to calculate the Fourier transform in Eq. 2.18 it will, in analogy with the continuous case, image certain discrete spatial frequencies in the signal onto certain coordinate points in the data set of the FFT. The relation is the same as in the continuous case, so that the FFT of a function at coordinates \((u,v)\) are determined solely by the amplitude and phase of the spatial frequencies \((f_x = \frac{\nu}{\lambda f}, f_y = \frac{\nu}{\lambda f})\) in the input. Since FFT yields an output with the same number of data points as the input, this severely restricts which spatial frequencies will show up in the FFT, as we now shall see. The calculations will be carried out in one dimension, as the generalization to two dimensions is trivial. Let us assume that we have an input function \( g \), sampled at \( N \) equidistant points \( x_n \). Let the distance between two sample points be \( \Delta x \). A FFT of this data would result in a function \( G \), also sampled at \( N \) equidistant points \( \lambda \). The distance between the sampled frequencies can be shown to be \( \Delta f_x = \left( \frac{N \Delta x}{\lambda f} \right)^{-1} \), which gives us a step length in the focal plane of

\[
\Delta u = \Delta f_x \lambda f = \frac{\lambda f}{N \Delta x}.
\] (2.24)

This means that if we sample the input from \( x_{\text{min}} = -N/2 \cdot \Delta x \) to \( x_{\text{max}} = \left( N / 2 - 1 \right) \cdot \Delta x \), in order to make sure that we include the origin (cf Sect. 4), the FFT will be given over the spatial coordinate \( u \) from

\[
\begin{align*}
  u_{\text{min}} &= -\frac{N \Delta u}{2} = \frac{N \lambda f}{4 x_{\text{min}}} \quad \text{and} \\
  u_{\text{max}} &= u_{\text{min}} + N \Delta u = \frac{(N - 2) \lambda f}{4 x_{\text{max}}}.
\end{align*}
\] (2.25)

This restriction will be of great importance when programming the simulation of the HS sensor.
2.3.2 Zernike Polynomials

When expressing the shape of an arbitrary wavefront, it is easier to do so by projecting it onto an orthogonal basis. One such basis, which has become international standard in the treatment of aberrations in human eyes, is the Zernike polynomials.[17] These are a complete basis on the unit circle, which means that any sufficiently nice function over the unit circle can be decomposed into an (infinite) sum of Zernike polynomials. If we introduce polar coordinates over the unit circle by letting $x = r \cos \varphi$ and $y = r \sin \varphi$, the Zernike polynomials are characterized by the (non-negative integer) order $n$ of the radial polynomial and the azimuthal frequency $m$, which takes on the values $m = -n, -n + 2, -n + 4, \ldots, n - 2, n$. In polar coordinates, they are defined as follows ($N$ is a normalization constant):

$$
Z_n^m(r, \varphi) = \begin{cases} 
N_n^m \left( \sum_{k=0}^{\frac{1}{2}(n-|m|)} \frac{(-1)^k (n-k)!}{k! (\frac{1}{2}(n+|m|-k))! (\frac{1}{2}(n-|m|-k))!} r^{n-2k} \right) \cos(m\varphi) & m \geq 0 \\
-N_n^m \left( \sum_{k=0}^{\frac{1}{2}(n-|m|)} \frac{(-1)^k (n-k)!}{k! (\frac{1}{2}(n+|m|-k))! (\frac{1}{2}(n-|m|-k))!} r^{n-2k} \right) \sin(m\varphi) & m < 0 
\end{cases}
$$

where $N_n^m = \frac{2(n+1)}{\sqrt{1+\delta m0}}$ and $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

(2.26)

By ocular inspection, it is not evident that this is a practical choice. One reason of the Zernike polynomials’ popularity is the fact that the all refractive errors and Seidel aberrations mentioned in Sect. 2, as well as other more exotic aberrations, can be described in a simple fashion by Zernike coefficients – the coefficients of the different polynomials in a series expansion of a given function. The Zernike polynomials are used to construct and reconstruct wavefronts in this work. For more information on Zernike polynomials and wavefront reconstruction, the reader can refer to [3].
3 Designing the HS Sensor

This section presents the proposed general design of the HS sensor and auxiliary optics, with the prerequisite of sufficient light economy even with a very high spatial resolution taken into account. The actual component recommendations can be found in Sect. 5. Note that some of the optical components recommended below might have aberrations not showing up in the simulated results, since they were assumed to be perfect in the simulations. However, it has been shown that even with moderately expensive, commercially available lenses, the image degrading effects of the aberrations in the telescope are negligible.[18]

3.1 Overview

When describing an optical system, it often helps to simplify it – describing a microscope or a telescope as a system of two thin lenses, for example. A description of the HS sensor in the version presented here, however, does not have much to gain from such an oversimplification. Of course, it is possible to build a more complex version in terms of components[3], but in this case portability was desirable.

Remember that the purpose of the HS sensor is to measure the gradient of the wavefront so that a reconstruction of the wavefront itself can be done by a computer. This is done by letting the wavefront be divided into approximately plane sections by an array of microlenses, and observing how much the resulting spots in the focal plane of the lenslets move compared to their reference positions straight behind their respective lenslets. The movements are registered by the CCD in a digital camera, and can be shown to be proportional to the gradient of the wavefront that gives rise to the spot. Remember also that the main problems that could occur were spots overlapping or changing places, and an insufficient amount of light reaching the CCD.

The HS sensor, then, does not have that many components to it. Fig. 3.1 gives a schematic view of the setup, and Fig. 3.2 is a photograph of a very similar setup with the rays shown in Fig. 3.1 drawn in white, and the lens and retina of the eye represented as black lines. Please note that Fig. 3.1 is not drawn to scale. Table 3.1 lists the components shown in Fig. 3.1. The corresponding components are indicated by the same letters in both figures. The telescope has been folded to save space and enable an easy way to change the distance between the lenses to correct for myopia or hyperopia in the eye. The prism-like object P between T1 and T2 in Fig. 3.2 is a 45°-45°-90°-prism with two outer surfaces silvered to serve as mirrors, and the mirrors directly above the prism are simply two very well-aligned plane mirrors. The setup proposed in this work will resemble the setup in Fig. 3.2 in many respects, and will have the folded telescope with adjustable tube length.
**Fig. 3.1. A schematic overview of the Hartmann-Shack sensor**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLD</td>
<td>Superluminscent diode, the light source.</td>
</tr>
<tr>
<td>CL</td>
<td>Collimating lens.</td>
</tr>
<tr>
<td>A</td>
<td>Aperture, diameter 1 mm.</td>
</tr>
<tr>
<td>BS1</td>
<td>Beamsplitter with intensity reflectance 0.1.</td>
</tr>
<tr>
<td>L</td>
<td>Lens of eye.</td>
</tr>
<tr>
<td>R</td>
<td>Retina of eye.</td>
</tr>
<tr>
<td>T1</td>
<td>First lens of telescope.</td>
</tr>
<tr>
<td>T2</td>
<td>Second lens of telescope.</td>
</tr>
<tr>
<td>BS2</td>
<td>Dichroic beamsplitter for the pupil camera.</td>
</tr>
<tr>
<td>ET</td>
<td>Pupil camera connected to screen (not shown).</td>
</tr>
<tr>
<td>LA</td>
<td>Lenslet array.</td>
</tr>
<tr>
<td>CCD</td>
<td>CCD array of camera. Camera housing not shown.</td>
</tr>
</tbody>
</table>

**Table 3.1. A list of the components in Fig. 3.1**
Fig. 3.2. Photograph of a Hartmann-Shack sensor and optics [i4]
3.2 The components

This is in no way a complete list of all things needed to construct a HS sensor, but rather a discussion concerning the more vital components. This section also includes an estimation of the light economy in the system, since this is often a crucial issue in measurements of this kind.

3.2.1 Light source

When choosing the light source for the HS sensor system, three demands have to be met. Primarily, the light has to be collimated (parallel) when emitted, or fairly easy to collimate. Secondly, it has to be intense enough to ensure that a sufficient intensity reaches the CCD, or the HS pattern of spots will be too faint to detect (see Sect. 3.2.6) or too noisy to find the correct centroid of. Thirdly, since the gecko for which the sensor is designed only has retinal receptors in the range from UV to blue light, red light would not provoke the pupil to contract. This could also be solved by keeping the exposures short enough, but this could cause the amount of light reaching the CCD to be insufficient, and red light sources are readily available.

The first type of light source that comes to mind when these specifications are given is a laser. It is easy to collimate, lasers come with much higher output powers than needed–more than 1.23mW on a human retina during our assumed exposure time of 0.1 seconds can severely damage it. The intensity levels of the light in this project follows the rules and guidelines set up by the Swedish Radiation Protection Insitute.[19] Since no data for gecko retinas is available, the same limits as for humans will be assumed to suffice. As we shall se, our light intensity will be a factor of four below the limit. There is, however, one catch with using a laser. The coherency properties of laser light are such that they might cause speckles to appear on the CCD. For a thorough description of the phenomenon, please refer to [20]. To our purposes, it will be enough to know that the coherence properties of the laser light can cause statistical interference on the CCD, and that this can severely degrade the information content in the image. Our light source of choice will instead be a superluminescent diode (SLD), which most easily could be described as a laser diode where the light only passes once through the cavity.
The light is fairly monochromatic (see fig 3.3), and the coherence is low.

![Fig. 3.3. Typical spectral distribution from superluminescent diode [i5]](image)

The light from the SLD is coupled into an optical fiber by the manufacturer, whereupon some 2/3 of the output power is irretrievably lost.[21] One drawback compared to using a laser is that the SLD/fiber combination gives fairly divergent light; the divergence is about 12.6°, given that the fiber core has a diameter of 3.8 µm.[22] This means that in order not to lose too much light, a collimating lens must be placed fairly close to the fiber end. It can be shown (see Sect. 3.2.6) that a commercially available lens will suffice.

### 3.2.2 The Eye

Very little is known *a priori* about the optics of the gecko eye, since very few measurements have been made. The gecko eye has a pupil diameter of around 3 mm when relaxed, and focal lengths between 5 mm and 8 mm are to be expected.[23] The fact that so little is known about the aberrations and focal properties of the eye, together with the fact that it is unclear which cell layer of the retina the main part of the reflected light comes from, makes it very hard to incorporate the entire effect of the eye’s optics in the simulations. Therefore, the retina has been assumed to be a single, perfectly diffuse surface. As already mentioned, the same safety limits of laser power as for a human eye are assumed to hold.

### 3.2.3 The Optics and Pupil Camera

The optical components in the setup are listed in Table 3.1 above. Information about the lenslet array of the sensor can be found in Sect. 3.2.5. In the order that the light from the SLD encounters the components, the first ones are the collimating lens and aperture. The collimating lens is placed with the end of the fiber from the SLD in one focal point. In order to get good collimation, and achromat with an *f*-number of around 5 and a focal length around 10 mm should be used (cf Sect. 3.3). This lens, together with the optics of
the eye, will (in absence of aberrations in the eye) serve to image the fiber end onto the retina. After passing the lens, the beam is narrowed down to a diameter of 1 mm by an aperture. The diffraction effects of this aperture are included in the simulation of the *psf* of the eye (cf Sect. 4). By choosing a focal length of the collimating lens which is greater than that of the eye, it is possible to ensure that the size of this image is slightly smaller than the size of the diffraction spot from parallel incoming light on the retina. The effects of this have not been included in the simulations since so little is known about the optics of the eye. However, this omission should not perturb the results significantly since the centroid of each HS spot remains unshifted.

The beamsplitter that serves to direct the light into the eye unfortunately also reflects a portion of the light from the eye back to the SLD. Since it is vital to keep as much of this light as reasonably possible in the system, a high transmittance is needed. The beamsplitter must also not be opaque to the wavelength used by the pupil camera.

The telescope with its prism and mirrors is constructed from two achromats, whose focal lengths are chosen as to give a magnified image of the pupil on the lenslet array without letting any light pass outside the array. Using the estimation that the gecko has a pupil diameter of 3 mm, an (lateral) magnification of $M_{\alpha}=7$ would be sufficient (cf Sect. 3.2.5). The silvered prism and the two mirrors are all commercially available components.

The pupil camera serves to give a live video feed to a connected screen (not depicted in Fig. 3.1), enabling more exact positioning of the eye with respect to the optical axis of the setup. In order not waste more of the reflected light from the eye than necessary, the eyetracker should operate in the infrared, and the beamsplitter directing light up to the camera be dichroic. A ring of IR-diodes can serve as light source for the eyetracker.

### 3.2.4 Camera

The spatial resolution achievable in the HS sensor is to a fairly large extent dependant on the camera in use, with the size and number of pixels in the CCD detector being the greatest concern. Smaller pixels means that the measurement point are packed more densely, yielding a higher resolution capacity. The size of the CCD array itself, however, is of paramount importance since it limits the size of the recordable HS pattern – if the image of the pupil on the lenslet array is greater than, or even of the same size as, the CCD array, spots will fall out of the measurement simply because the light never reaches the CCD. Since the simulations described in Sect. 4 rely on the pupil shape being circular or elliptic, this can cause severe errors. One way to work around this problem would be to minify the spot pattern onto the CCD. This option has been thoroughly investigated, and it was found that the aberrations in this imaging process introduce intolerable errors. The reasonable thing, then, seems to be a very large CCD array with very small pixels. However, it is hard to make a reliable CCD array with too many pixels without the price becoming very high indeed, and so the feasible thing to
do, both from a physical and economical standpoint, is to increase the pixel size somewhat. The camera in the simulations has an array of $2048 \times 2048$ $12\mu m$-pixels, which means that the CCD measures somewhat less than $24.6mm \times 24.6mm$. This enables us to magnify the image of the pupil several times onto the lenslet array, and thus get more spots in our pattern. The camera also has 12 bits ADC, meaning that it can convert the continuous variations in intensity incident on the CCD into $2^{12} = 4096$ discrete levels.

3.2.5 The Lenslet Array

The lenslet array consists of about 10,000 square lenslets, each with a side length (or pitch) of a few hundred microns. The array is made by molding epoxy of optical quality onto a flat substrate of BK7 glass polished to a surface flatness of $\lambda/4$ and anti-reflex-coated on one side. The lenslet array is mounted at focal length’s distance from the CCD array of the camera, so that if the incoming light is parallel, a spot results straight behind each lenslet. Due to the small size of the lenslets the effects of diffraction far surpasses the effects of any aberrations in the lenses. Since the lenses are square in shape, each lens gives a sinc$^2$-shaped intensity distribution on the focal plane, with a distance from peak to first minimum of (cf. Sect. 2.2) $\lambda f / \text{pitch}$. The choice of parameters on the lenslet array was guided by the simulations described in Sect. 4. The results presented there lead us to consider two arrays in the remainder of Sect. 3; both have focal lengths of 18 mm and pitches of 250 $\mu m$ and 325 $\mu m$, respectively.
3.3 Light Economy

As was previously mentioned, it is vital to have light economy in mind when designing. A bad light economy might mean that some spots become so faint that the signal to noise ratio becomes too much of a problem, or even worse; that the spots become too faint to detect at all. Fig. 3.4 represents the HS sensor system, but this time only with the components that restrict the amount of light that reaches the CCD of the camera. The P:s indicate the power (energy/unit time) of the light at different locations in the setup.

![Diagram of the light economy system](image)

**Fig. 3.4. The intensity restricting components in the system**

The output power of the fiber-coupled SLD is taken to be $P = 5\, \text{mW}$ according to [21,22]. This will be reduced by the aperture of the collimating lens and the 1mm-aperture, and further attenuated by the reflection in the first beamsplitter. Purely geometrical considerations, and assuming the fiber tip to be a point source, give the power just before the eye as

$$P_{in} = R_{BS1} \left( \frac{D_{ap}}{2\alpha f_{CL}} \right)^2 P, \quad (3.1)$$

where $R_{BS1} = 0.1$ is the intensity reflectance of the first beamsplitter, $D_{ap} = 0.001\, \text{m}$ is the diameter of the aperture, and $f_{CL} = 0.01\, \text{m}$ is the focal length of the collimating lens. $\alpha$ is half the angle that the emitted light subtends, or about $12.6^\circ$. A numerical evaluation of Eq. 3.1 yields about $25.8\, \mu\text{W}$, which is far below the safety limit of $1.23\, \text{mW}$ for an 0.1 second exposure stipulated in [19]. If the lens could be chosen with a focal length of 5 mm, the power would increase to $103.2\, \mu\text{W}$, which is still safe by a factor of more than ten.
The beam will now be focused onto the retina, which will reflect the light. Part of the light will be lost in internal reflections in the eye and in the surface between air and cornea. This part will be assumed to be 10% when the light passes into the eye, and the same upon exiting. As stated in Sect. 3.2.2, the exact parameters of the eye are unknown, but we can expect a pupil diameter of 3 mm and a focal length between 5 mm and 8 mm. Furthermore, we shall assume that 95% of the light is lost through absorption, scattering, and other processes in and on the retina. Let us now assume the reflection to take place at a point 7 mm from the (now assumed thin) lens, and assume the retina to be a lambertian surface. If we let $\alpha_{\text{eye}}$ be half the plane angle that the lens subtends as seen from the retina, the power exiting the eye will be

$$P_{\text{out}} = T_{\text{cornea}}^2 \cdot 0.05 \cdot \frac{1}{2} \left( \cos \left( 2\alpha_{\text{eye}} \right) - 1 \right) \cdot P_{\text{in}}, \quad (3.2)$$

which is about 0.46 $\mu$W with our given set of parameters. This light is now further attenuated by passing through the beamsplitter BS1, the telescope, the second beamsplitter and the lenslet array. An expression for the total power right before the lenslet array, denoted $P_L$ in fig 3.4, will then be $P_L = T_{\text{BS1}} \cdot T_{\text{telescope}} \cdot R_{\text{BS2}} \cdot P_{\text{out}}$.

The epoxy in the lenslet array has an amplitude transmittance of about 90% at all wavelengths up to 1500 nm, and 70% of the transmitted light in each lenslet ends up in the central spot of this lenslet.[24]

This, together with all the previous calculations, gives us a quite hideous expression for the power in each central spot as a function of system parameters:

$$P_{\text{spot}} = \frac{0.9 \cdot 0.7 \cdot T_{\text{BS1}} \cdot T_{\text{telescope}} \cdot R_{\text{BS2}} \cdot T_{\text{cornea}}^2 \cdot 0.05 \cdot \frac{1}{2} \left( \cos \left( 2\alpha_{\text{eye}} \right) - 1 \right) \cdot R_{\text{BS1}} \left( \frac{D_{\text{ap}}}{2\alpha f} \right)^2 \cdot P}{\pi \left( \frac{M_{\text{telescope}} D_{\text{pupil}}}{2 \cdot \text{pitch}} \right)^2} \quad (3.3)$$

In order to get somewhat of a worst-case scenario we use the following set of parameters: $T_{\text{telescope}} = 0.96$, $R_{\text{BS2}} = 0.96$, $M_{\text{telescope}} = 7$. The interesting thing is now to compare this to the sensitivity of the CCD array; the object here is to ensure that a sufficient amount of light reaches the detector. Since the total power in a spot depends on the pitch, we get two different values (cf. Sect. 3.2.5) of $P_{\text{spot}}$.  

39
\[
P_{\text{spot}} \left( \text{pitch} = 250 \right) \approx 4.3 \text{pW}
\]
\[
P_{\text{spot}} \left( \text{pitch} = 325 \right) \approx 7 \text{pW}
\]

Now, the camera of choice (cf. Sect. 5) has a responsivity given as \(36 \text{DN}/(\text{nJ} \cdot \text{cm}^{-2})\) for light with a wavelength of 530 nm, which means that if a pixel is illuminated with an intensity of \(1 \text{nJ} \cdot \text{cm}^{-2}\) during an exposure, the read-out value from that pixel will be 36 digital levels (remember that the camera can convert the signal into 4096 levels, cf.. Sect. 3.5).[25] Since we are working at 680 nm instead of 530, we must take the difference in quantum efficiency at different wavelengths into consideration.

The quantum efficiency of the camera as a function of wavelength is given in Fig. 3.5 below, with the relevant wavelengths and efficiencies indicated by the dashed lines.

![Fig 3.5. The quantum efficiency of the CCD in the camera][16]

We see that at 680 nm, we can expect a responsivity of roughly 60\% of that specified above.

Assuming an exposure time of 0.1 seconds, and using the mean intensity in a spot calculated simply as the total power divided by the area out to the first diffraction minimum (cf. Sect. 3.6), this would give

\[
\text{Response} \left( \text{pitch} = 250 \right) \approx 390 \text{ digital levels}
\]
\[
\text{Response} \left( \text{pitch} = 325 \right) \approx 1070 \text{ digital levels}
\]

Possible ways of increasing this would be to increase the exposure time, which would change the response proportionally, or to decrease the focal length of the collimating lens somewhat, since the response scales as the inverse square of this focal length.
However, what has to be remembered now is that we have done a worst-case calculation in terms of the light amounts – hopefully, and much likely, the intensity will be higher than this. Furthermore, this is the mean intensity in one spot, meaning that the center pixels will register higher intensities and the peripheral ones lower intensities. In fact, the center pixel in a spot will register four to five times the mean intensity in the spot. Taking this into account, together with the somewhat pessimistic choices of parameters in this section, the conclusion is that a sufficient amount of light reaches the detector, albeit with no great margin.
4 Simulations in MATLAB

This section describes what came to be the most time-consuming part of this work; to simulate a HS sensor, with emphasis on using the result to choose a lenslet array (cf Sect. 3). For the actual code used in the simulations, the reader is asked to refer to the appendix. All simulations were made in MATLAB versions 6.5.0 and 7.0.1

4.1 The Simulated System

The entire system, as described in Sect. 3, need not be included in the simulations. Figure 4.1 depicts the simulated part of the system, with the optical components in some way included in the simulations indicated by the text.

![Figure 4.1. Schematic of the simulated system](image)

Since in the developed simulation code it is possible for the user to control the radius of the beam sent into the eye, the aperture after the collimating lens (cf Sect. 3) could be included, but since all simulations were made with the same value of this parameter (500 µm) it is omitted below. The effects of it are incorporated when the psf of the eye is calculated, as described in Sect. 4.1.1.

As discussed in Sect. 3, the telescope is assumed to be free of aberrations. This might appear to be an oversimplification, since any aberrations in the telescope would add to the HS spot movements in an unpredictable way. However, the simplifications this provides to the calculations far surpass the impact that these aberrations might have, as it has been shown that these are nearly negligible.[18]

The light economy of the system is incorporated into the simulations when the photon quantum noise and ADC noise are implemented. Since both of these are strongly dependant on the number of photons arriving at each pixel during an exposure, the simulation includes a hefty safety margin. No other types of noise have been implemented. For a more complete discussion of different types of noise and their effects, the reader is urged to consult [14].
Most parameters of the system are controllable by the user from the main program in the file “SimMain.m”. Table 4.1 is a list of the parameters which are supposed to be easily changed. Since the program is written as to be easily adapted to deal with elliptical pupils instead of circular, some parameters may seem superfluous at first glance. The default value of the parameters is also listed, if it exists.

The user also controls the aberrations of the eye by giving the wavefront aberrations expressed as a sum of the 55 first Zernike polynomials. Since these are defined over the unit circle, it is necessary to give as input the radius over which they are defined in the simulations (normally the pupil radius). The aberrations are defined as the deviations from a plane wave, measured in µm, right after exiting the eye. The program then handles all re-scaling of coordinates.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_lenslets</td>
<td>Array consists of N_lenslets × N_lenslets elements</td>
<td>See Table 4.2</td>
</tr>
<tr>
<td>pitch</td>
<td>The pitch, or side length, of the square lenslets in µm</td>
<td>See Table 4.2</td>
</tr>
<tr>
<td>f</td>
<td>Focal length of lenslets in µm</td>
<td>See Table 4.2</td>
</tr>
<tr>
<td>mag</td>
<td>Lateral magnification of the telescope</td>
<td>7.5 (^1)</td>
</tr>
<tr>
<td>rpc</td>
<td>The radius of the Zernike polynomials in µm</td>
<td>1500µm</td>
</tr>
<tr>
<td>Rptheta</td>
<td>Orientation of elliptic pupil in radians measured from x-axis to minor axis</td>
<td>0</td>
</tr>
<tr>
<td>rpmin</td>
<td>Minor axis of the pupil in µm</td>
<td>1500µm</td>
</tr>
<tr>
<td>rpmax</td>
<td>Major axis of the pupil in µm</td>
<td>1500µm</td>
</tr>
<tr>
<td>eyebeam_radius</td>
<td>Radius of the beam entering the eye in µm</td>
<td>500µm</td>
</tr>
</tbody>
</table>

*Table 4.1, controllable parameters and their default values*

Apart from these, parameters such as the wavelength of the SLD, resolution and ADC properties of the camera and the coordinates where the center of the beam enters the eye can be set by the user, but not from the main program.

As we have seen, the setup serves to image the spot on the retina onto the CCD array. This means, according to the results presented in Sect. 2.2, that we, up to a scaling factor due to the telescope, can calculate the intensity distribution on the CCD by convolving the intensity distribution on the retina with the point spread function of the lenslet array including the aberrations in the eye. If we approximate the light source to be a perfect point source and neglect all aberrations in the collimating lens, the intensity distribution on the retina will simply be the point spread function of the eye. Since all aberrations in the eye are set by the user, we implicitly also control the \(psf\) of the eye, as we shall see below.

\(^1\) With the exception of array 190-10 (cf Table 4.2), where mag = 6.
Performing the actual convolution is a comparatively simple matter, as we know (cf Sect. 2.2) that a convolution can be calculating by inverse Fourier transform of the product of the Fourier transforms. The main concern thus becomes simulating the psfs in an effective and correct way.

### 4.1.1 Simulating psfs and HS pattern

**The Eye**

Since the user sets the aberrations of the eye, she also determines the point spread function of the eye. Let us consider a small part of the setup, as shown in Fig. 4.2: the beam entering the eye and the eye, comprised of the optics of the eye and the retina.

![Fig. 4.2. Arrangement for calculating the psf of the eye](image)

We recall from Sect. 2.2 that a psf can be calculated using Fourier transform. Since all optics of the eye a reversible, meaning simply that it does no difference whether the light is going into or out of the eye, it is of no consequence that we know only the shape of the wavefront as it exits the eye. By excising the part of the wavefront subtended by the beam entering the eye, we implicitly have all the information we need – the amplitude psf of the eye will be the Fourier transform of this part of the wavefront, scaled by a quadratic phase factor as shown in Eq. 2.18. The intensity psf is the simply taken to be the modulus square of this, neglecting the proportionality constant.

There is one more thing to bring to mind before continuing: The results in Eqs. 2.21-2.23 were derived for a system with a magnification of unity, so that the coordinate system of the object plane was the same as that of the image plane. In the simulated system, this is no longer the case. The magnification of the telescope introduces a need for rescaling the coordinate system in which the psf of the eye is described before convolution can take place. This is no big problem, but according to the results of Sect. 2.3, changing the size of a coordinate system changes the sampling density in the
Fourier plane of this system. This means that apart from re-scaling the values on the coordinate axes, the \textit{psf} of the eye must also be re-sampled to have the same sampling density as the \textit{psf} of the lenslet array before we can convolve them.

Figure 4.3 shows the \textit{psf} of an eye with aberrations before (left) and after (right) re-sampling. The images have been color inverted so that a dark shade corresponds to a high pixel value in the camera. The limited number of grayscales in printing also limits the number of pixels appearing to be non-zero in the right plot. The center coordinate is not in the same location in the images.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{psf_example.png}
\caption{The \textit{psf} of an eye before and after resampling to fewer pixels. The length scale is the same in both images.}
\end{figure}

From this we understand that it is important to ensure that the sampling density before re-sampling is higher than that of the \textit{psf} of the lenslet array, so that no interpolation is necessary. The program will print a warning if the user tries to set geometrical properties of the system which cause a risk of sampling errors.

\section*{The Lenslet Array}

The point spread function of the whole lenslet array is very unusual, as it not even nearly resembles a point. A system with heavy aberrations might have a \textit{psf} which is smeared out and asymmetrical as in Fig. 4.3, but the \textit{psf} of the lenslet array is not one, but several thousand points – one for each lenslet.

\begin{itemize}
\item In the \textit{psf} of the lenslet array is simulated in the program, the aberrations of the eye included, but the spot on the retina is assumed to be a perfect point source. This means that spot movements are taken into account. Thus a perfect retinal spot is imaged onto the CCD once by each lenslet as depicted in Fig. 4.4. Note that this figure does not depict any spot movements.
\end{itemize}
4.4. Arrangement for calculating the psf of the lenslet array

Now, it might be tempting to try to calculate the psf of the lenslet array by direct Fourier transform over the entire array at once. While this is of course theoretically possible, in reality there are some insurmountable problems. One that becomes obvious fairly early is that the output of such a Fourier transform (or rather FFT) is given in a far too small coordinate system: According to Eq. 2.25

\[
\begin{align*}
    u_{\text{min}} &= -\frac{N\Delta u}{2} = \frac{N\lambda f}{4x_{\text{min}}} \\
    u_{\text{max}} &= u_{\text{min}} + N\Delta u = \frac{(N-2)\lambda f}{4x_{\text{max}}}
\end{align*}
\]

(2.25 revisited)

where \( u \) is the coordinate in the (one dimensional) focal plane, \( x \) lives in the lens plane and \( N \) is the number of (evenly spaced) samples. Remember also that FFT works much faster when \( N \) is an even power of two. If we now, just for the sake of demonstration, assume that we have a wavefront which is 20 mm in diameter, and we wish for our simulation to take a reasonable amount of time. Assume that we could sample as dense as 8192 samples in each dimension (this is a high number – the actual simulations have 132 samples each time FFT is run). This would yield a coordinate system in the focal plane ranging from -2.5 mm to 2.5 mm which, while having a high number of samples per unit length, is far too small since the whole spot pattern should have about the same size as the wavefront. We would like a coordinate system that covers our CCD array, so that we are simulating the camera in a feasible way. In fact, we do not need that high a sampling density in the focal plane – it is pointless to sample denser than the pixels in the CCD we are simulating.

These circumstances – that our sampling density in the focal plane is in fact already set, and that we have restrictions on how small the coordinate system in the focal plane can be allowed to become, implies that we should let the coordinate system in the lens plane be set from what we want as an output, and not the other way around.
In the simulation, the problem is solved accordingly: The step length in the focal plane is set to one sample every 12 µm, corresponding to the center-to-center distance of pixels on the CCD. Each lenslet is then given its own separate coordinate system in the focal plane, with a range corresponding to five times the pitch of the lenslet. The sampling density is set by dividing the range by the step length and setting the number of sample points to the next higher power of two. The sampling density in the lenslet plane, as well as the range of the coordinate system there are given by Eqs. 2.24 and 2.25. This coordinate system subtends an area which is larger than one lenslet, which simply means that the function that is to be transformed (the wavefront) is set to zero outside the lenslet. This also negates the need to pad the matrix with zeros; recall that in order to be able to compute the FFT, the transformand is assumed to be periodic – repeating itself endlessly outside the coordinate system. A “padding” of zeros outside the actual transformand is therefore often necessary to avoid interference (not in the optical sense) from the surrounding “copies” of the transformand. Figure 4.3 depicts a single HS spot as simulated by the program. Note that this image has been color inverted and that some gray level adjustments have been made for reasons of clarity. Note also the sinc²-shape of the spot due to the quadratic aperture.

![Image](image.png)

*Fig. 4.5. One of the spots in the pattern on the CCD array.*

When the spots are simulated as described above, there is one more detail to which attention must be paid. There is in the general case no guarantee whatsoever that the center of a lenslet will have a CCD pixel located straight behind it. However, there is no compensation for this inherent in any of the methods used. Therefore, it is necessary to introduce subpixel shifts to most spots, to ensure that they end up straight behind the lens (if the input is a plane wave) and not just in the central pixel of the coordinate system in the focal plane. These shifts are fairly easy to calculate, since the positions of all lenslets are well known. The shifts are executed in the FFT itself: Since a multiplication with a linearly varying phase shift of an entire function corresponds to translation in Fourier space (Eq. 2.5), the problem can be solved by calculating the appropriate phase shifts and then let the FFT handle the actual shifting.
When this has been done, each of these small coordinate systems is pasted into the correct position in a large one, corresponding to the actual CCD, and each pixel value is added to the existing one (which is zero if nothing has been pasted there). Note that all these calculations, just as for the eye, are done in terms of complex amplitudes of the fields.

The resultant intensity distribution will also contain the movements of the spot due to the aberrations in the eye, if the point on the retina were in fact a perfect point. Since it is not, and since the \textit{psf} in the eye might be asymmetrical with a shifted center of gravity (centroid), a small amount of spot movement might result from the convolution. After convolving the \textit{psf} of the eye with the \textit{psf} of the lenslet array, however, we are very close to what an actual image taken by the camera would look like.

The one thing that remains to do is the add noise to the picture: Photon quantum noise, which is an effect of the fact that the number of photons actually arriving at a certain pixel during an exposure is in fact Poisson distributed, and ADC noise – an effect of the fact that the CCD cannot register a continuous spectrum of energies, but rounds the input to one of 4096 discrete gray levels. Matlab has a built-in function to add Poisson noise to a picture which the program utilizes.[26] ADC noise is added simply by rescaling the intensity to range (in integer steps) from zero to the maximum output in ADC levels desirable.

All simulation of the HS pattern and the \textit{psfs} is performed by the method \textit{“HSsim” (cf Appendix)} and its subroutine \textit{“pixelize”}, which handles the resampling of the \textit{psf} of the eye. When this is done, the HS pattern is simulated, and the simulation moves on to the next step, of which the main part is the attempted reconstruction of the wavefront given to the program as described below.
4.1.2 Creating and Re-creating Wavefronts

We have now discussed how the program simulates a HS image from a wavefront. It is thus time to turn our attention to whence this wavefront comes, and what is done once the HS pattern is simulated. A main purpose of the program is to enable comparing different lenslets arrays in order to assess which one would be best in our HS sensor. Software for calculating the wavefront from a certain spot pattern already exists[3], and only minor changes had to be made to these programs to make them compatible with the code written for this project.

The idea of the simulations, then, is to let several different lenslet arrays “handle” the same wavefronts, attempt re-creation of the wavefronts and then compare the reconstructed wavefront to the original (user-controlled) one. Comparison of the RMS errors of the different lenslet arrays on different wavefronts should enable us to choose an array, or at least weed out many of the arrays.

As briefly discussed above, the user sets the aberrations of the eye’s optics by giving the coefficients of the 55 first Zernike polynomials, thus creating a function equal to the deviations from a perfect plane wave. The reconstruction software attempts to reconstruct the wavefront in terms of Zernike coefficients, so another way of measuring “correctness” of the reconstruction would be to compare the Zernike coefficients given to the program with the resulting ones.

The reconstruction is based on already existing algorithms [3] up to a very high degree, and many existing algorithms were used as is. The only entirely new reconstruction algorithm is one for “untilting” the reconstructed wavefront. Since the reconstructions is based on how much the spots have moved relative to one another, the reconstructed wavefront almost always differs from the given one by a constant tilt introduced by the reconstruction process. This makes the direct comparison of the given and reconstructed wavefronts difficult, and so an algorithm for removing this tilt was introduced (this algorithm is incorporated in the method “WCompare”, which also calculates the RMS error in the reconstruction).

Untilting is done by least-square-fitting of a plane to the difference between the given and the reconstructed wavefront, and then subtracting this plane from the reconstruction. That this is indeed a correct thing to do can in part be inferred from the fact that the reconstructions show no systematic errors after being untilted.
4.2 The Simulations

This section describes what actual simulations were made to be used for deciding on a lenslet array for the sensor. The only thing changed between the simulations of different arrays, except of course the parameters concerning the lenslet array itself, was a parameter in the reconstruction algorithms. This parameter describes the size of the area in which the program expects to find a singular spot, and since the size of the spots as well as the distance between the spots differ between different arrays for a given wavefront, this parameter had to be changed when the array parameters change.

Table 4.2 lists the lenslet arrays which were simulated. The arrays are characterized by their focal length and pitch, and listed as (pitch in µm)-(focal length in mm). In the table is also \( N_{sim} \), the number of lenslet used in the simulation of each array, based on the fact that the used part of the array should have a diameter of about 23 mm so that all spots end up on the CCD.

<table>
<thead>
<tr>
<th>Array (pitch in µm)- (focal length in mm):</th>
<th>( N_{sim} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>188-8</td>
<td>122</td>
</tr>
<tr>
<td>190-10</td>
<td>98</td>
</tr>
<tr>
<td>203-7.8</td>
<td>113</td>
</tr>
<tr>
<td>250-18</td>
<td>92</td>
</tr>
<tr>
<td>312.5-34</td>
<td>74</td>
</tr>
<tr>
<td>325-18</td>
<td>71</td>
</tr>
<tr>
<td>400-53</td>
<td>58</td>
</tr>
<tr>
<td>400-71</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 4.2. A list of the simulated lenslet arrays

Note that one array, the one denoted 190-10, differs from the rest in that it is the only one where the actual size of the array limits the number of usable lenslets. All other are limited by the 23-mm criterion mentioned above.

Each array was given the same wavefronts, and the resulting HS pattern was simulated as described above. The wavefront were chosen by randomizing the Zernike Coefficients between -2 and 2 for the ten first, -0.5 and 0.5 for the eleven following and -0.1 and 0.1 for the remaining. The wavefronts were inspected to insure that they were neither too similar nor too flat. The wavefronts, or rather the wavefront aberrations (deviations from a plane wave), are depicted in Figs. 4.6a-e. The aberration is given in µm.
Fig. 4.6a. Given wavefront 1
Fig. 4.6b. Given wavefront 2
Fig. 4.6c. Given wavefront 3
Fig. 4.6d. Given wavefront 4
Fig. 4.6e. Given wavefront 5
Note that the scaling is not the same for the different wavefronts, so that a given shade does not correspond to the same value in the different pictures. Note also the very steep right edge of wavefront number four (Fig. 4.6d), as this will be important later. The unit on the axes of the figures is arbitrary; the figures are plotted with 2048×2048 data points. The wavefronts in the simulations have a diameter of 3 mm when exiting the eye.

When the HS pattern had been simulated, the image thus produced was given to the wavefront reconstruction algorithms. The reconstructed wavefront was compared to the given one as described in Sect. 4.1.2 above, and RMS errors were calculated and saved. Along with this, all wavefront data of both given and simulated wavefronts, as well as all simulated camera images were saved.

The reconstruction software was written to also present a number of auxiliary data, such as which lenslets gave spots that reached the CCD, which lenslets could not be connected to spots or vice-versa and more. Unfortunately, the simulated HS images do not lend themselves to printing, as much higher resolution and more space is needed for them to be viewed. All images are available by request from the author.
4.3 Results of the Simulations

When all eight arrays had been simulated and RMS errors calculated, the results were plotted for easy comparison. Figure 4.7 gives the RMS errors in arbitrary units for the different arrays and the different wavefronts in Figs. 4.6a-e. We primarily note that there are two arrays which have fairly flat curves, 203-7.8 and 188-8. We also note that wavefront number four (fig 4.6d) causes a very high RMS error in several arrays.

Upon closer inspection of the results of the simulations of wavefront four, it was discovered that the high gradient in that wavefront’s right edge had caused light to fall outside the CCD, that is, there were spots missing from the pattern due to too large spot movements and not due to the arrays themselves. Since the size of the CCD was fixed, this could only be mended by decreasing the pupil size or magnification of the telescope. Since decreasing the pupil size would have meant that the Zernike coefficients would have described larger aberrations relative to before, the magnification of the telescope was decreased. The changes were made in steps of 0.5 until a lowest possible value of the RMS error was found. It was discovered that the

![Fig. 4.7. Plot of the Root-Mean-Square errors when reconstructing wavefronts for the simulated arrays.](image-url)
same phenomenon had occurred also for wavefront number 5, albeit not nearly as severe. The most interesting arrays were therefore run with magnification modification also on wavefront 5.

Figure 4.8 depicts the RMS errors after changing the magnification to 5 for arrays 400-71, 400-53 and 312.5-34, and to 6 for array 325-18 on wavefront 4, and magnification 5.5 for array 250-18 and 6 for 325-18 on wavefront 5. How the array 190-10, which was limited by its number of lenslets and not the CCD size, still produces a much higher error on wavefront number 4.

Fig. 4.9 is a plot of these two last arrays only, as these produced the by far lowest RMS errors.
At first glance, the array 325-18 seems like the better one, as it produces a lower RMS error in three cases of five. However, the array 250-18 yields more and denser measurement points, something which might prove important if further testing are carried out (see Sect. 6). The light economy of the two different arrays (Sect. 3.3) must also be taken into consideration, as the array 325-18 gives brighter spots due to its larger pitch.

All in all, the only thing we can conclude from this is that further testing of these two arrays might be necessary. For recommendations of how to proceed simulated testing (which might of course be carried out on all lenslet arrays, disregarding of their results here), please refer to Sect. 6.
5 Conclusions and Recommendations

A high spatial resolution HS sensor has been designed, with crucial components chosen according to the recommendations below. It has been shown that we can expect to be able to use an area of more than 50 lenslets in diameter, depending on choice of array. In the simulations with the lowest magnification (6 for array 325-18 and 5.5 for array 250-18), the arrays utilized areas 55 and 66 lenslets in diameter, respectively. This corresponds to about 2400 and 3400 measurement points in all, but using the telescope to correct for myopia or hyperopia in the eye this could probably be increased as the magnification could be set to around 6.5 or 7.

Regarding the precise choice of lenslet array, it is recommended to run some more simulations (see Sect. 6), and if these are not conclusive, buy both and perform actual tests. Economically, this should not be an obstacle, since the price of the lenslet arrays is low compared to other components (about 6% of what the camera costs, and 15% of what the SLD costs.)

Table 5.1 lists precise recommendations for the crucial components. All other components have been investigated in the sense that they are available as per specifications, but no exact recommendations are given. More information on these components can be gained in Sect. 3. The two lenslet arrays are those previously denoted as “325-18” and “250-18”.

<table>
<thead>
<tr>
<th>Component</th>
<th>Product</th>
<th>Manufacturer/agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLD</td>
<td>Model SLD-261-HP1-DIL-SM-PD with Pilot 4 driver</td>
<td>Superlum Diodes [27]</td>
</tr>
<tr>
<td>Lenslet Array</td>
<td>AOA 0325-18-S or 0250-18-S, epoxy on window “B”</td>
<td>Adaptive Optics Ass. [28]</td>
</tr>
<tr>
<td>Camera</td>
<td>Dalsa 6M18</td>
<td>Dalsa/Parameter AB [29]</td>
</tr>
</tbody>
</table>

Table 5.1, recommended components

Choosing other crucial components, or other components for auxiliary optics than indicated in Sect. 3, might void the results presented in this report by introducing effects not taken into account or by changing important parameters.

The results and simulations presented herein are of course not free from errors. Numerical errors in the algorithms used certainly exist. As an example, the FFT always differs slightly from the actual Fourier transform. Other deviations from the results obtained when the apparatus has been built might include unsimulated noise in the camera, misalignment errors and unforeseen aberrations. However, the great uncertainty regarding the gecko eye remains a likely source of errors larger than these.
6 Outlook

In this concluding section it is discussed what can be done in the further development of the project.

Build the setup

The most interesting thing yet to be done in this project is of course the actual building of the HS sensor setup and to start taking some real measurements. However, if time and resources exist, there some improvements that could be made to the simulations. In order of importance, these are:

Simulate arrays more systematically

It is suggested that this is done by configuring the program to calculate RMS errors in some suitable unit and then setting a maximum allowable value of the RMS error. The arrays should then be given a radially symmetric aberration function which varies only slowly (Fig. 6.1 left). Next, all arrays below the set RMS value should be given a wavefront which varies slightly faster (Fig. 6.1 middle and right).

Investigate telescope aberrations more closely

Even though it has been shown that in normal cases, aberrations in the telescope can be neglected, no-one has (to our best knowledge) performed this kind of measurement on a wavefront which is not smooth, nor has it been done with this many lenslets. Further investigation in this area might however be very time-consuming.
7.1 References


MATLAB documentation. Execute commands “doc fft” and “doc fft2” in a MATLAB prompt.


According to SLD product sheet acquired from SuperlumDiodes Ltd. Can be found at www.superlumdiodes.com, visited January 2005

According to professor Ronald H. H. Kröger with the Lund Vision Group[1]


Execute command “doc imnoise” in a MATLAB prompt. Note that the Image Processing Toolbox is needed for the method to exist.


See homepage at www.parameter.se, last visited February 2005
7.2 Image sources

[i1] Image adapted from Aberration Theory, Geunyoung Yoon, Department of Ophthalmology, Center for Visual Science, University of Rochester.


[i3] Adapted from [8]

[i4] Photograph courtesy of Austin Roorda, University of Houston College of Optometry

[i5] Adapted from product sheet acquired from SuperlumDiodes Ltd. Can be found at www.superlumdiodes.com, visited January 2005

[i6] Adapted from [25]
8 Appendix

Herein is contained the main parts of the new code written for this project. Older files are excluded for reasons of space. They are treated in, for example, [3].

8.1 Matlab Code

SimMain.m

% Main .m-file for HS sensor/camera simulation
% Calls function in files HSsim.m (which calls on pixelize.m),
% HSmain2.m, WCompare.m, ScalePupil.m and AlignC.m
% Note that HSmain2 calls subroutines in some or all of Findspot.m,
% Scaling.m, PPupil.m,
% ReadData.m, Rotation.m, SortSpot.m, HSpupil.m, fitellipse.m,
% Wavefront.m,
% DT.m, ChgWL.m, ZernikeSphere.m

clear all;
close all;
format long;
format compact;
warning('off','MATLAB:dispatcher:InexactMatch');

%Editable constants
%All lengths in microns
N_lenslets=92; %The array will consist of N_lenslets x N_lenslets lens elements
pitch=250; %Side length of (square) lenslets
f=18000; %focal length of lenslets
mag=7.5; %Telescope magnification

%Constants in eye coordinates:
rpc=1500; %Radius of Zernike polynomials
RPtheta=0; %Pupil tilt. Measured in radians from the x-axis to RPmin.
rpmin=1500; %Pupil minor axis
rpmax=1500; %Pupil major axis
eyebeam_radius=500; %Radius of beam entering eye

if rpmin*mag>N_lenslets*pitch
    warning('SimMain:SmallArray':'
The simulated array is smaller then
the magnified wavefront. Decrease magnification and buy some flowers
for your spouse.
');
end

% Zernike coefficients for incoming wavefront at the eye.
Ctot=importdata('C.txt');

%Create reference image.
reference_image=imread(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),'-
',num2str(f/1000),'\ref.tif']);
%plot results of array simulation? (Recommended only for debugging or if you have memory and time to waste)
plots=false;

%Save HS images?
save_im=false;

%Loop through the wavefronts given by Zernike coefficients in C.txt
RMS=[];
minerrs=[];
maxerrs=[];
for c=1:5
    C=Ctot(:,c);
    %Create a HS image of the wavefront
    [system_parameters, pupil_data, noisy_image]=HSsim(C,N_lenslets,pitch,f,eyebeam_radius,mag,rpmax,rpmin,rp c,RPtheta,plots,save_im);
    %Attempt to re-create the wavefront
    [C_simulated, RPmaxsim, RPminsim, RPthetasim]=HSmain2(system_parameters, noisy_image, reference_image);
    save(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),'-',num2str(f/1000),'
        ',num2str(c),'
        ',num2str(f/1000),'
        ',num2str(c),'
        noimage_image']);
    clear noisy_image
    %Scale Zernike coefficients to correct pupil size
    C_simulated=ScalePupil(C_simulated,pupil_data(1)/RPmaxsim);
    %Remove piston and tilt terms.
    %Note that this might introduce a constant error in the gradient of the
    %wavefront. This is corrected for in the function WCompare
    C_simulated=AlignC(C_simulated);
    %Compare the simulated wavefront to the given, correct for error in
    %gradient and calculate RMS and maximum error.
    [Wavefront,Wavefront_sim,WdiffP,RMS(c)]=WCompare(C,C_simulated,pupil_data,system_parameters);
    minerrs(c)=min(WdiffP(:));
    maxerrs(c)=max(WdiffP(:));
    save(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),'-',
        num2str(f/1000),'
        ',num2str(c),'
        Wavefront',num2str(c),'
        Wavefront_sim']);
    save(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),',
        ',num2str(f/1000),'
        ',num2str(c),'
        Wavefront_sim',num2str(c),'
        Wavefront_sim']);
    save(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),',
        ',num2str(f/1000),'
        ',num2str(c),'
        Wavefront_diff',num2str(c),'
        WdiffP']);
end

error_range=maxerrs-minerrs;
%Plot results
figure(20);
cf;
high=max(max(Wavefront(:)),max(Wavefront_sim(:)));
low=min(min(Wavefront(:)),min(Wavefront_sim(:))):


subplot(1,3,1),imagesc(Wavefront,[low high]);axis equal;
title('The wavefront given to the program')
subplot(1,3,2),imagesc(Wavefront_sim,[low high]);axis equal;
title('The measured wavefront')
subplot(1,3,3),imagesc(WdiffP,[low high]);axis equal;
title('The difference')

colorbar

figure(21);
cf;
imagesc(WdiffP);
axis equal;
title('The difference')
colorbar

%%Print error values
disp('The differences between the true WF and the measured, aligned and
untilted WF range from ')
disp([num2str(minerrs), ' to ', num2str(maxerrs)])
disp('Which gives error ranges of')
disp(num2str(error_range))
disp('with RMS errors of')
disp(num2str(RMS))
save(['Z:\Matlab\Lenslets\HS images\',num2str(pitch),'-'
'\results'],[minerrs,maxerrs,RMS])

HSSim.m

% Lenslet array simulation version 7:
% HSSim(C,N_lenslets,pitch,f,eyebeam_radius,mag,rpmax,rpmin,rp,t,save_im)
% Simulates the HS pattern given by simulated lenslet array and wavefront
% (described by the
% Zernike coefficients in C), including poisson and ADC noise
% C should be a 55x1 matrix.
% Save_im is a boolean telling whether you want to save your HS images as
% .tif files for later use.
% Returns two vectors of parameters for use by SimMain, as well as the HS
% image

% Notes:
% Further attempts to introduce a correct subpixel shift. (now tested)
% Checks if lens is within pupil
% Allows elliptic and tilted elliptic pupil
% Wrong sign of C fixed
% Convolves with retinal spot
% Checks what part of spot is within CCD area
% Allows light from lenslet slightly outside CCD area (not yet
implemented)
% //Otto Manneberg
function [system_parameters, pupil_data, noisy_image]=HSsim(C,N_lenslets,pitch,f,eyebeam_radius,mag,rpmax,rpmin,rpfc,RPtheta,plots,save_im)

disp('Initializing HS simulation...')
%Camera constants and wavelength used:
%All lengths in microns
N_pixels=2048;
lambda=.680;
pixelsize=12;

%Don't mess with stuff below this line unless you have a decent idea of
%what you are doing unless you just did "save as"

%Constants in lenslet coordinates, assuming a perfect telescope:
RPcenter=[0 0]; %Pupil center
RPc=rpc*mag; % in lenslet coordinates
RPMin=rpmin*abs(mag); % in lenslet coordinates
RPMax=rpmax*abs(mag); % in lenslet coordinates
Eyebeam_Radius=eyebeam_radius*abs(mag); % in lenslet coordinates
Eyebeam_Center=[0 0]; % in lenslet coordinates

%Vectors for calling HSmain2
pupil_data=[rpmax rpmin RPtheta rpc];
system_parameters=[N_pixels pixelsize pitch f mag];

if abs(RPc)<abs(RPmax)
warning('Lenslets:SmallRPC','
The specified value of RPC is less
than RPmax. This might cause erroneous results as well as nightly
attacks by rouge Frenchmen');
end

if eyebeam_radius>RPmin
warning('Lenslets:LargeBeam','
The beam entering the eye is larger
than the minor axis of the pupil. Bad things often happen to good
people.');
end

%Setting up the coordinate systems. We assume that each lenslet
%only to the field in an area the size of 5x5 lenslets centered straight
%behind it.
disp('Creating coordinate systems, lens position matrices in CCD plane')
disp('and quadratic phase factor...')
tic
urange=5*pitch; %Sampling range in CCD plane for light from one lens
ustep=pixelsize; %Sampling density in CCD plane
N_points=2^nextpow2(urange/ustep); %Number of sampling points in sampling range

%Coordinate system in lens plane
xstep=lambda*f/(N_points*ustep); %Sampling density in lens plane
x=[-N_points/2:N_points/2-1]*xstep;
[X,Y]=meshgrid(x);
Y=flipud(Y);
% Coordinate system in CCD plane in microns
u=[-N_points/2:N_points/2-1]*ustep;
[U,V]=meshgrid(u);
V=flipud(V);

% The part of (X,Y) system subtended by lens
center=abs(x)<(pitch/2);
xcenter=x(find(center));
[Xc,Yc]=meshgrid(xcenter);
Yc=flipud(Yc);

% Pupil shape, i.e. a square. I,J are indices of nonzero elements (portion
subtended by lens)
P=(abs(X)<(pitch/2)).*(abs(Y)<(pitch/2));
[I,J]=find(P);

% Vectors containing coordinates of all lens centers.
x0=(-(N_lenslets-1)/2:(N_lenslets-1)/2)*pitch;
[X0,Y0]=meshgrid(x0);
X0=X0(:);
Y0=Y0(:);

% Lens center in pixel coordinates
Upix_exact=X0/pixelsize;
Vpix_exact=Y0/pixelsize;

% Nearest integer pixel. Diffraction pattern will be shifted here by brute
force
Upix_int=round(Upix_exact);
Vpix_int=round(Vpix_exact);
% Remainder. This shift will be done by introducing a phase factor in the
wavefront
Upix_rem=Upix_exact-Upix_int;
Vpix_rem=Vpix_exact-Vpix_int;

% Phase factors
Xshifts=exp(i*2*pi*Upix_rem*pixelsize/(lambda*f));
Yshifts=exp(i*2*pi*Vpix_rem*pixelsize/(lambda*f));

% Quadratic phase factor
k=2*pi/lambda;
quad_factor=1/(i*lambda*f)*exp(i*mod(k/(2*f)*(U.^2+V.^2), 2*pi));

disp('Done')
disp('Calculating WFAs and FFT:ing pupil functions. Percentage
complete:')

% Matrices for the diffraction pattern from the lenslets
E_lens=zeros(N_pixels); % Diffraction pattern from current lens, amplitude
Etot=E_lens; % Total diffraction pattern, amplitude
Ec=N_pixels/2+1; % The matrix element with coordinates (u,v)=(0,0)
percentage=0; % counter
for m=1:N_lenslets^2
  % Lens centrum
  x0=X0(m);
  y0=Y0(m);
  ...
% Check if any part of lens is inside image of pupil
if (((x0-pitch/2*sign(x0))*cos(RPtheta)+(y0-pitch/2*sign(y0))*sin(RPtheta))./RPmin).^2+((y0-pitch/2*sign(y0))*cos(RPtheta)-(x0-pitch/2*sign(x0))*sin(RPtheta))./RPmax).^2 <= 1;

% Calculates wavefront aberrations from the Zernike coefficients in vector C
% Sets P as the pupil function
WFA=W(C,(Xc+x0)/RPC,(Yc+y0)/RPC);
WF=exp(-1i*mod(k*WFA,2*pi));
P(min(I):max(I) , min(J):max(J))=WF;

% Elliptic pupil function
P_elliptic=((X+x0)*cos(RPtheta)+(Y+y0)*sin(RPtheta))./RPmin).^2+((y0+Y)*cos(RPtheta)-(X+x0)*sin(RPtheta))./RPmax).^2 <= 1;

% Fourier transform (including subpixel positioning) and multiply by the quadratic phase factor
new_spot=quad_factor.*fftshift(fft2(fftshift((Xshifts(m).^X).*((Yshifts(m).^Y).*P.*P_elliptic))));

% Paste new spot onto CCD:
% Tests if the part of the new spot is outside CCD, and in that case calculates what part of the spot is to be included
U_lower_CCD=Ec-Vpix_int(m)-N_points/2; % Lower edge of new spot, MATLAB matrix coordinates in CCD coordinate system
U_lower_spot=1; % Lower edge of new spot, MATLAB matrix coordinates in spot's own coordinate system
if U_lower_CCD < 1 % Is part of the new spot outside?
    U_lower_spot=abs(U_lower_CCD)+2; % This is where the spot will be cut
    U_lower_CCD = 1; % This is where it will be pasted
end

% Three more edges:
U_upper_CCD=Ec-Vpix_int(m)+N_points/2-1;
U_upper_spot=size(new_spot,1);
if U_upper_CCD > N_pixels
    U_upper_spot = size(new_spot,1)-U_upper_CCD+N_pixels;
    U_upper_CCD=N_pixels;
end

V_lower_CCD=Ec+Upix_int(m)-N_points/2;
V_lower_spot=1;
if V_lower_CCD < 1
    V_lower Spot=abs(V_lower_CCD)+2;
    V_lower_CCD = 1;
end

V_upper_CCD=Ec+Upix_int(m)+N_points/2-1;
V_upper_spot=size(new_spot,1);
if V_upper_CCD > N_pixels
    V_upper_spot = size(new_spot,2)-V_upper_CCD+N_pixels;
    V_upper_CCD = N_pixels;
end

% Paste the spot as calculated above into the CCD plane
E_lens(U_lower_CCD:U_upper_CCD , V_lower_CCD:V_upper_CCD) =
new_spot(U_lower_spot:U_upper_spot , V_lower_spot:V_upper_spot);
Etot=Etot+E_lens; % Total diffraction pattern so far
E_lens=zeros(N_pixels);
end

% Display percentage complete
if (100*m/N_lenslets^2) >= percentage
    disp(num2str(percentage));
    percentage=percentage+5;
end
m=m+1;
end
% clear U V X Y quad_factor E_lens new_spot P P_elliptic WF WFA

toc
disp('Done')
tic

% Calculate PSFs of array and eye (normalised to have peak values of 1)
disp('Calculating point spread function of lenslet array and eye:')
PSF=Etot.*conj(Etot);
PSF=PSF/max(PSF(:));
clear Etot

% Coordinate system large enough to encompass part of wavefront
% corresponding to pupil
% function of eye plus padding, in lens plane:
totpoints=2048;
xtot=[-totpoints/2:totpoints/2-1]*xstep;
[Xtot,Ytot]=meshgrid(xtot);
Ytot=flipud(Ytot);
current_step=lambda*f/(xstep*totpoints);
step_quotient=ustep/current_step;

% Check if the (Xtot,Ytot) coordinate system will FFT to a system large
% enough to contain the ten first zeros of a perfect Bessel spot
if (lambda*f/(2*xstep)) < (32.1897/pi*lambda*f/(2*Eyebeam_Radius))
    warning('Lenslets:BadSampling','\n\nThe (u,v) coordinate system for
the PSF of the eye might be too small.\nTake a break. You have earned
it\n');
end

% Pupil shape of the eye
P=(abs(Xtot-Eyebeam_Center(1))<Eyebeam_Radius).* (abs(Ytot-
Eyebeam_Center(2))<Eyebeam_Radius);
[I,J]=find(P);
Xtotc=Xtot(min(I):max(I),min(J):max(J));
Ytotc=Ytot(min(I):max(I),min(J):max(J));
%Calculating the total WFA
disp('Calculating total WFA...')
WFA_P=W(C,(Xtotc+Eyebeam_Center(1))/RPC,(Ytotc+Eyebeam_Center(2))/RPC);

%Pupil function of the eye is calculated for the area where it will be
%non-zero and the pasted into a larger matrix
disp('Calculating pupil function and subpixel shifts...')
P_eye_small=exp(-i*mod(k*WFA_P,2*pi)).*(((Xtotc-Eyebeam_Center(1)).^2+(Ytotc-Eyebeam_Center(2)).^2)<=Eyebeam_Radius^2);
P_eye=zeros(size(Xtot));
P_eye(round(size(P_eye,1)/2-size(P_eye_small,1)/2) +1 : round(size(P_eye,1)/2+size(P_eye_small,1)/2) , round(size(P_eye,2)/2-size(P_eye_small,2)/2)+1 : round(size(P_eye,2)/2+size(P_eye_small,2)/2))=P_eye_small;

%Phase shift pupil function to make sure that the center of the transform
%ends up in the pixel corresponding to (u,v)=(0,0) - a plane wave should
%give a symmetric diffraction pattern!
xshift=exp(i*2*pi*(step_quotient-1)/2*current_step/(lambda*f)*Xtot);
yshift=exp(-i*2*pi*(step_quotient-1)/2*current_step/(lambda*f)*Ytot);

%Fourier transform
disp('FFTing pupil function...')
eye_spot=fftshift(fft2(fftshift(xshift.*yshift.*P_eye)));
clear Xtot Ytot xshift yshift P P_eye

%PSF of the eye
PSF_eye_dense=eye_spot.*conj(eye_spot);
clear eye_spot

%Resampling of the PSF of the eye to a sampling density equal to that of
%PSF of lens matrix
disp('Resampling and convolving...')
resampled=pixelize(PSF_eye_dense,step_quotient);
PSF_eye=zeros(size(PSF));
PSF_eye(round(size(PSF_eye,1)/2-size(resampled,1)/2) +1 : round(size(PSF_eye,1)/2+size(resampled,1)/2) , round(size(PSF_eye,2)/2-size(resampled,2)/2)+1 : round(size(PSF_eye,2)/2+size(resampled,2)/2))=resampled;
PSF_eye=PSF_eye/max(PSF_eye(:));

%Convolution of array PSF with eye PSF by multiplication of transforms
Itot=real(ifftshift(ifft2(ifftshift(fftshift(fft2(fftshift(PSF_eye)))).*f fftshift(fft2(fftshift(PSF)))))));
disp('Done')

%Plotting the results of HSsim is plots is set to "true"
if plots==true
    disp('Plotting...')
    figure(10);
    clf;
    colormap(jet(256));
    imagesc(PSF)
hold on;
plot(Upix_exact+Ec,Vpix_exact+Ec,'yo');
t=[0:0.01:2*pi];
P_edgeV=(RPcenter(1)+RPmax*cos(t)*cos(RPtheta)- RPmin*sin(t)*sin(RPtheta))/pixelsize;
P_edgeU=(RPcenter(2)+RPmax*cos(t)*sin(RPtheta)+RPmin*sin(t)*cos(RPtheta))/pixelsize;
plot(P_edgeU+Ec,P_edgeV+Ec,'w-')
axis equal;
title(['Psf with ',num2str(N_pixels),' pixels, f=',num2str(f/1000),'mm, pitch=',num2str(pitch),' um'])
xlabel('u');ylabel('v');
clear PSF

figure(11);
clf;
colormap(jet(256));
imagesc(PSF_eye(462:562,462:562));
axis equal;
title(['Image of retinal spot on CCD'])
clear PSF_eye

figure(12);
clf;
colormap(jet(256));
imagesc(PSF_eye_dense(912:1112,912:1112));
axis equal;
title(['Image of retinal spot on CCD, magnified ',num2str(step_quotient),' times'])
clear PSF_eye_dense

figure(13);
clf;
colormap(jet(256));
imagesc(real(Itot)/max(Itot(:)));
hold on;
plot(Upix_exact+Ec,Vpix_exact+Ec,'yo');
t=[0:0.1:3*pi];
P_edgeV=(RPcenter(1)+RPmax*cos(t)*cos(RPtheta)- RPmin*sin(t)*sin(RPtheta))/pixelsize;
P_edgeU=(RPcenter(2)+RPmax*cos(t)*sin(RPtheta)+RPmin*sin(t)*cos(RPtheta))/pixelsize;
plot(P_edgeU+Ec,P_edgeV+Ec,'w-')
axis equal;
title(['Intensity with ',num2str(N_pixels),' pixels, f=',num2str(f/1000),'mm, pitch=',num2str(pitch),' um'])
xlabel('u');ylabel('v');
else
    clear PSF PSF_eye PSF_eye_dense
end
%Adding Poisson distributed noise (photon quantum noise) and ADC noise
noisy_image=imnoise(uint16(round(2e6/4096*Itot)),'poisson);
clear Itot

if plots==true
    figure(14);
    clf;
    imagesc(noisy_image);
    axis equal;
end
toc

%Save image if save_im is set to "true" when HSsim is called
if save_im==true
    [imfile,impath] = uiputfile('*.tif','Select image file for noisy image');
    IM=imfile(1)~=0;
    if IM
        imwrite(noisy_image,[impath imfile],'tiff','Compression','none','Resolution',260);
    end
end

Pixelize.m

% function new_image=pixelize(old_image,factor)
% pixelizes the image old_image to new_image, whose size is size(old_image)/factor.
% factor should of course be a positive integer for decent results.

function new_image=pixelize(old_image,factor)

    if (factor<1) | ((round(factor)-factor)~=0 )
        warning('Pixelize:Badfactor','\n\nAnd a mighty voice cried out from above, speaking thusly:\nThe pixelation factor should be set to an integer greater than one\\n')
    end

    N_pixels=size(old_image,1)/factor;
    points_per_px=factor;
    pixelized=zeros(N_pixels);

    for n=0:N_pixels-1
        for m=0:N_pixels-1
            pix_val=sum(sum(old_image(n*points_per_px+1:(n+1)*points_per_px,
                m*points_per_px+1:(m+1)*points_per_px)));
            pixelized(n+1,m+1)=pix_val;
        end
    end

    new_image=pixelized;
WCompare.m

function [WAC,WACsim_untilted,WdiffP,RMS] = WCompare(C,Csim,pupil_data,sys_parameters)

function Wcompare(C,Csim,pupil_data)
%Calculate the wavefront aberrations (in microns) described by the
Zernike
%coefficients C and Csim as well as the difference between them.
%Corrects for error in tilt in the simulated wavefront by fitting a plane
to the difference
%and adding this plane to the wavefront.
%Also calculates the RMS difference between wavefronts.
%Pupil data is a vector looking like [Rmax Rmin Rtheta RPC]
%The coefficients in C should be defined at least over pupil radius RPC
[microns].
%RPC should also be at least as large as Rmax
%Returns wavefront described by C and wavefront described by Csim
%Based on old method PlotW

%Otto 04-12-07:
%Used method PlotW to create new version to be used in simulation of
lenslet arrays.
%All fancy graphics alternatives gone, returns wavefronts, difference and
%errors. Sets parts outside of pupil to 0 instead of NaN.
%
%Peter Unsbo 02-06-04
%
%Linda 15 may 03:
%Removes M=PlotW()
%Sets the parts of WA outside the pupil to NaN in the plot
%
%rpmax=pupil_data(1);
%rpmin=pupil_data(2);
%rptheta=pupil_data(3);
%rpc=pupil_data(4);
pitch=sys_parameters(3);
mag=sys_parameters(5);

if abs(rpc)<abs(rpmax)
    warning('WCompare:SmallRPC','
          The specified value of RPC is less
          than Rmax. This might cause erroneous results as well as nightly
          attacks by rouge Frenchmen');
end

%Restrict pupil
rpc=abs(rpc);
if rpc<rpmax,
    b=rpmax/rpc; %fraction of pupil used
    a=rpmin/rpc;
else
    b=1;
    a=rpmin/rpmax;
end
%Exclude an annulus of width pitch/2 at the periphery, since the numerical errors here are too large
%A coordinate system:
n_points=2047;
x=[-n_points/2:n_points/2]*2*b/(n_points);
[X,Y]=meshgrid(x);

%Calculate the wavefronts from their Zernike coefficients
disp('Calculating wavefront aberrations...')
WAC=W(C,X,Y);
WACsim=W(Csim,X,Y);

%Cut away everything outside the pupil, and exclude and annulus of width pitch/2 at the periphery, since the numerical errors here are too large
disp('Restricting wavefronts and calculating differences...')
R=sqrt(X.^2+Y.^2);
theta=sign(Y).*acos(X./R);
WAC(R>(a*b./sqrt((a*sin(theta-rptheta)).^2+(b*cos(theta-rptheta)).^2))-pitch/(2*mag*rpc))=0;
WACsim(R>(a*b./sqrt((a*sin(theta-rptheta)).^2+(b*cos(theta-rptheta)).^2))-pitch/(2*mag*rpc))=0;

%The difference between the wavefronts
Wdiff=WAC-WACsim;

%To eliminate tilt difference, which is of no physical significance, fit a plane to Wdiff. (Xs,Ys) cover largest possible square inside the pupil. "s" is for "square"
disp('Correcting tilt and calculating errors...')
xs=x(abs(x)<((a-pitch/(2*mag*rpc))/sqrt(2)));
[Xs,Ys]=meshgrid(xs);
Wdiffs=Wdiff(round((n_points+1)/2-length(xs)/2)+1 : round((n_points+1)/2+length(xs)/2) , round((n_points+1)/2-length(xs)/2)+1 : round((n_points+1)/2+length(xs)/2));
Plane_coeffs=[Xs(:) Ys(:) ones(size(Xs(:)))] \ Wdiffs(:);
Plane=Plane_coeffs(1)*X+Plane_coeffs(2)*Y+Plane_coeffs(3);

%Calculate the untilted difference and the tilt corrected simulated wavefront
WdiffP=Wdiff-Plane;
WACsim untilted=WAC-Wdiff+Plane;
WdiffP(R>(a*b./sqrt((a*sin(theta-rptheta)).^2+(b*cos(theta-rptheta)).^2))-pitch/(2*mag*rpc))=0;
WACsim(R>(a*b./sqrt((a*sin(theta-rptheta)).^2+(b*cos(theta-rptheta)).^2))-pitch/(2*mag*rpc))=0;

%The RMS error and the maximum error of the difference
RMS=sqrt(sum(WdiffP(:).^2)/length(WdiffP));

%Set points outside pupil to zero
WdiffP(R>(a*b./sqrt((a*sin(theta-rptheta)).^2+(b*cos(theta-rptheta)).^2))-pitch/(2*mag*rpc))=0;
WACsim_untilted(R>(a*b./sqrt((a*sin(theta-\theta)).^2+(b*cos(theta-\theta)).^2))-pitch/(2*mag*rpc))=0;
disp('Done')