ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MASTER THESIS

Evaluation of Optical Aberrations using Phase Diversity

Author:
Jordan VOIRIN

Supervisors:
Dr. Laurent JOLISSAINT
Dr. Jean-Paul KNEIB

A thesis submitted in fulfillment of the requirements for the degree of Master in Applied Physics

in the

Astrophysics laboratory
Basic Sciences

February 22, 2018

Declaration of Authorship

I, Jordan VOIRIN, declare that this thesis titled, "Evaluation of Optical Aberrations using Phase Diversity" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"For the sake of persons of ... different types, scientific truth should be presented in different forms, and should be regarded as equally scientific, whether it appears in the robust form and the vivid coloring of a physical illustration, or in the tenuity and paleness of a symbolic expression. "

James Clerk Maxwell

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

Introduction

???

Chapter 1

Theoretical background

In this chapter, we will present the theory upon which this work is based. First, the light propagation formalism will be reminded through the scalar diffraction theory based on the Goodman (1968). Then we will describe in general an imaging system and its properties. And finally we will discuss the wavefront aberration theory.

1.1 Scalar Diffraction Theory

1.1.1 Scalar Field and Helmholtz equation

A monochromatic wave, at position P and time t, can be represented by a scalar field $u(P,t)$ written as :

$$u(P,t) = A(P)exp\left[-j2\pi\nu t + j\phi(P)\right], \tag{1.1}$$

where $A(P)$ and $\phi(P)$ are the amplitude and phase, respectively, of the wave at position P and ν is the wave frequency.

The spatial part of eqt. (1.1), also called phasor in the literature,

$$U(P) = A(P)e^{j\phi(P)}, \tag{1.2}$$

must verify the Helmotz equation :

$$(\nabla^2 + k^2)U = 0, \tag{1.3}$$

where k is the wave number given by

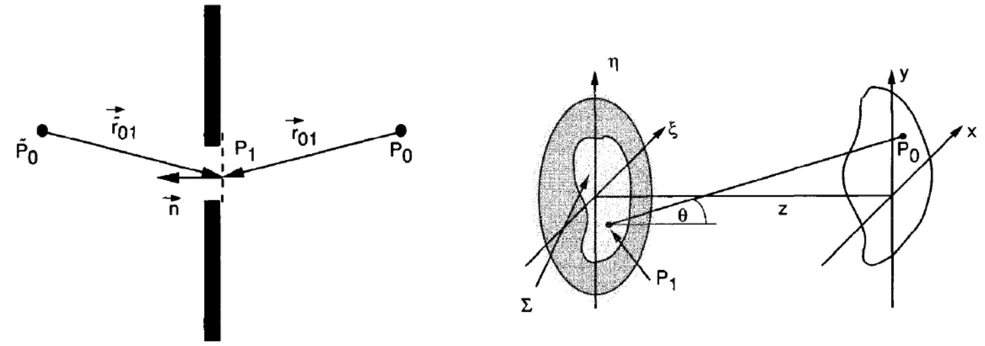
$$k = 2\pi n\frac{\nu}{c} = \frac{2\pi}{\lambda}, \tag{1.4}$$

and λ is the wavelength in the dielectric medium.

1.1.2 Rayleigh-Sommerfeld integral

Rayleigh and Sommerfeld developed a formalism using the Helmholtz equation and Green's Theorem to compute the induced diffraction by a plane screen. Let's suppose that we have a monochromatic source at $\widetilde{P_0}$ on the left of a plane screen with aperture Σ, the Rayleigh-Sommerfeld formula allows to compute the complex amplitude at P_0 on the right of the plane screen (see Figure 1.1a).

$$U(P_0) = \frac{1}{j\lambda}\iint_{\Sigma} U'(P_1)\frac{exp(jkr_{01})}{r_{01}}cos(\mathbf{n},\mathbf{r_{01}})\mathbf{ds} \tag{1.5}$$



(A) Rayleigh-Sommerfeld formulation of diffraction by a plane screen, (Goodman, 1968, Chapter 3.5).

(B) Diffraction geometry, (Goodman, 1968, Chapter 4.1).

FIGURE 1.1: Diffraction Schemas

$U'(P_1)$ is the complex amplitude on the screen, $cos(\mathbf{n}, \mathbf{r_{01}})$ is the cosine of the angle between the aperture plane normal toward the source and the vector $\mathbf{r_{01}} = \mathbf{P_0P_1}$ given by

$$r_{01} = \sqrt{z^2 + (x-\xi)^2 + (y-\eta)^2}. \tag{1.6}$$

We can rewrite eqt. (1.5) using $cos(\mathbf{n}, \mathbf{r_{01}}) = cos(\theta) = \frac{z}{r_{01}}$ and the coordinate systems (ξ, η) and (x, y), see Figure 1.1b,

$$U(x,y) = \frac{z}{j\lambda} \iint_{-\infty}^{\infty} U(\xi,\eta) \frac{exp(jkr_{01})}{r_{01}^2} d\xi d\eta. \tag{1.7}$$

We can integrate from $-\infty$ to ∞, using $U(\xi,\eta) = P(\xi,\eta)U'(\xi,\eta)$ where $P(\xi,\eta)$ is the pupil function. The latter equals to one in the pupil and zero outside.

1.1.3 Fresnel approximation

To reduce eqt. (1.7), also known as the Huygens-Fresnel principle, one can approximate the distance r_{01} using the taylor expansion of the square root :

$$r_{01} = z\sqrt{1 + \frac{x-\xi}{z} + \frac{y-\eta}{z}} \approx z\left[1 + \frac{1}{2}\left(\frac{x-\xi}{z}\right)^2 + \frac{1}{2}\left(\frac{y-\eta}{z}\right)^2\right] \tag{1.8}$$

To obtain the Fresnel approximation, one has to replace r_{01} by eqt. (1.8) in eqt. (1.7). At the denominator, only the first term z is kept, since the introduced error is small, but in the exponential everything is kept. Then the final expression is given by,

$$U(x,y) = \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U(\xi,\eta) exp\left\{j\frac{k}{2z}\left[(x-\xi)^2 + (y-\eta)^2\right]\right\} d\xi d\eta. \tag{1.9}$$

In this form, the Fresnel approximation can be seen as a convolution between $U(\xi,\eta)$ and $h(x,y) = \frac{e^{jkz}}{j\lambda z} exp\left[\frac{jk}{2z}\left(x^2+y^2\right)\right]$.

Another form is found by developing $\left[(x-\xi)^2 + (y-\eta)^2\right]$,

$$U(x,y) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \iint_{-\infty}^{\infty} \left\{ U(\xi,\eta) e^{j\frac{k}{2z}(\xi^2+\eta^2)} \right\} e^{-j\frac{2\pi}{\lambda z}(x\xi+y\eta)} d\xi d\eta, \tag{1.10}$$

it is the Fourier transform of the complex field in the pupil multiplied by a quadratic phase exponential.

1.1.4 Fraunhofer approximation

In addition to the Fresnel approximation, we can introduce another approximation using the condition,

$$z >> \frac{k(\xi^2+\eta^2)_{max}}{2}. \tag{1.11}$$

If eqt. (1.11) is satisfied the Fresnel approximation simplifies, since the quadratic phase factor in (ξ,η) is approximately one on the entire pupil, as

$$U(x,y) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \iint_{-\infty}^{\infty} U(\xi,\eta) e^{-j\frac{2\pi}{\lambda z}(x\xi+y\eta)} d\xi d\eta. \tag{1.12}$$

For instance, at a wavelength of 637.5 nm and a pupil diameter of 3.6 mm the Fraunhofer approximation constrain z to be greater than 63 meters to be valid.

1.1.5 Converging lens introduction

The Fraunhofer conditions are severe as shown above, but one can reduce the distance z by observing at the focal plane of a converging lens. Indeed, using the paraxial approximation, i.e. small angles with respect with the optical axis, the lens transmission function is given by,

$$\begin{aligned} t_l(\xi,\eta) &= exp\left[jkn\Delta_0\right] exp\left[-jk(n-1)\frac{\xi^2+\eta^2}{2}\left(\frac{1}{R_1}-\frac{1}{R_2}\right)\right] \\ &= exp\left[-j\frac{k}{2f}(\xi^2+\eta^2)\right], \end{aligned} \tag{1.13}$$

where n is the refractive index of the lens material, R_1 and R_2 are the radii of curvature of the front and back surface of the lens, respectively and f is the focal length of the lens defined as,

$$\frac{1}{f} \equiv (n-1)\left(\frac{1}{R_1}-\frac{1}{R_2}.\right) \tag{1.14}$$

We can define $U_l(\xi,\eta) = U(\xi,\eta)t_l(\xi,\eta)$, which represents the complex amplitude passing through a lens. Finally, replacing $U(\xi,\eta)$ by $U_l(\xi,\eta)$ in Fresnel approximation and setting the observing distance to the focal length of the converging lens, we recover the Fraunhofer approximation,

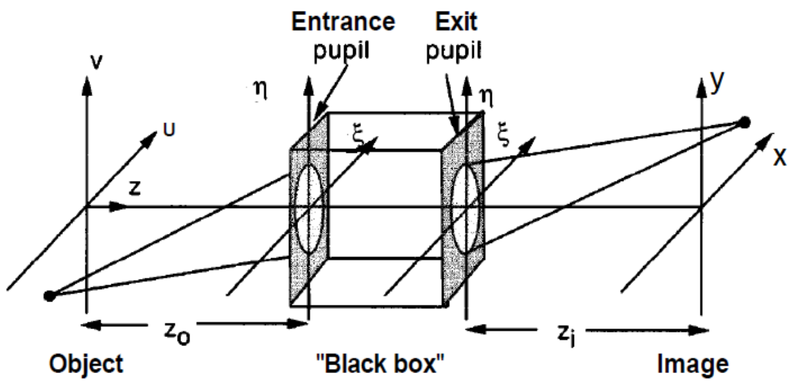


FIGURE 1.2: Schema of a imaging instrument, (Goodman, 1968, Chapter 6.1)

$$
\begin{aligned}
U(x,y) &= \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \mathcal{F}\left\{ U(\xi,\eta) exp\left[j\frac{k}{2}(\xi^2+\eta^2)(\frac{1}{z}-\frac{1}{f}) \right] \right\} \\
&\stackrel{z=f}{=} \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \mathcal{F}\left\{ U(\xi,\eta) \right\}.
\end{aligned} \tag{1.15}
$$

1.2 Imaging system

An imaging system, such as a telescope, is used to acquire images of an object as perfectly as possible. An optical system, forming an instrument, is composed by lenses, mirrors, etc... and a detector (can be the human eye). A complex optical system can be reduced to a pupil, $P(\xi,\eta)$, and a focal length, f. The diffraction of the wave can be determined by the Fraunhofer approximation as long as the paraxial approximation is valid, see subsection 1.1.5. And the observed image of an incoherent object at the focal plane of the system is proportional to the square modulus of the complex amplitude $U(x,y)$,

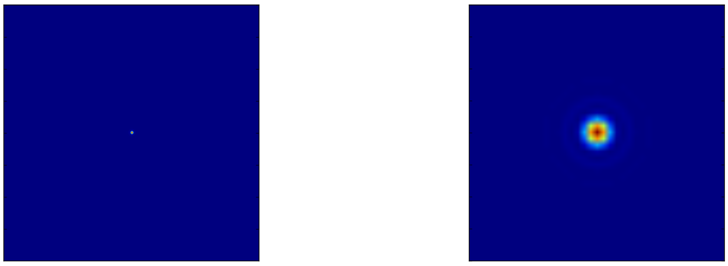
1.2.1 Impulse Response (IR)

The impulse response or point spread function (PSF), $h(x,y;u,v)$, of an optical system is the field amplitude induced at coordinates (x,y) by a unit-amplitude point source at object coordinates (u,v). Using the linearity of the wave propagation, we can write the imaged amplitude as the superposition integral,

$$
U(x,y) = \iint_{-\infty}^{\infty} h(x,y;u,v)U(u,v)dudv \tag{1.16}
$$

1.2.2 Optical Transfer Function (OTF)

The optical transfer function, OTF, is defined as the Fourier transform of the impulse response, see Figure 1.4. Using the Fourier transform properties it can also be given by the autocorrelation of the pupil function,



(A) PSF (B) PSF zoom x10

FIGURE 1.3: PSF of a perfect imaging system composed by a 3.6 mm pupil and a focal length of 80 mm at a wavelength of 637.5 nm. The size, N, of the PSF is 400 and the pixel size is 5.3 μm.

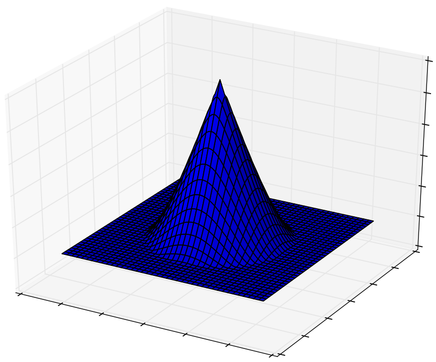


FIGURE 1.4: OTF of a perfect imaging system composed by a 3.6 mm pupil and a focal length of 80 mm at a wavelength of 637.5 nm.

$$\widetilde{h}_{optical}(\xi,\eta) = \mathcal{F}\left\{h_{optical}(x,y)\right\} = (P \otimes P)(\xi,\eta), \tag{1.17}$$

where ξ and η are the conjugate variables of x and y with respect to the Fourier transform.

1.2.3 From Object to Image

A detector only senses the energy distribution produced by an electromagnetic wave. Therefore, an image is given by the square modulus of the complex amplitude at the focal plane,

$$i(x,y) = |U(x,y)|^2 = |\iint_{-\infty}^{\infty} h(x,y;u,v)U(u,v)dudv|^2. \tag{1.18}$$

This integral simplifies differently depending on the type of object we are observing. For a **coherent object**, Goodman (1968, Chapter 6.2) showed that the imaging is linear in complex amplitude, thus eqt. (1.18) becomes,

$$i(x,y) = |\iint_{-\infty}^{\infty} h(x-\widetilde{u},y-\widetilde{v})U(\widetilde{u},\widetilde{v})d\widetilde{u}d\widetilde{v}|^2, \tag{1.19}$$

where $(\widetilde{u} = Mu, \widetilde{v} = Mv)$ are the normalized object coordinates and M is the magnification of the imaging system. The image is given by the squared modulus of the convolution of the impulse response and the object complex amplitude.

For an **incoherent object**, Goodman (1968, Chapter 6.2) showed that the imaging is linear in intensity,

$$i(x,y) = \iint_{-\infty}^{\infty} |h(x-\widetilde{u},y-\widetilde{v})|^2 o(\widetilde{u},\widetilde{v})d\widetilde{u}d\widetilde{v} = (h_{optical} \otimes o)(x,y), \tag{1.20}$$

where $o(x,y) = |U(x,y)|^2$. We can recognize the convolution of an object with an intensity impulse response, $h_{optical}(x,y) = |h(x,y)|^2$.

In this study, we will study stars radiation which are object with an incoherent emission. The stars are point source objects given there distances. The image that an optical system gives of a point source is called the point spread function, PSF, or impulse response, IR, of the system, see Figure 1.3. A point source is characterized by an infinite distance to the instrument and therefore the wave is planar, which means that the phasor is reduced to $U(\xi,\eta) = P(\xi,\eta)$. The PSF or IR is given by,

$$h_{optical}(x,y) = |\left[\mathcal{F}\left\{P(\xi,\eta)\right\}\right](x,y)|^2 \tag{1.21}$$

The domain where the impulse response is invariant under translation is called the **isoplanatic domain**.

In presence of aberrations, which will be discussed in the section 1.3, the wavefront is deformed with respect to the perfect planar or spheric form. The pupil function at the exit of the imaging system is modified as following,

$$\mathcal{P}(\xi,\eta) = P(\xi,\eta)e^{j\phi_{Ab}(\xi,\eta)}, \tag{1.22}$$

where $\phi_{Ab}(\xi,\eta)$ is the dephasing caused by the aberration present between the object and the image planes. Replacing the new pupil function in eqt. (1.21), we obtain the PSF of an imaging system having aberrations on the optical path.

1.3 Aberrations

The aberrations present on the optical path between the object and the image planes decrease the quality of the resulting image. Indeed, they induce fluctuations of the amplitude and phase of the wave in the pupil plane. Since the amplitude fluctuations are negligible with respect to the phase fluctuations, the latter are the ones studied. In figure 1.5, one can see the effect of the aberrations on an hypothetical spherical wavefront at the exit pupil of an imaging system.

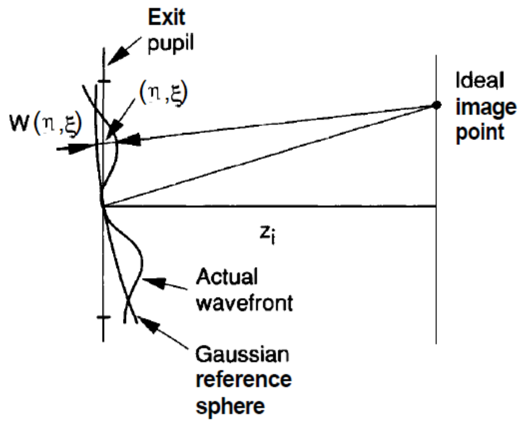


FIGURE 1.5: Gaussian reference sphere vs. aberrated Wavefront, (Goodman, 1968, Chapter 6.4). The dephasing is equal to the wave number multiplied by the aberration function, $\phi_{Ab}(\xi, \eta) = kW(\xi, \eta)$.

1.3.1 Sources of aberrations

1.3.2 Zernike polynomials

Chapter 2

Phase Diversity Experiment

Here, we will describe the experiment put in place in the optical laboratory at the HEIG-VD to reconstruct wavefronts with unknown static aberrations introduced using phase screens. At first, we study the behavior of the phase diversity algorithm put in place by Mugnier, Blanc, and Idier (2006) at ONERA with respect to number of averaging images, in other words noise level, and number of Zernike coefficients retrieved. Then we test the algorithm using a known aberration introduced by a parallel plane plate in the beam comparing the result to Zemax simulation. And finally, we introduce the phase screen to have random aberrations in the pupil and try to compare the phase diversity results with the Shack Hartman wavefront sensor results.

2.1 ONERA algorithm

2.2 Experimental Setup

The design of the experiment was already done by Bouxin (2017). The system is built according to her plans and specifications.

The experiment is mounted on a pressurized legs optical table. The assembly contains six main components : a light source, an entrance pupil, an imaging system, a converging lens to focus the beam on the camera, a camera and a wavefront sensor.

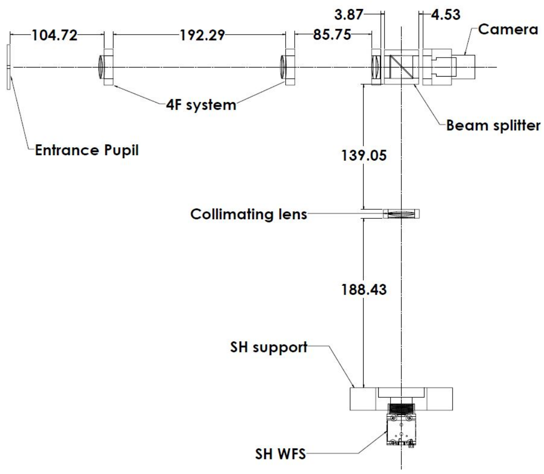


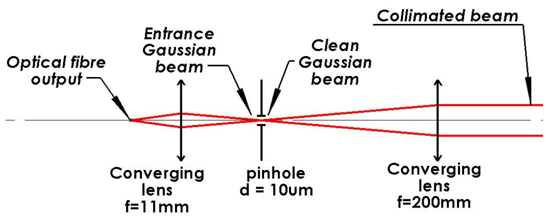
FIGURE 2.1: Experimental setup schema with the relevant distances, (Bouxin, 2017).

2.2.1 Light source

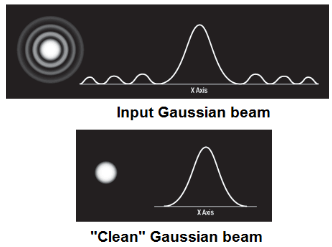
The final application of the phase diversity will be to characterize the optical aberrations induced by the imperfect optical path to a scientific detector of a telescope. For this reason, the light source has to simulate a distant star aberration-free wavefront. A distant star wavefront is considered planar since the object distance, z, is far greater than the telescope size, r, see Fig. 2.2. The source of our experiment must then be characterized by a planar wavefront.

TABLE 2.1: Optical Components

#	Components	Model	Reference
1	Pigtailed laser diode	Thorlabs, LPS-635-FC	A.1
2	Converging lens, f = 11 mm	Thorlabs, A220TM-A	A.2
3	Pinhole, 10 μm	Thorlabs, P10S	A.3
4	Converging lens, f = 200 mm	Thorlabs, AL100200	A.4
5	3.2 mm Hole milled in metal sheet	...	...
6	Converging lens, f = 100 mm	Thorlabs, AC254-100-A	A.5
7	Converging lens, f = 80 mm		
8	Camera CMOS	Ximea, MQ013MG-E2	A.6
9	Converging lens, f = 100 mm		
10	Shack-Hartman WFS	Thorlabs, WFS150-5C	A.7



(A) Source ray tracing.



(B) Beam view before and after the pinhole, (Thorlabs, 2017).

FIGURE 2.3: Source schema and pinhole effect on the beam.

In order to obtain such a planar wavefront at the entrance pupil, the light source consist of a "pigtailed laser diode", a f=11mm converging lens, a pinhole and a f=200 mm converging lens, see Table 2.1. The pigtailed laser diode emits a Gaussian beam centred at 637.5 nm slightly diverging. The converging lens concentrates the beam at the center of the 10μm pinhole to filter the noise. The second converging lens collimates the beam, obtaining a collimated beam with a planar wavefront, see Fig. 2.3a and 2.3b.

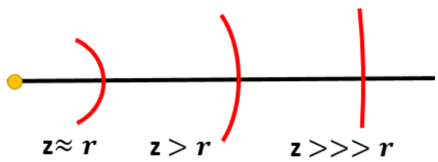


FIGURE 2.2: Wavefront curvature for different point source's distances, z. r represents the characteristic size of the arc of interest.

2.2.2 Entrance pupil

The entrance pupil of our optical system is a circular aperture of 3.2 mm diameter placed after the collimating lens of the light source. It is milled in a metal plate and centred in his support, to avoid positioning with a XY table. The diameter is chosen in available material to fit in the different detector's surfaces.

2.2.3 Pupil imaging system

The phase diversity technique requires PSFs images as input, which means that the beam as to be focused onto the detector surface. To analyse the aberration in the pupil plane, one needs to focus an image of the beam passing through the entrance

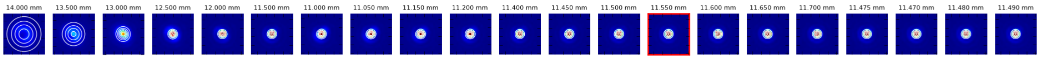


FIGURE 2.4: PSFs example of an alignment procedure

pupil. The simplest assembly to achieve this goal is the 4F system, which consist of two converging lenses of focal 100 mm. The two lenses are separated by 200 mm, see Fig. 2.1. This places the image of the entrance pupil 100 mm after the second converging lens.

2.2.4 Detectors

The image of the entrance pupil, obtained with the 4F system, is focused onto a CMOS Ximea camera by a f = 80 mm converging lens to acquire the PSFs for the phase diversity wavefront retrieval. The camera has a surface composed by 1280x1024 pixels of 5.3 μm, see Appendix A.6. It is mounted on sliding support in order to be able to acquire in/out-of-focus images. A beam splitter is placed in the converging beam to separate it in two. The second beam is collimated and a Shack-Hartman WFS is placed on the entrance pupil image plane, to check the results of the phase diversity wavefront retrieval. The Shack-Hartman WFS has a 39 X 31 lenslets grid and a CCD with a resolution of 1280x1024 pixels of 4.65 μm, see Appendix A.7.

2.3 Data Acquisition

2.3.1 Ximea Camera

The ONERA algorithm takes at least one focused and one defocused PSFs, as described in section 2.1. The PSFs are acquired using a python script which uses an open-source library to control the ximea camera, `pyXimea`[1], available on GitHub. The acquisition is done following these steps :

1. The first step in order to acquire PSFs is to determine the position of the camera's focus point using the python script `AlignementScriptXimeaCamera.py`, see Appendix B.1. This script let's you acquire consecutively PSFs at different camera's positions and computes their FWHM. It finally returns the minimum FWHM and the camera's position, see Figure 2.4.

2. Once you have the focus point position

2.3.2 Shack-Hartman WFS

2.4 Results

This section presents the results of the phase diversity experiment, with the introduction of different sources of aberration.

[1] `https://github.com/pupil-labs/pyximea`

2.4.1 Parallel plane plate

The first source of aberration studied in this work is a tilted parallel plane plate which is used as a calibrated source of astigmatism.

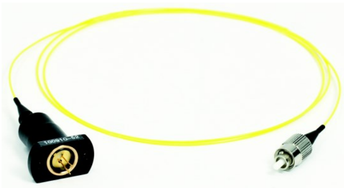
Appendix A

Optical Component Datasheets

A.1 Pigtailed laser diode

THORLABS

Pigtailed Laser Diode, SMF



Description

Thorlabs' Single Mode Pigtailed Laser Diodes are standard TO-packaged diodes that have been pigtailed to a 1 m long single mode fiber with an FC/PC connector. Each unit is tested before shipment. Please refer to the unit-specific test datasheet for optimal operating parameters.

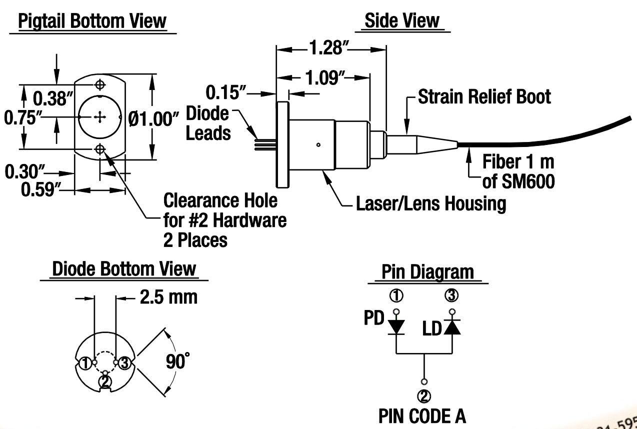
Specifications

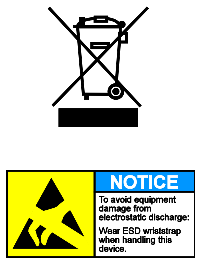
LPS-635-FC Specifications		
LD Reverse Voltage (Max)	2 V	
PD Reverse Voltage (Max)	30 V	
Optical Output Power	2.5 mW (Typ.)	3.5 mW (Max)
Operating Temperature	0 to 50 °C	
Storage Temperature	-10 to 65 °C	
Pin Code	9A	
Laser Diode	HL6320G	
Fiber	SM600	
Connector	FC/PC	

LPS-635-FC Specifications			
	Min	Typ.	Max
Wavelength	625 nm	635 nm	640 nm
Threshold Current*	20 mA	50 mA	75 mA
Slope Efficiency*	0.13 mW/mA	0.18 mW/mA	-
Operating Current @ P_O = 2.5 mW*	-	70 mA	95 mA
Operating Voltage @ P_O = 2.5 mW*	-	2.2 V	2.7 V
Monitor Current @ P_O = 2.5 mW*	0.05 mA	0.17 mA	0.3 mA

*Temperature = 25 °C

Drawing





US, Canada, & South America: +1-973-300-3000 | France: +33 (0) 970 444 844 | Europe: +49 (0) 8131-5956-0 | UK & Ireland: +44 (0)1353-654440
Brazil: +55-16-3413 7062 | Scandinavia: +46-31-733-30-00 | Japan & Asia: +81-3-5979-8889 | China: +86 (0)21-60561122

www.thorlabs.com

June 6, 2013
5175-S01, Rev B

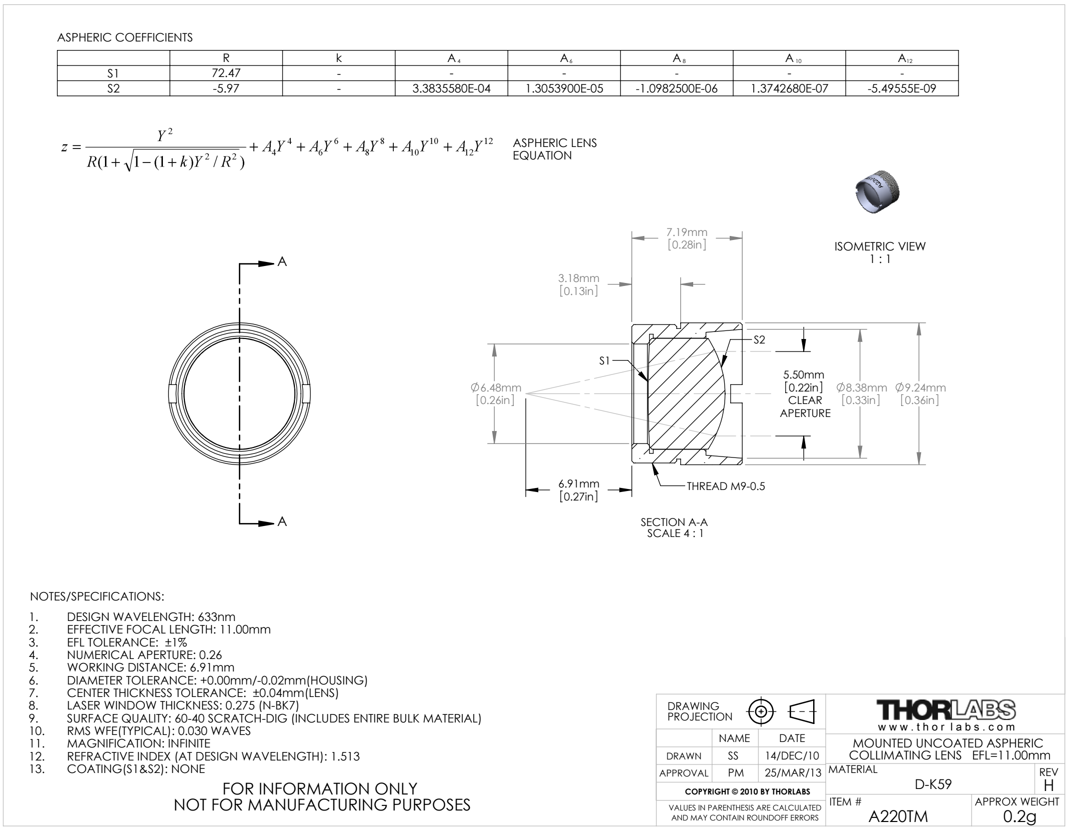
Source : www.thorlabs.com

A.1.1 Power supply modification

The former student, ??? its name ???, mounted a diode driver card to power it. Unfortunately, for the phase diversity experiment, chapter , the power was too high and the Ximea camera was always saturated. So I modified the driver circuit and added two resistances to lower the current so that the detector do not reach the saturation.

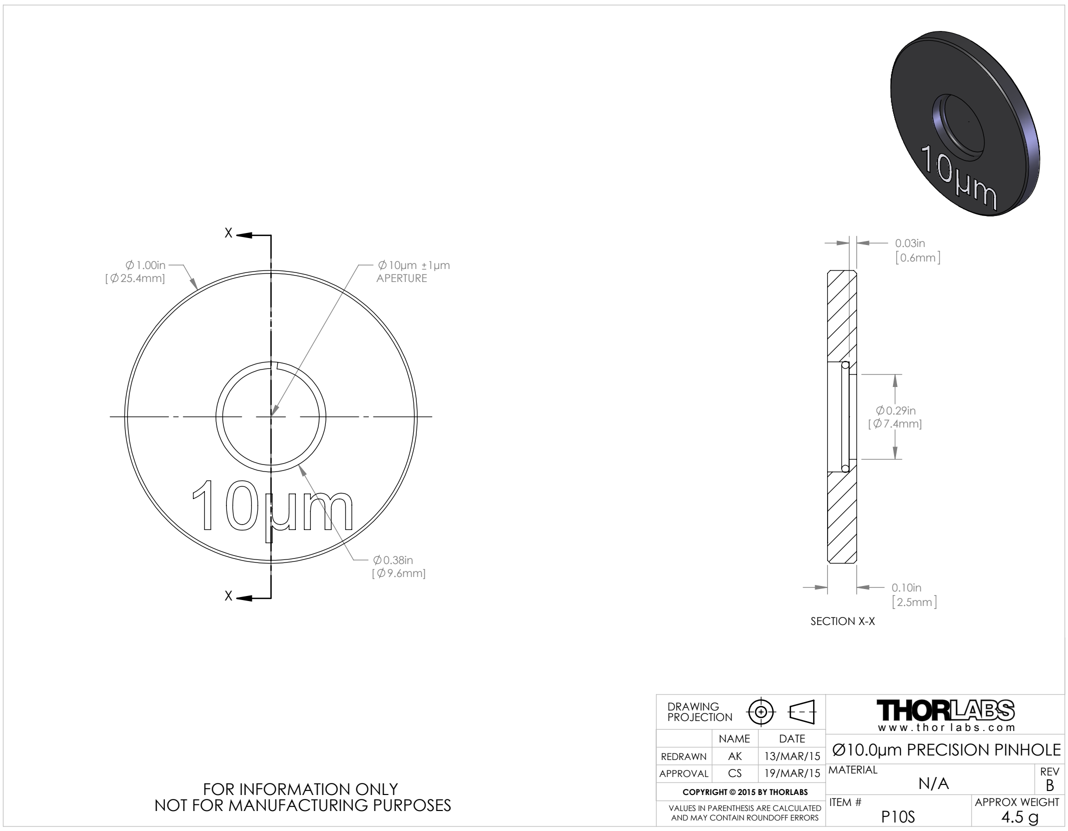
!!! mettre la photo du driver et de la modif !!!

A.2 Converging lens A220TM-A, f = 11 mm



Source : www.thorlabs.com

A.3 Pinhole 10 μm



Source : www.thorlabs.com

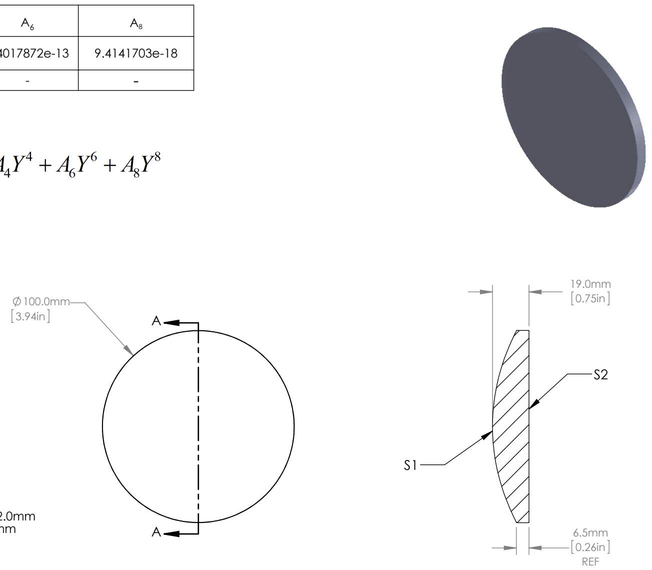
A.4 Converging lens AL100200, f = 200 mm

ASPHERIC COEFFICIENTS

	R	k	A_4	A_6	A_8
S1	102.24	-1	4.9646003e-08	7.4017872e-13	9.4141703e-18
S2	INFINITE	-	-	-	-

ASPHERIC LENS EQUATION

$$z = \frac{Y^2}{R(1+\sqrt{1-(1+k)Y^2/R^2})} + A_4Y^4 + A_6Y^6 + A_8Y^8$$



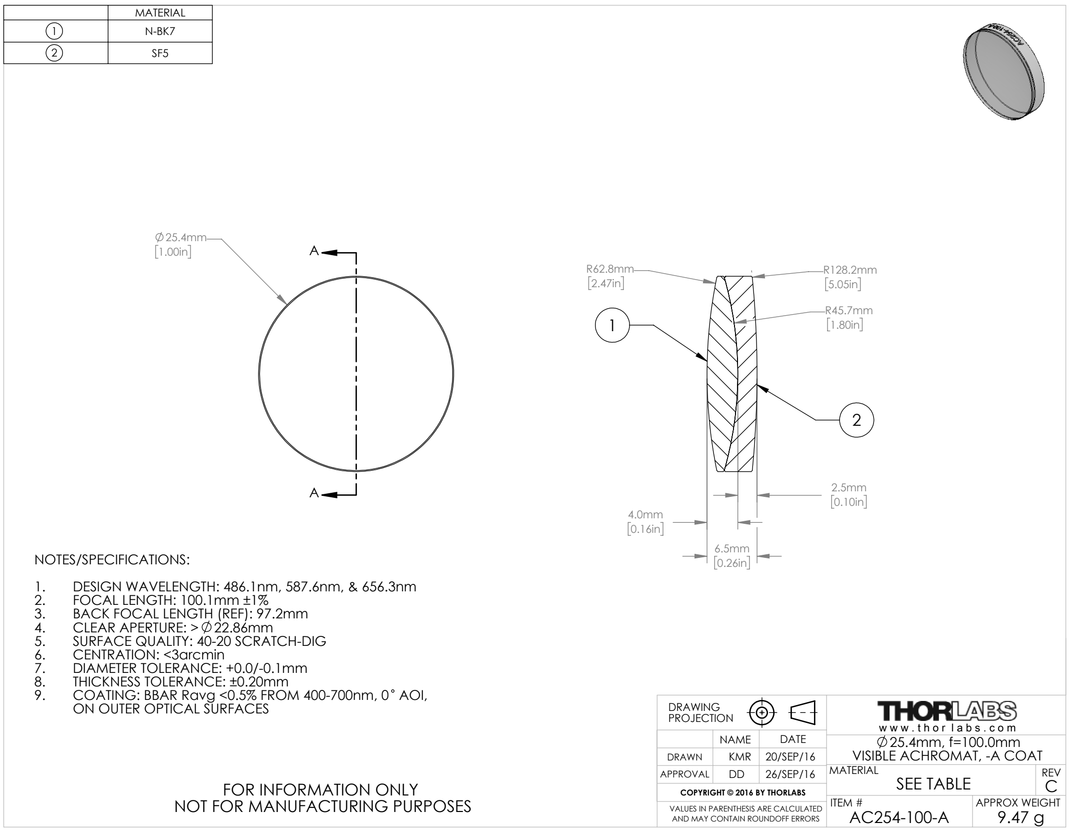
NOTES/SPECIFICATIONS:
1. CLEAR APERTURE (COLLIMATION): 92.0mm
2. CLEAR APERTURE (FOCUSING): 89.4mm
3. FOCAL LENGTH: EFL=200mm ±1%
4. Nd: 1.517 ±0.001 Vd: 64.167 ±1.0%
5. DESIGN WAVELENGTH: 780nm
6. NUMERICAL APERTURE: 0.23
7. WORKING DISTANCE (REF): 187.4mm
8. DIAMETER TOLERANCE: +0.00/-0.10mm
9. SURFACE QUALITY: 40-20 SCRATCH-DIG
10. RMS WAVEFRONT ERROR: <0.5µm
11. SAG DEVIATION (POWER)(S1): ±7.5µm
12. RMS IRREGULARITY (S1): <3 FRINGES
13. CENTRATION: <3arcmin
14. MAGNIFICATION: INFINITE

FOR INFORMATION ONLY
NOT FOR MANUFACTURING PURPOSES

DRAWING PROJECTION			THORLABS www.thorlabs.com	
	NAME	DATE	∅100.0mm ASPHERIC LENS, NA=0.23, f=200mm	
DRAWN	NE	10/OCT/14		
APPROVAL	BW	18/NOV/14	MATERIAL N-BK7	REV A
COPYRIGHT © 2014 BY THORLABS				
VALUES IN PARENTHESIS ARE CALCULATED AND MAY CONTAIN ROUNDOFF ERRORS			ITEM # AL100200	APPROX WEIGHT 0.6 kg

Source : www.thorlabs.com

A.5 Converging lens AC254-100-A, f = 100 mm



Source : www.thorlabs.com

A.6 Ximea Camera, MQ013MG-E2

Specifications:

Resolution:	1.3 MP 1280 × 1024 pixels
Sensor type:	CMOS Matrix B/W
Sensor model:	e2V EV76C560 ABT-EQV
Sensor size:	1/1.8"
Sensor active area:	6.9 × 5.5 mm
Pixel size:	5.3 µm
Bits per pixel:	8, 10
Dynamic range:	60 dB
Frame rates:	60 fps
On-chip binning:	1x1, 2x2
Image data interface:	USB 3.0
Data I/O:	GPIO IN, OUT
Power requirements:	0.9 Watt
Lens mount:	C or CS Mount
Weight:	26 grams
Dimensions WxHxD:	26 x 26 x 26 mm
Operating environment:	50 °C
Customs tariff code:	8525.80 30 (EU) / 8525.80 40 (USA)
ECCN:	EAR99

Source : www.ximea.com/en/products/usb3-vision-cameras-xiq-line/mq013mg-e2

A.7 Shack-Hartmann wavefront sensor, WFS150-5C

8 Appendix

8.1 Technical Data

8.1.1 WFS150/300

Item #	WFS150-5C	WFS150-7AR	WFS300-14AR
Microlenses			
Microlens Array	MLA150M-5C	MLA150M-7AR	MLA300M-14AR
Substrate Material	Fused Silica (Quartz)		
Number of Active Lenslets	Software Selectable		
Max. Number of Lenslets	39 x 31		19 x 15
Camera			
Sensor Type	CCD		
Resolution	max. 1280 x 1024 pixels, Software Selectable		
Aperture Size	5.95 mm x 4.76 mm		
Pixel Size	4.65 µm x 4.65 µm		
Shutter	Global		
Exposure Range	79 µs - 65 ms		
Frame Rate	max. 15 Hz		
Image Digitization	8 bit		
Wavefront Measurement			
Wavefront Accuracy [1])	λ/15 rms @ 633 nm		λ/50 rms @ 633 nm
Wavefront Sensitivity [2])	λ/50 rms @ 633 nm		λ/150 rms @ 633 nm
Wavefront Dynamic Range [3])	> 100 λ @ 633 nm		> 50 λ @ 633 nm
Local Wavefront Curvature [4])	> 7.4 mm	> 10.0 mm	> 40.0 mm
External Trigger Input			
Save Static Voltage level	0 to 30 V DC		
LOW Level	0.0 V to 2.0 V		
HIGH Level	5.0 V to 24 V		
Input current	> 10 mA		
Min Pulse Width	100 µs		
Min. Slew Rate	35 V / msec		
Common Specifications			
Optical Input	C-Mount		
Power Supply	<1.5 W, via USB		
Operating Temperature Range [5])	+5 to +35 °C		
Storage Temperature Range	-40 to 70 °C		
Warm-Up Time for Rated Accuracy	15 min		
Dimensions (W x H x D)	32.0 mm x 40.4 mm x 45.5 mm		
Weight	0.1 kg		

[1]) Absolute accuracy using internal reference. Measured for spherical wavefronts of known RoC.
[2]) Typical relative accuracy. Achievable after, and with respect to a user calibration, 10 image averages
[3]) Over entire aperture of wavefront sensor
[4]) Radius of wavefront curvature over single lenslet aperture
[5]) non-condensing

All technical data are valid at 23 ± 5°C and 45 ± 15% rel. humidity (non condensing)

 99

Source : WFS Series Operation Manual, www.thorlabs.com

Appendix B

Python Code

B.1 AlignementScriptXimeaCamera.py

```
##Script to compute the FWHM of the beam on the camera averaging
 #over "nbrImgAveraging" images and see which position minimizes it.

from ximea import xiapi
import numpy as np
from matplotlib import pyplot as plt
import scipy.optimize as opt
import datetime
import functionsXimea as fX
import seaborn as sns
import os
sns.set()
#%% instanciation

dataFolderPath =
    'C:/Users/Jojo/Desktop/PdM-HEIG/Science/data/PD/phaseScreen/alignement/'
plotFolderPath =
    'C:/Users/Jojo/Desktop/PdM-HEIG/Science/fig/PD/phaseScreen/alignement/'
#create the matrix grid of the detector CCD
x = np.linspace(0,1280,1280)
y = np.linspace(0,1024,1024)
x, y = np.meshgrid(x, y)

#initial guess for the fit depending on the position of the beam in the CCD
initial_guess = [250, 481, 706, 3, 3]

#number of image to average
nbrImgAveraging = 10

#%%data acquisition and treatment

#create instance for first connected camera
cam = xiapi.Camera()
#start communication
print('Opening camera...')
cam.open_device()
#settings
cam.set_imgdataformat('XI_MONO8') #XIMEA format 8 bits per pixel
cam.set_gain(0)
#create instance of Image to store image data and metadata
img = xiapi.Image()
#start data acquisition
```

```
print('Starting data acquisition...')
if cam.get_acquisition_status() == 'XI_OFF':
        cam.start_acquisition()

cam.set_exposure(fX.determineUnsaturatedExposureTime(cam,img,[60,10000],1))

#instanciation for the while loop
answer ='y'
i=0
relativePos = []
data = []
data_fitted = []
FWHMx = []
FWHMy = []
x0 = []
y0 = []
sigmaX0 = []
sigmaY0 = []

while answer == 'y':

    try:
        relativePos.append(float(raw_input('What is the position on the
            screw [mm] ? ')))
    except ValueError:
        print('Not a float number')

    [tmpdata,stdData] = fX.acquireImg(cam,img,nbrImgAveraging)
    data.append(tmpdata)
    #Fit the img data on the 2D Gaussian to compute the FWHM
    print('Fitting 2D Gaussian...')
    popt, pcov = opt.curve_fit(fX.TwoDGaussian, (x,y), data[i].ravel(), p0 =
        initial_guess)
    print('Fitting done')

    FWHMx.append(2*np.sqrt(2*np.log(2))*popt[3])
    FWHMy.append(2*np.sqrt(2*np.log(2))*popt[4])
    x0.append(popt[2])
    sigmaX0.append(popt[4])
    y0.append(popt[1])
    sigmaY0.append(popt[3])

    print 'Fig %d : (x,y) = (%3.2f,%3.2f), FWHM x = %3.2f, FWHM y = %3.2f'
        %(i,x0[i],y0[i],FWHMx[i],FWHMy[i])

    data_fitted.append(fX.TwoDGaussian((x, y),
        popt[0],popt[1],popt[2],popt[3],popt[4]).reshape(1024, 1280))

    #plot the beamspot
    fig, ax = plt.subplots(1, 1)
    ax.imshow(data[i], cmap=plt.cm.jet,origin='bottom',
        extent=(x.min(), x.max(), y.min(), y.max()))
    ax.contour(x, y, data_fitted[i], 5, colors='w',linewidths=0.8)
    plt.xlim( (popt[2]-4*popt[4], popt[2]+4*popt[4]) )
    plt.ylim( (popt[1]-4*popt[3], popt[1]+4*popt[3]) )
    plt.show()

```

```
    #ask if the person wants to acquire a new image to improve the alignement
    pressedkey = raw_input('Do you want to acquire an other image [y (yes)
        or n (no)]: ')
    if (pressedkey =='n'):
        answer = pressedkey
    #increase i
    i+=1

#stop data acquisition
print('Stopping acquisition...')
cam.stop_acquisition()

#stop communication
cam.close_device()

#convert list to np.array
relativePos = np.array(relativePos)
data = np.array(data)
FWHMx = np.array(FWHMx)
FWHMy = np.array(FWHMy)
x0 = np.array(x0)
y0 = np.array(y0)
sigmaX0 = np.array(sigmaX0)
sigmaY0 = np.array(sigmaY0)

#plot the FWHM vs. relPos
fig, ax = plt.subplots(1,1)
ind = np.argsort(relativePos)
ax.plot(relativePos[ind],(np.sqrt(FWHMx**2+FWHMy**2))[ind])
ax.set_xlabel('Position [mm]')
ax.set_ylabel('FWHM [px]')
ax.grid()
date = datetime.datetime.today()
if not os.path.isdir(plotFolderPath):
    os.makedirs(plotFolderPath)
plt.savefig(plotFolderPath+date.strftime('%Y%m%d%H%M%S')+'FWHM_pos.pdf')
plt.savefig(plotFolderPath+date.strftime('%Y%m%d%H%M%S')+'FWHM_pos.png')


indOfMinFWHM = np.argmin(np.sqrt(FWHMx**2+FWHMy**2))

fig, axarr = plt.subplots(1,np.size(data,0))
#plot all the images besides each other
for iImg in ind:
    axarr[iImg].imshow(data[iImg], cmap=plt.cm.jet,origin='bottom',
        extent=(x.min(), x.max(), y.min(), y.max()))
#    axarr[iImg].contour(x, y, data_fitted[iImg], 5,
    colors='w',linewidths=0.8)
    axarr[iImg].set_xlim( (x0[iImg]-12, x0[iImg]+12) )
    axarr[iImg].set_ylim( (y0[iImg]-12, y0[iImg]+12) )
    axarr[iImg].set_yticklabels('',visible=False)
    axarr[iImg].set_xticklabels('',visible=False)
    axarr[iImg].set_title('%5.3f mm'%relativePos[iImg],fontsize=8)
    if iImg == indOfMinFWHM:
        axarr[iImg].set_frame_on(True)
        for pos in ['top', 'bottom', 'right', 'left']:
            axarr[iImg].spines[pos].set_edgecolor('r')
```

```
            axarr[iImg].spines[pos].set_linewidth(2)
    else:
        axarr[iImg].set_frame_on(False)
plt.show()
date = datetime.datetime.today()

plt.savefig(plotFolderPath+date.strftime('%Y%m%d%H%M%S')+'ImgPSF.pdf')
plt.savefig(plotFolderPath+date.strftime('%Y%m%d%H%M%S')+'ImgPSF.png')


#save data
if not os.path.isdir(dataFolderPath):
    os.makedirs(dataFolderPath)
date = datetime.datetime.today()
np.save(dataFolderPath+date.strftime('%Y%m%d%H%M%S')+'data.npy',data)
np.save(dataFolderPath+date.strftime('%Y%m%d%H%M%S')+'relativePos.npy',relativePos)
```

B.2 AcquisAndSaveXimea.py

```
#%% Script to acquire images average over nbrImgAveraging images and save
    them into fits file

from ximea import xiapi
import datetime
import functionsXimea as fX
import winsound
import numpy as np

#%%instanciation
    ----------------------------------------------------------------
#number of image to average
nbrImgAveraging = 5000
numberOfFinalImages = 1

#Cropping information
sizeImg = 256

#Parameter of camera and saving
folderPathCropped =
    '../../data/PD/astigmatism/angle_study_3/wth/cropped/20/'
darkFolderPathCropped =
    '../../data/dark/astigmatism/angle_study_3/wth/cropped/20/'
folderPathFull = '../../data/PD/astigmatism/angle_study_3/wth/full/20/'
darkFolderPathFull =
    '../../data/dark/astigmatism/angle_study_3/wth/full/20/'
nameCamera = 'Ximea'
focusPos = 11.63

#Sound
duration = 1000 # millisecond
freq = 2000 # Hz

#initial guess for the fit depending on the position of the beam in the CCD
initial_guess = [250, 468, 954, 3, 3]

```

```python
#------------------------------------------------------------------------------
#%% data acquisition ----------------------------------------------------------

#Opening the connection to the camera
cam = xiapi.Camera()
cam.open_device()
cam.set_imgdataformat('XI_MONO8') #XIMEA format 8 bits per pixel
cam.set_gain(0)

img = xiapi.Image()
if cam.get_acquisition_status() == 'XI_OFF':
    cam.start_acquisition()
#%% exposition
cond = 1
while bool(cond):
    source = ''
    winsound.Beep(freq, duration)
    source = int(raw_input('Is the source turned on and at focus point (usually %5.3f mm) (yes = 1) ? '%focusPos))
    if source == 1:
        cond = 0
    else:
        print 'Please turn on the source and place the camera on the focus point (%5.3f mm)'%focusPos

if bool(source):
    #Set exposure time
    cam.set_exposure(fX.determineUnsaturatedExposureTime(cam,img,[1,10000],1))
    #get centroid
    centroid = fX.acquirePSFCentroid(cam,img,initial_guess)
    print 'centroid at (%d, %d)' %(centroid[0],centroid[1])

#%%Acquire images at different camera position

acquire = 1
while bool(acquire):
    cond = 1
    while bool(cond):
        dark = ''
        winsound.Beep(freq, duration)
        dark = int(raw_input('Is the source turned off (yes = 1) ? '))
        if dark == 1:
            cond = 0
        else:
            print 'Please shut down the source.'

    winsound.Beep(freq, duration)
    pos = float(raw_input('What is the position of the camera in mm focused (%5.3f mm) dephase 2Pi (pos+ = %5.3f mm, pos- = %5.3f) ? '%(focusPos,focusPos+3.19,focusPos-3.19)))

    if bool(dark):
        print 'Acquiring dark image...'
        # Acquire dark images
        [darkData,stdDarkData] = fX.acquireImg(cam,img,nbrImgAveraging)
        print 'Cropping'
```

```
        [darkdataCropped,stddarkDataCropped] =
            fX.cropAroundPSF(darkData,stdDarkData,centroid,sizeImg,sizeImg)
        print 'saving'
        fX.saveImg2Fits(datetime.datetime.today(),darkFolderPathCropped,nameCamera,darkdataCro
        fX.saveImg2Fits(datetime.datetime.today(),darkFolderPathFull,nameCamera,darkData,stdDa

    #Acquire images -------------------------
    cond = 1
    while bool(cond):
        source = ''
        winsound.Beep(freq, duration)
        source = int(raw_input('Is the source turned on (yes = 1) ? '))
        if source == 1:
            cond = 0
        else:
            print 'Please place turn on the camera'

    if bool(source):
        print 'Acquiring images...'
        # Acquire focused images
        for iImg in range(numberOfFinalImages):
            imgNumber = iImg+1
            print 'Acquiring Image %d'%imgNumber
            [data,stdData] = fX.acquireImg(cam,img,nbrImgAveraging)
            print 'Cropping'
            [dataCropped,stdDataCropped] =
                fX.cropAndCenterPSF(data-darkData,stdData+stdDarkData,sizeImg,initial_guess)
            print 'Saving'
            fX.saveImg2Fits(datetime.datetime.today(),folderPathCropped,nameCamera,dataCropped,
            fX.saveImg2Fits(datetime.datetime.today(),folderPathFull,nameCamera,data-darkData,s

    cond = 1
    while bool(cond):
        acquire = ''
        winsound.Beep(freq, duration)
        acquire = int(raw_input('Do you want to acquire at an other camera
            position (yes = 1, no = 0) ? '))
        if acquire == 1:
            cond = 0
        elif acquire == 0:
            cond = 0
        else:
             print 'please answer with 0 or 1 for no or yes, respectively'


##Stop the acquisition
cam.stop_acquisition()
cam.close_device()

print 'Acquisition finished'
```

Bibliography

Bouxin, A. (2017). “Phasor diversity to measure the static aberrations of an optical system”. MA thesis. HEIG-VD.

Goodman, Joseph W. (1968). *Introduction to Fourier Optics*. Ed. by L. Cox and John M. Morris. 2nd. McGraw-Hill Companies, Inc.

Mugnier, Laurent M., Amandine Blanc, and Jérôme Idier (2006). “Phase Diversity: A Technique for Wave-Front Sensing and for Diffraction-Limited Imaging”. In: ed. by Peter Hawkes. Vol. 141. Advances in Imaging and Electron Physics. Elsevier, pp. 1 –76. DOI: https://doi.org/10.1016/S1076-5670(05)41001-0. URL: http://www.sciencedirect.com/science/article/pii/S1076567005410010.

Thorlabs (2017). *Principles of Spatial Filters*. Thorlabs. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1400.