ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

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

January 13, 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

ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Abstract

Physics
Basic Sciences

Master in Applied Physics

Evaluation of Optical Aberrations using Phase Diversity

by Jordan VOIRIN

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

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

Contents

Declaration of Authorship iii

Abstract vii

Acknowledgements ix

1 Introduction 1

2 Phase Diversity Experiment 3
- 2.1 Experimental Setup . . . 3
  - 2.1.1 Light source . . . 3
  - 2.1.2 Entrance pupil . . . 4
  - 2.1.3 Pupil imaging system . . . 4
  - 2.1.4 Detectors . . . 4
- 2.2 Data Acquisition . . . 5
- 2.3 ONERA algorithm characterization . . . 5
- 2.4 Results . . . 5
  - 2.4.1 Parallel plane plate . . . 5

A Optical Component Datasheets 7
- A.1 Pigtailed laser diode . . . 8
  - A.1.1 Power supply modification . . . 9
- A.2 Converging lens A220TM-A, f = 11 mm . . . 9
- A.3 Pinhole 10 μm . . . 10
- A.4 Converging lens AL100200, f = 200 mm . . . 11
- A.5 Converging lens AC254-100-A, f = 100 mm . . . 12
- A.6 Ximea Camera, MQ013MG-E2 . . . 13
- A.7 Shack-Hartmann wavefront sensor, WFS150-5C . . . 14

B Python Code 15
- B.1 AlignementScriptXimeaCamera.py . . . 15
- B.2 AcquisAndSaveXimea.py . . . 18

Bibliography 21

List of Figures

2.1 Schema of the experimental setup . . . 3
2.3 Source schema and pinhole effect on the beam. . . . 4
2.2 Wavefront curvature . . . 4
2.4 PSFs example of an alignment procedure . . . 5

List of Tables

2.1 Optical Components 3

List of Abbreviations

FWHM	**F**ull **W**idth **H**alf **M**aximum
PSF	**P**oint **S**read **F**unction
WFS	**W**ave**F**ront **S**ensor

Physical Constants

Speed of Light $c_0 = 2.997\,924\,58 \times 10^8\ \mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

Introduction

???

Chapter 2

Phase Diversity Experiment

2.1 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.

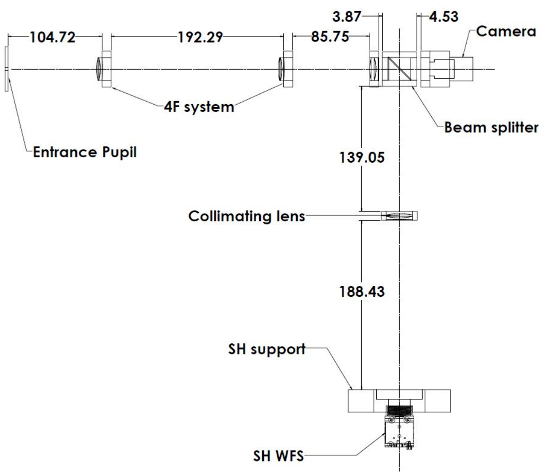


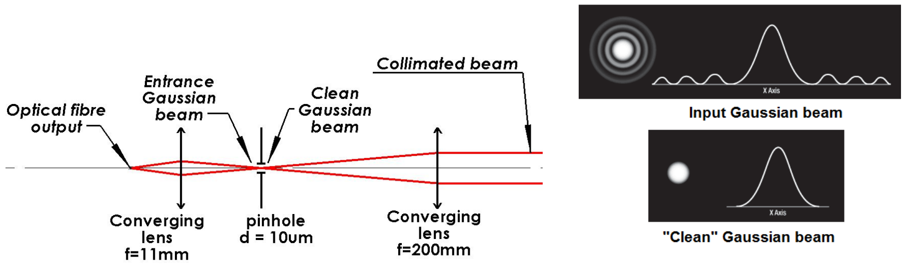
FIGURE 2.1: Experimental setup schema with the relevant distances, (Bouxin, 2017).

2.1.1 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

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.

than the telescope size, r, see Fig. 2.2. The source of our experiment must then be characterized by a planar wavefront.

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.

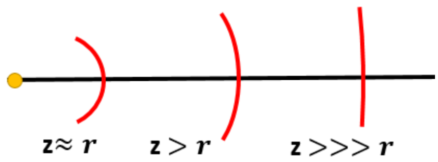


FIGURE 2.2: Wavefront curvature for different source's distances, z. r represents the characteristic size of the arc of interest.

2.1.2 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.

2.1.3 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 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.

2.1.4 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

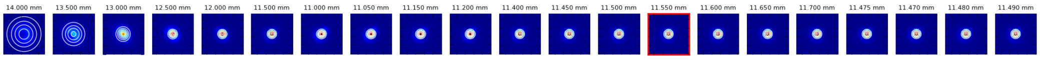


FIGURE 2.4: PSFs example of an alignment procedure

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.

2.2 Data Acquisition

2.2.1 Ximea Camera

The PSFs are acquired using a python library, `pyXimea`[1], available on GitHub. The acquisition is done following these steps :

1. Find the focus point on the camera using `AlignementScriptXimeaCamera.py`, see Appendix B.1. This script saves consecutively PSFs at different camera's position and computes their FWHM. And it highlights the minimum FWHM and the camera's position, see Figure 2.4.

2.

2.3 ONERA algorithm characterization

2.4 Results

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

2.4.1 Parallel plane plate

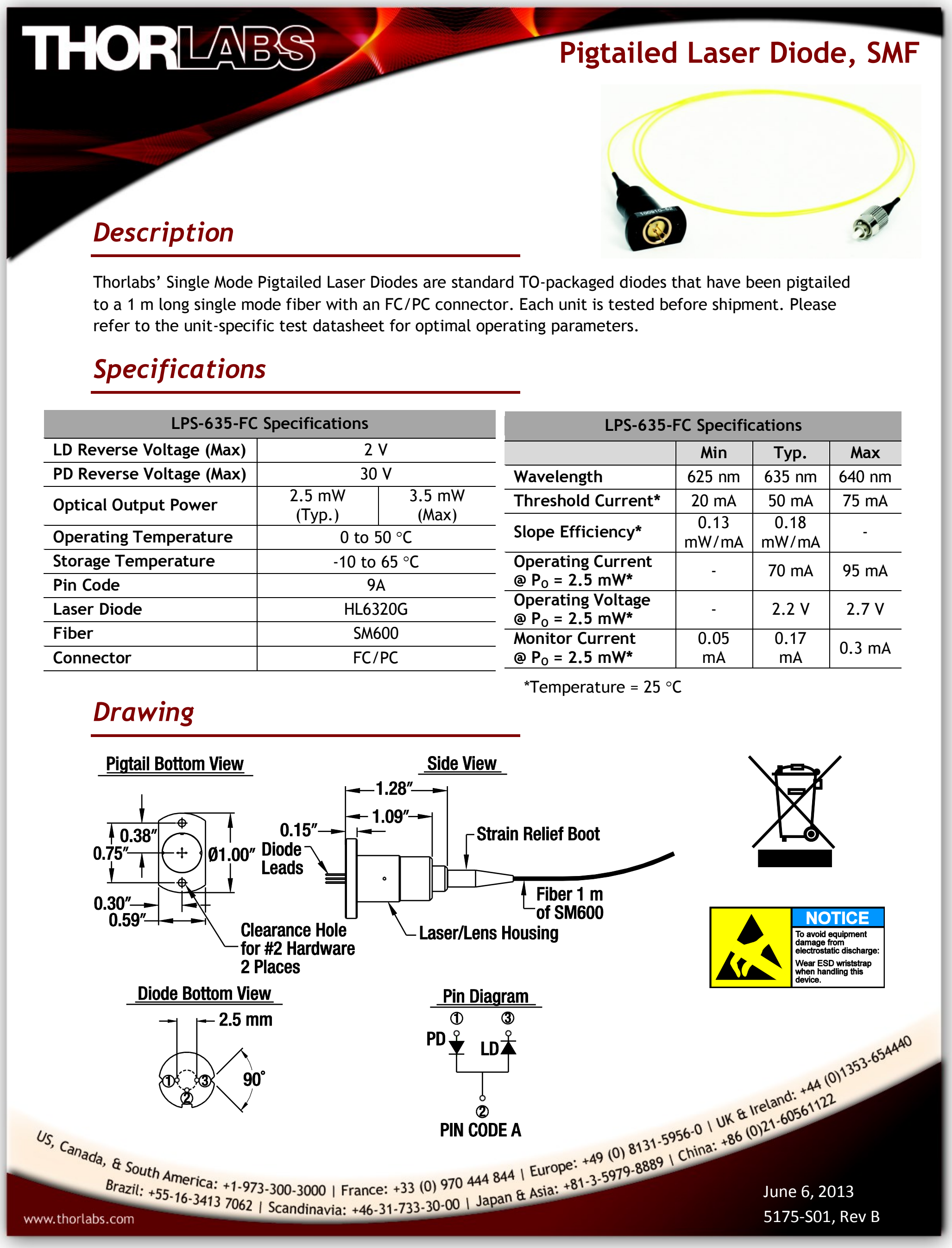
The first source of aberration studied in this work is a tilted parallel plane plate which is used as a calibrated source of astigmatism.

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

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

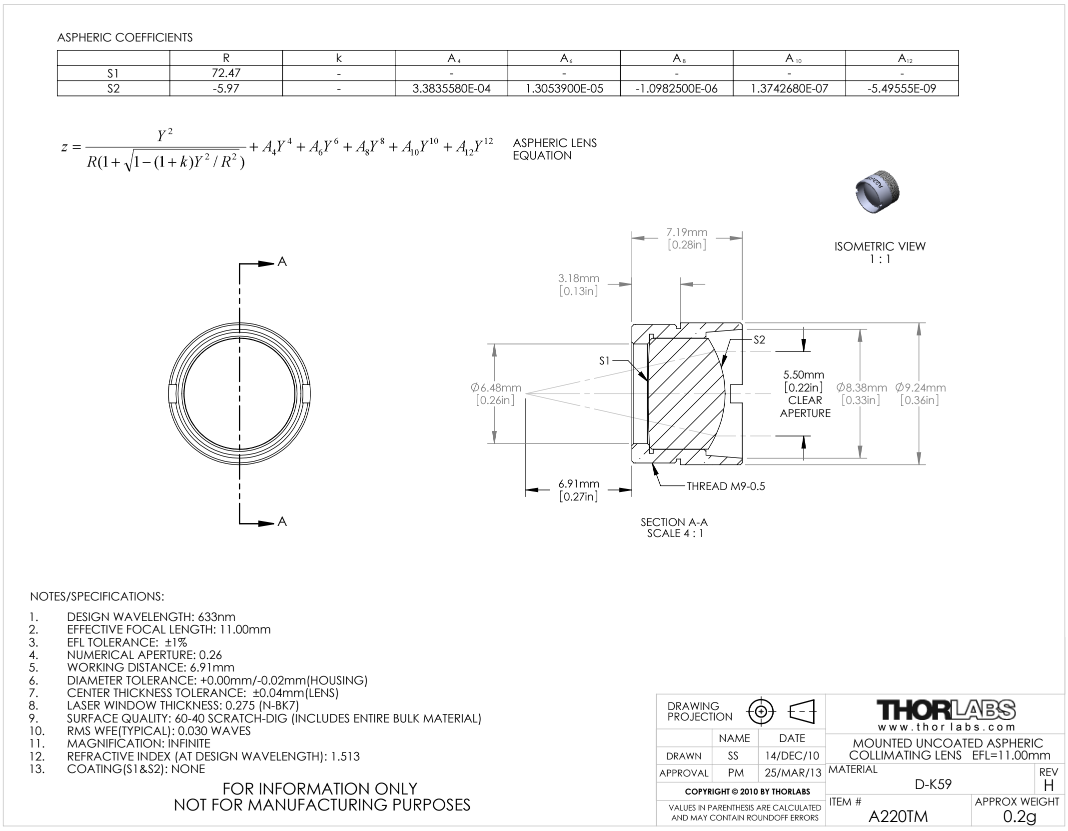
Source : www.thorlabs.com

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 !!!

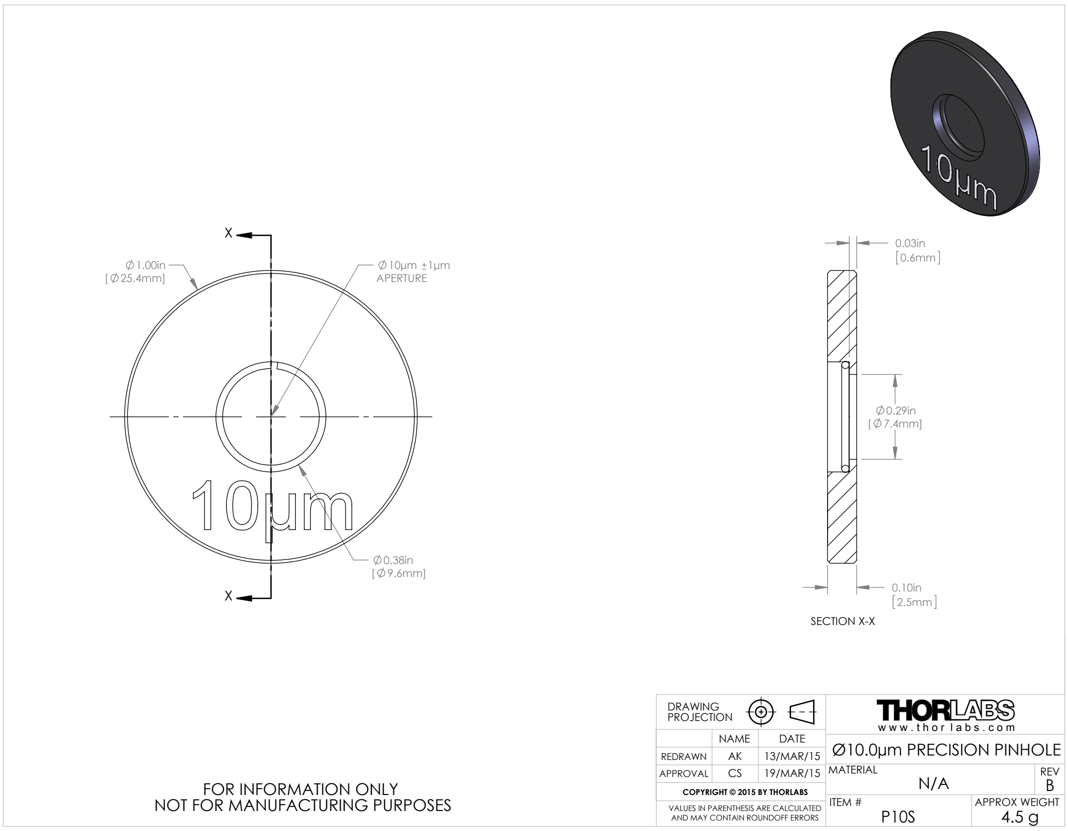
A.2 Converging lens A220TM-A, f = 11 mm



$$z = \frac{Y^2}{R(1+\sqrt{1-(1+k)Y^2/R^2})} + A_4Y^4 + A_6Y^6 + A_8Y^8 + A_{10}Y^{10} + A_{12}Y^{12}$$

Source : www.thorlabs.com

A.3 Pinhole 10 μm



Source : www.thorlabs.com

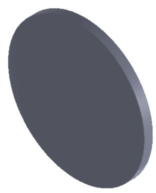
A.4 Converging lens AL100200, f = 200 mm

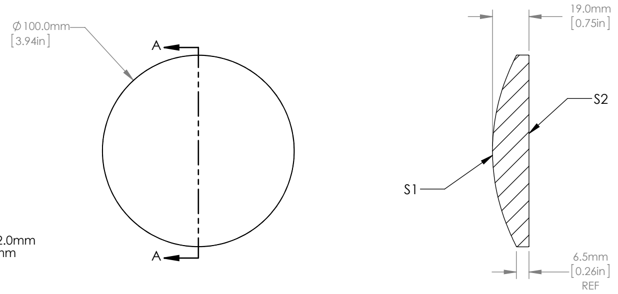
ASPHERIC COEFFICIENTS

	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

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





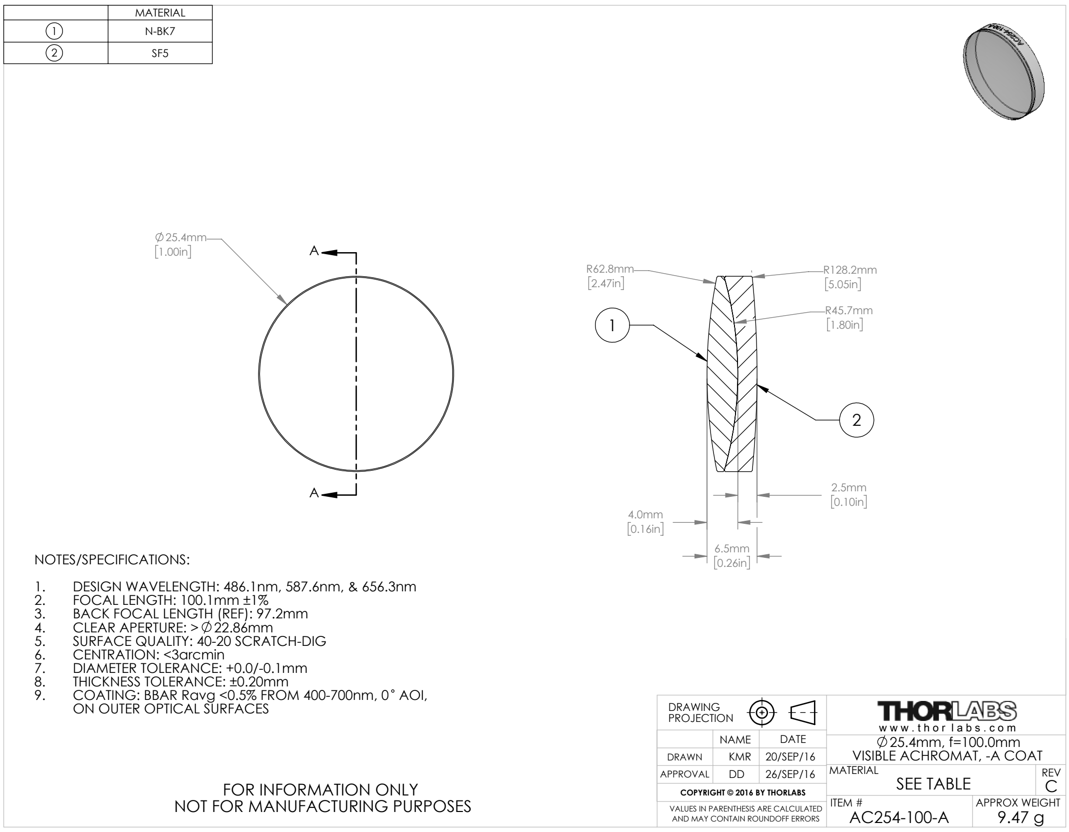
NOTES/SPECIFICATIONS:
1. CLEAR APERTURE (COLLIMATION): 92.0mm
2. CLEAR APERTURE (FOCUSING): 89.4mm
3. FOCAL LENGTH: EFL=200mm ±1%
4. Nd: 1.517 ±0.001 Vd: 64.167 ±1.0%
5. DESIGN WAVELENGTH: 780nm
6. NUMERICAL APERTURE: 0.23
7. WORKING DISTANCE (REF): 187.4mm
8. DIAMETER TOLERANCE: +0.00/-0.10mm
9. SURFACE 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	MATERIAL N-BK7	REV A
APPROVAL	BW	18/NOV/14		
COPYRIGHT © 2014 BY THORLABS			ITEM # AL100200	APPROX WEIGHT 0.6 kg
VALUES IN PARENTHESIS ARE CALCULATED AND MAY CONTAIN ROUNDOFF ERRORS				

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

8.1 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)

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

Appendix B

Python Code

B.1 AlignementScriptXimeaCamera.py

```python
##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]

```

```
#---------------------------------------------------------------------------
#%% 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.

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