University of Crete

Master Thesis

White light interferometry in vision science applications

Evangelos Tzardis

A thesis submitted in partial fulfillment of the requirements
for the Interdepartmental Postgraduate Programme in Optics and Vision

of the

University of Crete, Heraklion
in collaboration with the
Institute of Electronic Structure and Laser, FORTH

thesis committee
Associate Professor Dimitris Papazoglou (Supervisor)
Professor Miltiadis Tsilibaris
Professor Michael Taroudakis

February, 2020

Acknowledgements

I would like to express my everlasting esteem and gratitude towards my thesis supervisor Prof. Dimitris Papazoglou who gave me the inspiration and the will to learn and work on new for me scientific fields.

A BIG thanks as well to Dimitris Mansour and Matina Vlachou who gave me the necessary insight of the used experimental equipment, offered inconceivable aid and of course provided a friendly collaborative environment.

Last but not least, a HUGE thanks to my family and my close friends for their continuous and constant support throughout my university life and especially when emerging situations stumped me.

Abstract

This thesis focuses on the use of white light scanning interferometry methods to study the topology/topography of surfaces of transparent materials that are met in vision science applications in a non-invasive way. The focus of this study is to measure the surface profile of different kinds of optical elements and additionally to deduce quantitative information regarding the surface such as surface roughness for every element and curvature specifically for lenses. The main objects under study are rigid contact lenses (CL) of various radii of curvature (RoC) and typical glass lenses of different focal distances. Additionally, a stylus calibration refererence sample and a set of polymer films were measured. Concerning one of the most important factors of a device for measurement and analysis, the spatial resolution of the current device in terms of surface elevation is in the order of tenths of nanometers. Due to the nature of the used scanning method, there is no corresponding time resolution, since the procedures of measurement and analysis are consecutive. The results of relatively flat surfaces render the device stable, reliable, reproducible and within the manufacturer's tolerances whereas for spherical ones there exists an accuracy error in the range of 5 to 20% in RoC estimation.

Περίληψη

Η παρούσα μεταπτυχιακή εργασία μελετά την τοπολογία/τοπογραφία επιφανειών διάφανων υλικών που συναντώνται σε εφαρμογές της επιστήμης της όρασης. Η μελέτη αυτή κάνει χρήση μη επεμβατικών μεθόδων συμβολομετρίας λευκού φωτός με σάρωση οπτικού δρόμου, με στόχο τη μέτρηση του προφίλ επιφάνειας διαφόρων οπτικών στοιχείων. Επιπλέον, υπολογίζονται διάφορα ποσοτικοποιημένα χαρακτηριστικά επιφάνειας όπως η τραχύτητα για κάθε οπτικό στοιχείο και η καμπυλότητα ειδικά για τους φακούς. Τα κύρια αντικείμενα μελέτης είναι σκληροί φακοί επαφής διαφόρων ακτινών καμπυλότητας και τυπικοί γυάλινοι φακοί διαφορετικών εστιακών αποστάσεων. Επιπρόσθετα, μετρήθηκαν ένα δείγμα αναφοράς για βαθμονόμηση stylus και ένα σετ φίλμ πολυμερών. Σχετικά με τους πιο σημαντικούς παράγοντες μίας συσκευής για τη μέτρηση και ανάλυση, η χωρική ανάλυση της παρούσας συσκευής, όσον αφορά τα επιμέρους ύψη της επιφάνειας, είναι της τάξης των δεκάδων νανομέτρων. Λόγω της φύσης της χρησιμοποιούμενης μεθόδου σάρωσης, δεν υπάρχει αντίστοιχη χρονική ανάλυση της συσκευής, καθώς οι διαδικασίες της μέτρησης και της ανάλυσης είναι διαδοχικές. Τα αποτελέσματα των σχετικά επίπεδων επιφανειών καθιστούν τη συσκευή σταθερή, αξιόπιστη, αναπαράξιμη, και εντός των ανοχών κατασκευής, ενώ για τις σφαιρικές επιφάνειες υπολογίζονται σφάλματα ακρίβειας στην εκτίμηση των καμπυλοτήτων σε εύρος 5 με 20%.

Contents

List of Figures

List of Abbreviations

CL	**C**ontact **L**ens
RoC	**R**adius of **C**urvature
OPL	**O**ptical **P**ath **L**ength
OPD	**O**ptical **P**ath **D**ifference
LED	**L**ight **E**mitting **D**iode
AFM	**A**tomic **F**orce **M**icroscopy
SEM	**S**canning **E**lectron **M**icroscope
FWHM	**F**ull **W**idth **H**alf **M**aximum
DAC	**D**igital-to-**A**nalog **C**onverter
IDE	**I**ntegrated **D**evelopment **E**nvironment
API	**A**pplication **P**rogramming **I**nterface
SDK	**S**oftware **D**evelopment **K**it
FSM	**F**inite **S**tate **M**achine
GUI	**G**raphical **U**ser **I**nterface
ROI	**R**egion **O**f **I**nterest
POI	**P**oint **O**f **I**nterest

Chapter 1

Introduction

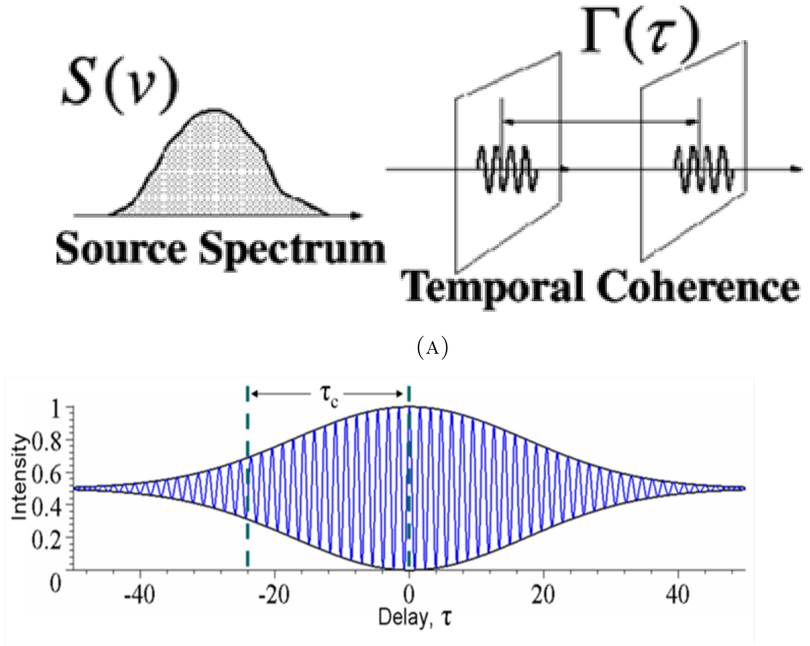
1.1 Purpose

The goal of this thesis is to focus on white light interferometry from the aspect of topography and produce 2D and 3D surface elevation maps. Using the Michelson setup and the coherence scanning interferometry method, various flat and curved reflective samples are measured. Particularly, a set of optical lenses, contact lenses and additionally a set of polymer films and a stylus calibration reference sample are measured. By post-processing the results, the local surface roughness and/or flatness are extracted for flat samples and the radius of curvature of the front surface for the curved ones.

1.2 Method: Principle

When seeking ways to measure surface topology changes in general, one may easily discern two different procedures. Firstly, the examiner can make a *tactile measurement* using a highy sensitive, in terms of the specific occasion, probe. Nowadays and in situations that a nanometer-order surface height resolution is required, the role of the tactile measurement device is played by the *contact stylus instrument* (see Sec: 2.1:*invasive / contact techniques*). Secondly, the examiner can make a *non-invasive measurement* using methods of imaging. Just capturing a photograph of the surface does not provide any hard data (quantifiable information) because the detected intensity of the reflected light from the surface does not depend on the surface height but rather to the reflectivity. Therefore, a more sophisticated method must be applied and be searched in the properties of light.

A way of making the surface heights dependent to the detected light intensities, is by exploiting the natural phenomenon of light interference. Assuming *monochromatic light* and no losses, a wave of light propagating in space and time(e.g the electrical component) is constantly oscillating in a specific frequency, but its intensity is constant with respect to space and time. Adding a second wave with the same frequency and polarization not vertical to the first wave, to interfere with the first wave, the resulted wave will simply be the superposition of the two component waves. Furthermore, it can be shown that the resulted intensity of light is not simply the sum of the two individual intensities. There is also an interference adding term (fringes) that depends sinusoidally on the phase difference of the two components. This phase difference is the key to understanding the use of light interference for surface topology applications. It is actually the lag between the two component signals and more



(A)

(B)

Figure 1.1: Fringe visibility is directly related to the spectral distribution of the light source. The more broadband the source is, the less visible the fringes are (A). Fringe visibility drops as the OPD of the sample wavefront increases (B).
Source: [1]

specifically their optical path difference (OPD)[1] times their wavenumber. Consequently, the intensity of the interference signal depends sinusoidally to the phase difference.

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2}\, cos(k\, OPD) \tag{1.1}$$

Towards a topology related approach, one can imagine two waves of the same initially phase being reflected from a sample comprised only by two areas at different elevations (compared to a reference ground). The area with the highest elevation will cause the incoming wave to travel a smaller optical path compared to the other area. Therefore, this means that a phase step that depends on the height difference between the two areas of the sample is engraved on the reflected sample wavefront. At this point, if the reflected wave is put to interfere with a reference wave, an interference signal will be produced, thus incorporating the OPD of the sample wavefront. Consequently, in order to gauge the elevation of the sample, the interference signal must be captured. This signal that from here on will be called interferogram, can be analysed so that the phase difference and consequently the height distribution is extracted.

All the above specifically refer to monochromatic light rays. In realistic occasions, light has a finite bandwidth. In general, the intensity contrast of the interference pattern or otherwise the fringe visibility is directly related to the spectral distribution of the light source. The more broadband the source is, the less visible the fringes are.

[1] optical path difference: difference in travelled distance times the refractive index of the propagation medium

Specifically, the contrast drops as the OPD of the sample wavefront increases (see Fig. 1.1) [1]. This is a consequence of the temporal coherence of the light source. Coherence is defined as the phase correlation between the interfering wavefronts with respect to their delay in time or equivalently to their OPD.

1.3 Method: Implementation

The wavelength of visible light is approximately half a micrometer (nominal value for green light). Very small changes in the OPD in the order of the wavelength of the source of the interfering wavefronts produce measurable changes in the intensity of the interference pattern. As a result, optical interferometry permits extremely accurate measurements. Following the principles of interferometry as stated above, the measurement of the height distribution of a sample surface is related to the measurement of the OPDs of each sample point with respect to a reference wave (usually a plane wave). This can be accomplished using a Michelson interferometer, an optical arrangement named after its inventor, Nobel prize Laureate, Albert Abraham Michelson.

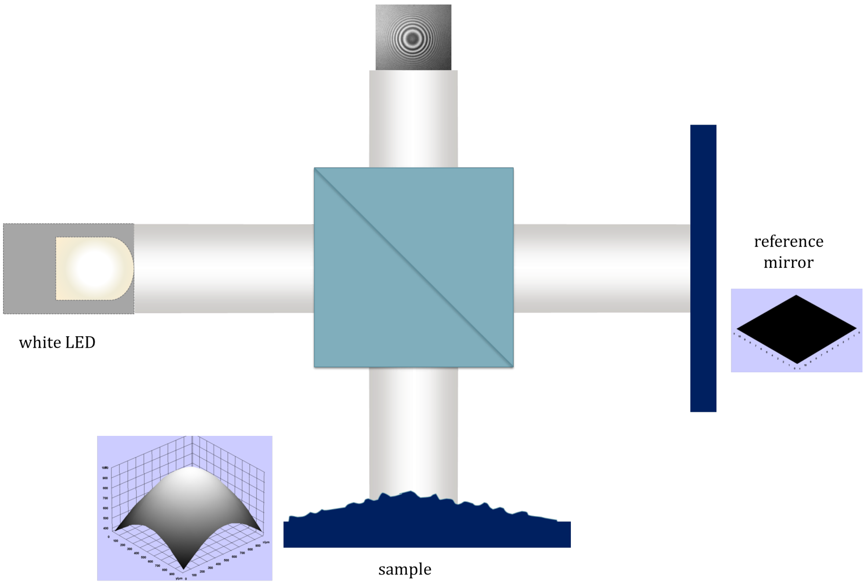


FIGURE 1.2: Interference construction: Michelson setup

In such a device, light is emitted towards a beam splitter which in turn splits the initial light beam to two different paths. The first one is the path "beam splitter-sample-beam splitter" and the second one is "beam splitter-reference mirror-beam splitter". When the two individual beams meet in the beam splitter they interfere with each other. The interference pattern is then captured on a camera.

In this thesis, the implementation of *Coherence Scanning Interferometry* is followed using white light illumination.

COHERENCE SCANNING INTERFEROMETRY (CSI)

As stated above, the intensity of interference fringes is modulated in space to such a degree that fringe contrast drops while moving away from zero OPD. For a conventional white LED, the fringe envelope drops quickly. For the specific white led source used, it drops to half the intensity (Full Width at Half Maximum (FWHM)) in just $2.2\mu m$ of OPD. Therefore, the position corresponding to zero OPD can be easily identified.

In a sample, each sample point will actually have its corresponding zero OPD appear in a different position compared to the others. In this context the OPD refers to the optical path difference to a flat reference plane (mirror of the interferometer reference arm). This position will depend on the height of the sample point. Hence, to deduce the sample heights, the positions of the zero OPD of each sample point must be identified. In particular, this is done with the aid of a scanning micromovement system. Eventually, **the absolute height of each sample point** (see Fig. 1.3) **equals the scan displacement of the reference plane in order to reach zero OPD between them** (see Fig. 1.4).

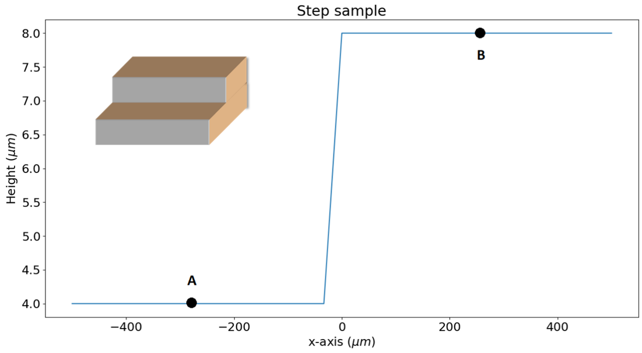


FIGURE 1.3: Sample with a single abrupt step

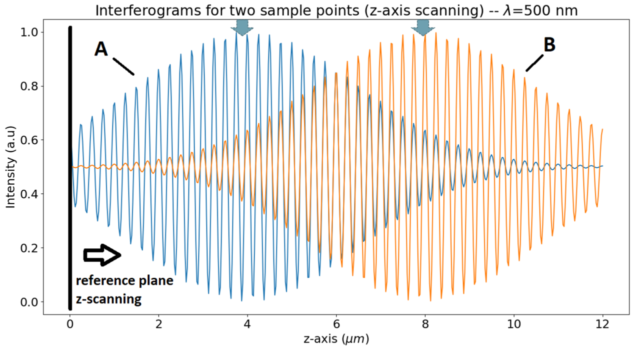


FIGURE 1.4: Corresponding interferograms of point A and B. Contrast of fringes maximizes when OPD of each point with the reference plane is equal to zero. By finding the OPL that the reference plane must travel in order to reach a zero OPD with a sample point, the elevation of the sample point can be deduced. It is half the traveld OPL.

Chapter 2

Background Information

2.1 Various methods

There is a variety of approaches to study topography in general. Which method(s) to use depends on the scale and size of the area under study. There are two big categories concerning the existence or absence of physical contact between the measuring device and the specimen:

Invasive / Contact Techniques

STYLUS PROFILOMETER

The stylus profilometer is a fundamentally linear profiling method [2]. It has been used to measure surface texture for over one hundred years and is still widely used in the manufacturing industry. A stylus instrument consists of a sharp tip placed in contact with the surface being measured. By scanning the tip across the surface and monitoring its response to surface heights, it is possible to measure surface topography. A schema of a stylus instrument is shown in Fig: 2.1 [3]. Stylus instruments can have vertical resolutions below 1 nm, although values of tens of nanometers are more common in industry. As the stylus tip is scanned across the surface, its vertical displacement is recorded and converted into an electrical signal using an electromechanical transducer. The tip plays a critical role in the performance of a stylus instrument, as it is in physical contact with the surface.

The stylus tip is usually made of diamond but other materials, such as aluminium oxide, are often employed depending of the material of the surface being measured. Other parameters that should be taken into account are the shape and size of the stylus tip, which directly affect the spatial frequency response of the instrument. Depending on the application, the stylus tip can have different geometries; the most frequently used has a conical shape with a rounded contacting edge.

The widespread use of stylus profilometers is justified by the fact that they are surface independent. Because the stylus is in contact with the surface, this method is not sensitive to surface reflectance or color. In addition to this, no mathematical modeling or theory background is required.

Besides the high precision in terms of vertical displacement, the main drawback of this device apparently is the physical contact with the samples, which may cause surface deformation if they are soft and/or fragile. With respect to the duration of the measurement, the key to determining the time resolution is the driving mechanism and the data signal sampling rate. Focusing on the driving mechanism, except for its inherent speed limit, one must take into account that large stylus tip speed will be risky for lateral deflections of the specimen surface and/or surface deformations. A typical driving mechanism speed is $0.1mm/s$. If for example, 1000 points are required for a profile of $1mm$, then the measurement will take $10s$. Escalating this, if a square grid of points is required for an areal measurement of a few millimeters-wide surfaces, then the time of measurement will reach the hour. Consequently, the line scanning nature of this device along with the limited time resolution precludes its use in production or in-line applications.

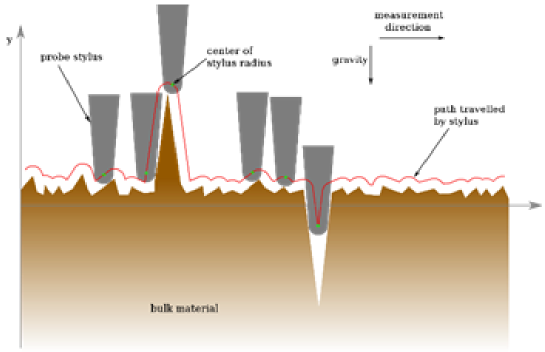


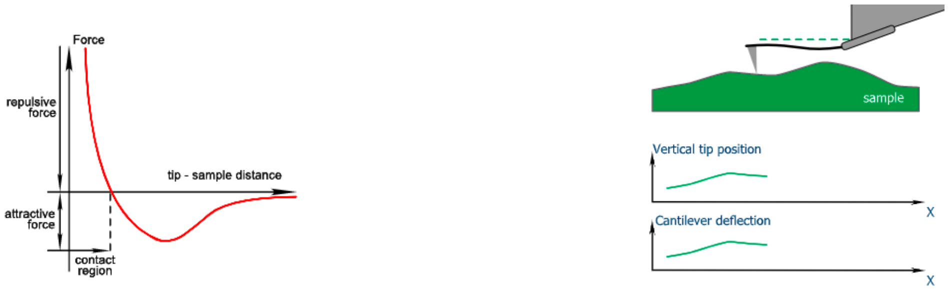
FIGURE 2.1: Stylus function and travelling path
Source: University of Cambridge [3]

CONTACT ATOMIC FORCE MICROSCOPY (AFM)

Atomic force microscopy (AFM) is a *linear profiling* method which was invented in 1985 by G. Binnig, C.F.Quate and Ch. Gerber [4]. It has successfully achieved many outstanding results on micro-/nanoscales and even on atomic/molecular scales by simple contact measurements, although contact AFM cannot achieve atomic resolution in a stable manner. The basic principle is that under interaction with sample surface, a macroscopic cantilever (rigid structural element, such as a beam or a plate, anchored at one end to a support from which it protrudes) provided with sharp tip can be bended by atomic forces to sufficiently large amount to be measured by the common facilities (Fig. 2.2) [5].

A laser beam is transmitted to and reflected from the cantilever for measuring the cantilever orientation. The reflected laser beam is detected with a position-sensitive detector, preferably a bi-cell detector[1]. The output of the bicell is provided to a

[1] photo detector with two active photodiode areas, used to measure extremely small changes in the position of a light beam



(A) Idealized sketch of tip-sample forces

(B) Contact AFM scanning. (Image shows contact height mode

FIGURE 2.2: Contact Atomic Force Microscopy principle
Source: NTMDT-SI [5]

computer for processing of the data for providing a topographical image of the surface with atomic resolution.

Cantilever can be bended not only by the direct contact forces under the tip-sample surface interaction, but also by far-ranging forces. Van der Waals attraction forces, capillary, electrostatic and repulsion forces at the point where the tip touches the sample and forces acting upon the tip from the deformed cantilever compensate each other in equilibrium. Repulsion force F acting upon the tip is related to the cantilever deflection value x under Hooke's law: F = -kx, where k is cantilever spring constant. Two submodes of candilever operation are significant. *Constant height mode*: the scanner of the microscope maintains fixed end of cantilever on the constant height value, thus the topography of the sample is reflected by the deflection of the cantilever under scanning. *Constant force mode*: the deflection of the cantilever is maintained by the feedback circuitry on the preset value, thus the topography of the sample is reflected by the vertical displacement of the scanner under scanning.

The main advantage of Constant Height mode is high scanning speeds. It is restricted only by resonant frequency of the cantilever. Regarding Constant Force mode, the main advantage is the possibility to measure with high resolution simultaneously with topography some other characteristics such as friction forces, spreading resistance.

The main disadvantage lies in the restrictions concerning the nature of the sample surface. Samples must be sufficiently smooth. When exploring soft samples (like polymers, biological samples) they can be destroyed by the scratching because the probe scanning tip is in direct contact with the surface. Especially, under scanning soft samples with relatively high relief the pressure upon the surface varies and the local flexure of the sample surface varies simultaneously. As a result acquired topography of the sample can prove distorted. Possible existence of substantial capillary forces imposed by a liquid adsorption layer can decrease the resolution.

Non-Invasive Techniques

NON-CONTACT ATOMIC FORCE MICROSCOPY (NC-AFM)

The first Noncontact AFM (NC-AFM) was designed by Martin et al. in 1987 [6]. The absence of repulsive forces (presenting in Contact AFM) in NC-AFM permits its use in the imaging of "soft" samples. The tip of the instrument does not have contact with the sample surface, which requires the utilization of attractive (interatomic) forces. The corresponding detection scheme exploits the change in the amplitude, A, of the oscillation of a cantilever due to the interaction of the tip with a sample.

NC-AFM uses a piezoelectric modulator to obtain the information regarding the surface topography. Piezoelectric modulator enables to keep minimum distance to the sample surface by Van der Waals forces between the tip and the sample surface. Therefore, all the time the distance between the tip and the sample surface remains in the same level by z-servo feedback loop control system. Just like in the Contact AFM case, laser light is sent to the cantilever, and then reflected back to photodetector which monitors the surface topography information (Fig. 2.3) [7].

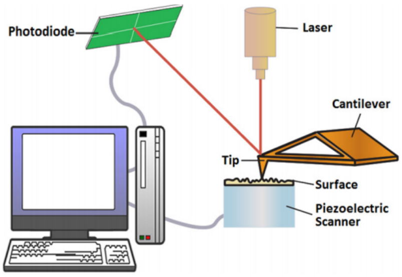


Figure 2.3: Non-Contact AFM scanning.
Source: Topological Evaluation of Surfaces in Relation to Surface Finish, Chapter by P Demircioglu, Adnan Menderes University [7]

STEREO SCANNING ELECTRON MICROSCOPE (SEM) - TOPOGRAPHIC MODE

The Scanning Electron Microscope (SEM) is one of the electronic microscopy instruments family with which a sample surface can be probed by utilizing electron beams [8]. In correspondence with optical microscopes that use optical lenses to image an object, at a SEM electrons and electromagnetic lenses are used to create the image of a sample surface onto a detector. Electromagnetic lenses in particular are electromagnets which cause shrinking or spreading of the electron beam. The scanning is achieved with electromagnets that deflect the beam towards a specific point on the sample surface. For the proper functionality of this method, adequate "vacuum" must be accomplished with a constant air pumping after the introduction of the specimen to the microscope.

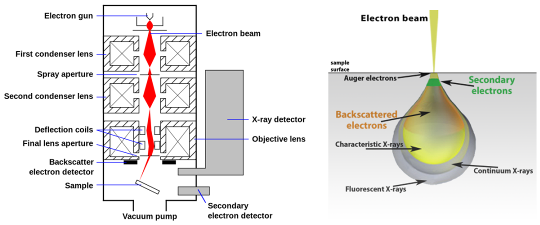


Figure 2.4: Scanning Electron Microscope schematic.
Source: NanoScience Instruments [9]

When the electron beam interacts with a sample in a scanning electron microscope (SEM), multiple events happen. In general, different detectors are needed to distinguish secondary electrons, backscattered electrons, or characteristic x-rays. Depending on the accelerating voltage and sample density, the signals come from different penetration depths (Fig: 2.4) [9]. Concerning sample topography, secondary electrons are only used. This type of electrons come from surface areas (after Auger electrons which come from the shallowest penetration areas on the sample). For this reason, out of the many different SEM detectors, the secondary electron detector, or SED, is used to produce a topographic SEM image.

However, SEMs do not *naturally* provide 3D images. 3D data can be obtained by integrating SEM images with different methods. One representing method is the 3D

SEM reconstruction from a stereo pair of images using the technique of photogrammetry[2]. The drawbacks are that it works only if there is a minimum texture, and it requires two images to be acquired from two different angles, which implies the use of a tilt stage. Actually, due to the fact that SEM stages are ill-equiped for tilt angle measurements, SEM topography has a vertical resolution of approximately $0.5 microns$ which is actually dependent on the lateral resolution and the tilt angle.

Regarding the advantages of the SEM with topographic orientation of use, along with the aforementioned vertical resolution, a lateral resolution of up to $10nm$ is achieved but it is very dependent on the specific microscope used. Additionally, there is small to none sample damage.

On the other hand, due to the difficulty of focusing the SEM beam on smooth surfaces or of exactly superposing the two images used for the photogrammetry technique, SEM topography is unsuitable for smooth surfaces. Furthermore, SEM needs conducting sample surfaces in order to permit the procedure of electron emission. To overcome this, one solution is to create a very thin layer of conducting material such as gold or chrome/chrome oxide on top of them with the method of sputtering.

2.2 Interferometry - *Michelson setup*

Interferometry utilizes the phenomenon of light interference. For the measurement of the surface profile of an unknown sample, a reference ideally flat object is needed (flat mirror). For each sample point, an elevation value is measured with respect to the corresponding point on the reference plane. To perform a measurement, an optical device is used which is built in order to produce the interference phenomenon. In this thesis, the Michelson setup is used. In this setup, plane waves are sent towards both the sample and the reference object. The reference object reflects plane waves as well, whereas the sample reflects according to its surface distribution. The two reflected waves are then superimposed and exhibit an interference pattern which incorporates information about the surface of the sample. Specifically, the surface elevation of the sample affects the OPD of the two individual waves before their superposition, which in turn changes the contrast and the density of the interference pattern. Hence, by analysing this measured pattern, the surface elevation can be calculated. The high accuracy of interferometry measurements is ought to the fact that very small changes in the OPD of the sample and reference produce measurable changes in the intensity of the interference pattern.

2.2.1 Interference

In Introduction Chapter (Sec: 1.2), it was stated that interference of two harmonic and not vertically between them polarized waves results in an intensity modulation that is actually the addition of the two individual intensities plus a sinusoidal modulation depended on the wavelength and their OPD as well. Starting from the wave definition, the interference equation can be obtained as follows [10, 11].

A polarized harmonic wave is defined as:

$$\boldsymbol{E}(\boldsymbol{r},t) = \boldsymbol{e}(\boldsymbol{r}) \cdot e^{i[g(\boldsymbol{r})-\omega t+\phi_0]} \tag{2.1}$$

[2] photogrammetry calculates absolute heights using triangulation methods

, where $\boldsymbol{e}(\boldsymbol{r})$: polarization vector,
ω : angular frequency,
$g(\boldsymbol{r}) = \boldsymbol{kr}$: phase due to wave propagation in space,
$\boldsymbol{k} = \mid k \mid \hat{k} = \frac{2\pi}{\lambda}\hat{k}$: wavevector,
$\mathbf{r} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}}$: position vector (starting from source point),
ϕ_0 : initial phase.

The superposition of two such waves is simply the sum of the two individual waves ($E_{tot} = E_1 + E_2$). The quantity of interest is the intensity of light of the superimposed waves. Starting from the definition of the intensity:

$$I \equiv \frac{c\epsilon_0}{2} E_{tot} E^*_{tot} \tag{2.2}$$

the total intensity of the superimposed waves can be calculated as:

$$I = I_1 + I_2 + c\epsilon_0 \Re\{\boldsymbol{e}_1(\boldsymbol{r}) \cdot \boldsymbol{e}^*_2(\boldsymbol{r}) e^{i[g_1(\boldsymbol{r}) - g_2(\boldsymbol{r}) + (\phi^1_0 - \phi^2_0)]}\} \tag{2.3}$$

, where the third term is called the interference term. By normalizing the polarization vectors, the interference term is expressed as follows:

$$\Re\{\boldsymbol{e}_1(\boldsymbol{r}) \cdot \boldsymbol{e}^*_2(\boldsymbol{r}) e^{i[g_1(\boldsymbol{r}) - g_2(\boldsymbol{r}) + (\phi^1_0 - \phi^2_0)]}\} = \frac{2}{c\epsilon_0}\sqrt{I_1 I_2}\, \Re\{\hat{\boldsymbol{e}}_1(\boldsymbol{r}) \cdot \hat{\boldsymbol{e}}^*_2(\boldsymbol{r}) e^{i[g_1(\boldsymbol{r}) - g_2(\boldsymbol{r}) + (\phi^1_0 - \phi^2_0)]}\} \tag{2.4}$$

Tidying up, the **interference equation** becomes:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cdot P(\boldsymbol{r}) \cdot cos[\Delta g(\boldsymbol{r}) + \phi(\boldsymbol{r}) + \Delta\phi_0] \tag{2.5}$$

, where $P(\boldsymbol{r}) = \mid \hat{\boldsymbol{e}}_1(\boldsymbol{r}) \cdot \hat{\boldsymbol{e}}^*_2(\boldsymbol{r}) \mid$,
$\Delta g(\boldsymbol{r}) = g_1(\boldsymbol{r}) - g_2(\boldsymbol{r})$,
$\phi(\boldsymbol{r}) = arg\Big[\hat{\boldsymbol{e}}_1(\boldsymbol{r}) \cdot \hat{\boldsymbol{e}}^*_2(\boldsymbol{r})\Big]$,
$\Delta\phi_0 = \phi^1_0 - \phi^2_0$

Coherence

So far, the above equation describes the rise of the interference phenomenon when two monocromatic waves superimpose [12]. The intensity modulation of the interference pattern depends on the notion of the coherence. Coherence is defined as the phase correlation between the wavefronts. It is divided in spatial and temporal coherence (Fig. 2.6) [13]. Spatial coherence is a measure of the correlation between the phase of a light wave at different points transverse to the direction of propagation. The wavefront formation in principle depends on the distance of the light source to the interference areas, since the farther the waves propagate, the spatial coherence is evolving. In addition to this, a significant factor that affects coherence is the extent of the source. Intuitively, the bigger the source the more individual emmited wavelets combine with one another, forming interference fringes of low contrast.

But in order for two wavefronts to interfere, it is not only adequate to be spatially coherent. The spectral bandwidth of the two sources affects as well the formation of the interference pattern. Using the principle of superposition, it can be assumed that the corresponding waves of the same wavelength interfere first and then as a set of superimposed waves, each wavelength gets superimposed with one another. The waves are considered temporally coherent over a time interval inversely proportional to the spectral width of the source(s) ($t_c = 1/\Delta\omega$). From this, it is evident, that the equivalent distance that light travels over the coherence time is the upper limit for the optical path difference of the two sources in order to exhibit interference. This distance is called coherence length.

Coherence Function

Effectively, the coherence of the two beams that travel through an interferometer (mirrors as sample and reference as well) can be expressed in terms of the correlation between them, or specifically the auto-correlation of the beam of the light source since the two splitted beams are ideally identical. Since coherence depends both on the physical dimensions of the light source and its spectral bandwidth, two different kinds of autocorrelation must be expressed.

Temporal Coherence

Firstly, the autocorrelation with respect to a time-delay τ expresses the coherence degree as the two beams get an increasing OPD along a single propagation axis ($\boldsymbol{r}_1, \boldsymbol{r}_2$ coaxial). In this way, the interferogram as a function of the delay τ can be written as:

$$\gamma(\tau) = \int E(t)E(t+\tau)dt \tag{2.6}$$

Taking the integral from $-\infty$ to $+\infty$, we define the Autocorrelation Function of the field to be:

$$\Gamma(\tau) = \int_{-\infty}^{+\infty} E(t)E(t+\tau)dt \tag{2.7}$$

From the autocorrelation and Fourier Transform relation, it holds that:

$$\mathcal{F}\{\Gamma(\tau)\} = F^*(\omega) \cdot F(\omega) = |F(\omega)|^2 \tag{2.8}$$

, where $\mathcal{F}\{\omega\} = \mathcal{F}\{E(t)\}$

This means that the Fourier Transform of the autocorrelation of a signal is the spectral power density of the signal.

In the current thesis, the temporal coherence length can be calculated by isolating the intensity variations (interferogram) of single point of a flat mirror while the latter moves across the z-axis. Then, the FWHM of the envelope of the absolute interferogram gives the coherence length for the specific white LED that was used. The measured coherence length is 2.2 μm (see Fig. 2.5).

Spatial Coherence

Secondly, the autocorrelation with respect to the separation $|\boldsymbol{r}_1 - \boldsymbol{r}_2|$ of two points in space expresses the coherence degree as the two beams get an increasing OPD while the angle between them increases and the difference of their individual distances from the source is constant ($|\boldsymbol{r}_1 - \boldsymbol{r}_2|$ increases, τ remains constant).

Following these, the Autocorrelation Function is now defined as:

$$\Gamma(\boldsymbol{r}_1,\ \boldsymbol{r}_2) = \int_{-\infty}^{+\infty} E(\boldsymbol{r}_1,\ t)E(\boldsymbol{r}_2,\ t)d\boldsymbol{r} \tag{2.9}$$

Combining, the above two expressions, a Mutual Coherence Function is defined:

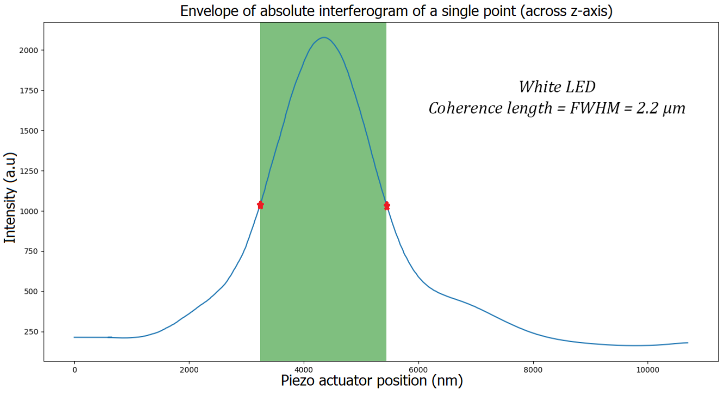


FIGURE 2.5: Temporal coherence length for the specific white LED that was used in the current thesis.

$$\Gamma(\boldsymbol{r}_1,\ \boldsymbol{r}_2, \tau) = \Gamma_{12}(\tau) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} E(\boldsymbol{r}_1,\ t)E(\boldsymbol{r}_2,\ t+\tau)\ d\boldsymbol{r}dt \tag{2.10}$$

The normalised Mutual Coherence Function with respect to the individual intensities is defined as the Mutual degree of Coherence:

$$\gamma_{12}(\tau) = \frac{\Gamma_{12}(\tau)}{\sqrt{I(\boldsymbol{r}_1)I(\boldsymbol{r}_2)}} \tag{2.11}$$

Since the degree of coherence is directly related to the visibility of the interference fringes, the equation of interference (Eq. 2.5) is generalised as:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cdot P(\boldsymbol{r}) \cdot \gamma_{12}(\tau) \cdot cos[\Delta g(\boldsymbol{r}) + \phi(\boldsymbol{r}) + \Delta\phi_0] \tag{2.12}$$

, where in this context τ is the temporal delay between the two wavefronts.

The following figure shows the relation between temporal and spatial coherence.

Broadband source analysis

The visibility of the interference fringes is affected by coherence. Firstly, one can imagine two waves emitted from different points (A and B respectively) with the same frequency are about to interfere at a point C. For the sake of simplicity, the initial phase difference and the angle of polarizations can be assumed as zero. The resulting intensity expressed in wavenumber terms is as follows:

$$\begin{gathered} I = A(k) + B(k) \cdot P(\boldsymbol{r}) \cdot cos[\boldsymbol{k}_1\boldsymbol{r}_1 - \boldsymbol{k}_2\boldsymbol{r}_2] = \\ I = A(k) + B(k) \cdot P(\boldsymbol{r}) \cdot cos[k(\hat{\boldsymbol{k}}_1\boldsymbol{r}_1 - \hat{\boldsymbol{k}}_2\boldsymbol{r}_2)] \end{gathered} \tag{2.13}$$

, where $\hat{\boldsymbol{k}}_i\boldsymbol{r}_i = k_{x,i}\ x_i + k_{y,i}\ y_i + k_{z,i}\ z_i$,
$A(k) = I_1(k) + I_2(k)$,
$B(k) = 2\ \sqrt{I_1(k)I_2(k)}$

Adding more pairs of waves with increasing frequencies will lead to an integration of Eq. 2.13 across the bandwidth of the sources. Hence, the interference pattern for the broadband sources will be:

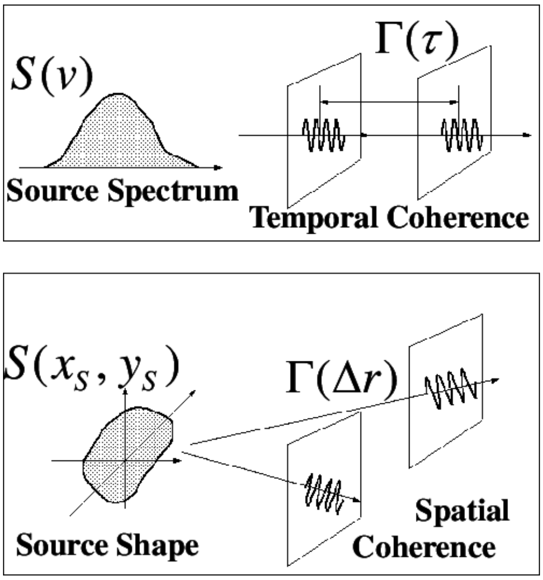


FIGURE 2.6: Relation between temporal and spatial coherence
Source: Scientific Figure on ResearchGate [13]

$$I = \int_{-\infty}^{+\infty} A(k)dk + P(\boldsymbol{r}) \cdot \int_{-\infty}^{+\infty} B(k) \cdot cos[k(\hat{\boldsymbol{k}}_1\boldsymbol{r}_1 - \hat{\boldsymbol{k}}_2\boldsymbol{r}_2)]dk \tag{2.14}$$

Expressing the cosine term in complex exponential form (omitting the conjucate part) and writing the spatial phase difference $\hat{\boldsymbol{k}}_1\boldsymbol{r}_1 - \hat{\boldsymbol{k}}_2\boldsymbol{r}_2$ as ρ:

$$I = \int_{-\infty}^{+\infty} A(k)dk + P(\boldsymbol{r}) \cdot \int_{-\infty}^{+\infty} B(k) \cdot e^{\iota\rho k}dk \tag{2.15}$$

In the above equation, the $B(k)$ term undergoes a Fourier Transform. In this way, the spectral distribution of the mutual intensity $B(k)$ arises in the total interference pattern. Under this concept, the more broadband the sources are, the smaller the area where the interference fringes are cleary visible.

The inverse relation of spectrum width and interference visibility is simulated and shown in Fig. 2.7. In each image, there are two plane wave fronts starting from the up left and bottom left corners respectively. The wavefronts propagate with an angle of 120 degrees with one another. The central wavelength of the two beams is 1000 nm and the image pixel size is 10 nm. Solid black area corresponds to absence of light, while solid gray areas correspond to constant intensity.

EXTENDED SOURCE

According to the van Cittert-Zernike Theorem [14], the spatial coherence area A_c is given by:

$$A_c = \frac{D^2\lambda^2}{\pi d^2} \tag{2.16}$$

, where d: the diameter of the light source and D: the distance away

This means that the smaller the light source, or the larger the distance away from the source, then the more spatially coherent the wavefronts are.

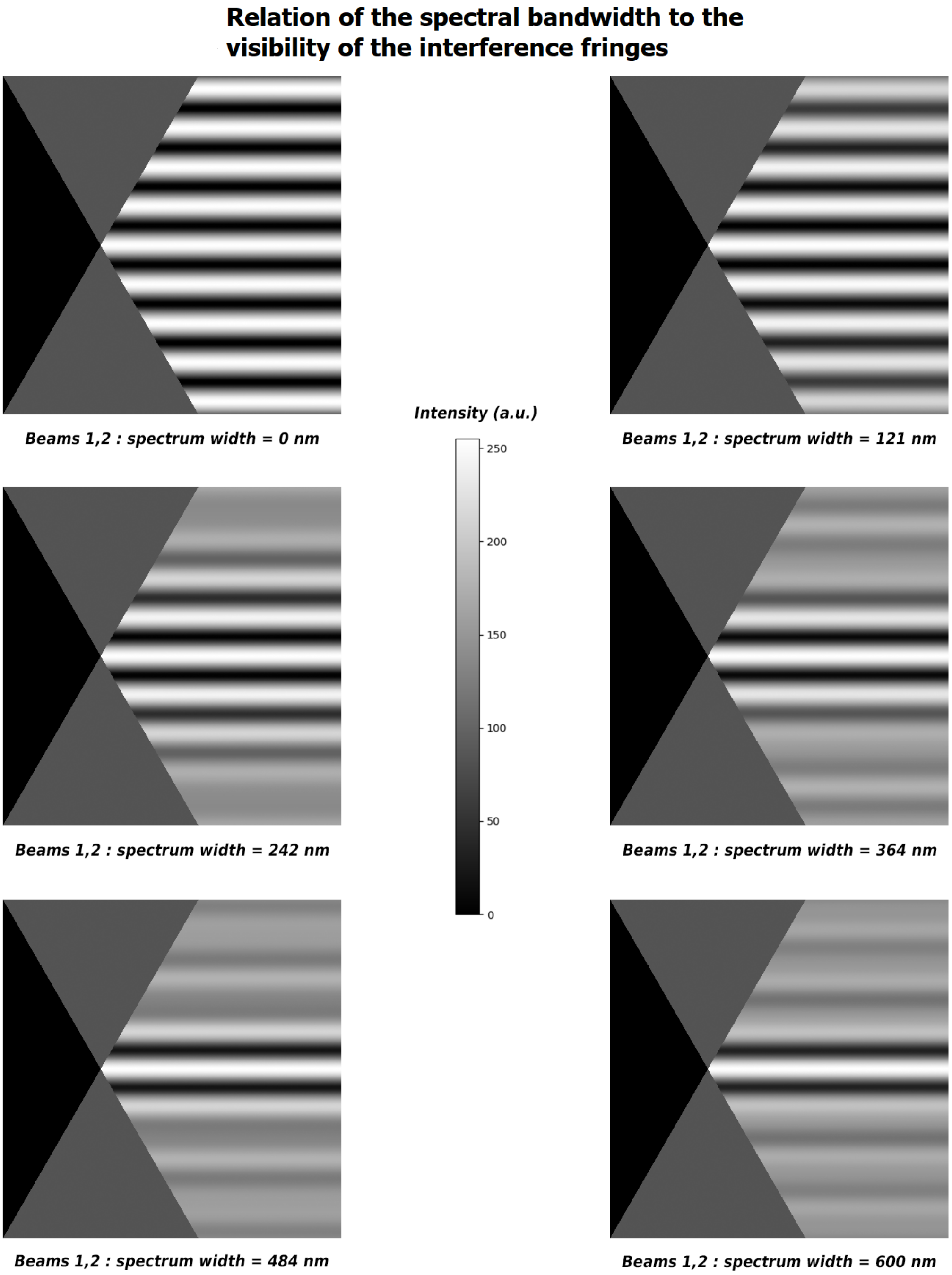

Relation of the spectral bandwidth to the visibility of the interference fringes
Beams 1,2 : spectrum width = 0 nm
Intensity (a.u.)
250
200
150
100
50
0
Beams 1,2 : spectrum width = 121 nm
Beams 1,2 : spectrum width = 242 nm
Beams 1,2 : spectrum width = 364 nm
Beams 1,2 : spectrum width = 484 nm
Beams 1,2 : spectrum width = 600 nm

FIGURE 2.7: In each image, there are two plane wave fronts starting from the up left and bottom left corners respectively. The wavefronts propagate with an angle of 120 degrees with one another. Solid black area corresponds to absence of light, while solid gray areas correspond to constant intensity.

2.2.2 Profilometry

One can imagine two waves of the same initial phase being reflected from a sample comprised only by two points at different elevations (compared to a reference ground). The highest point will cause the incoming wave to travel a smaller optical path from the point of reflection and on compared to the second wave. Therefore, this means that the two reflected waves are travelling in the same direction but with a phase difference that depends on the height difference between the two points of the sample. At this point, if the reflected wave is set to interfere with a reference wave, an interference signal will be produced, thus incorporating the OPD between the two waves. (Note: The offset of the OPDs of every sample point to the ideally flat reference plane does not play any role to the elevation measurement, since it is the relative OPDs among the sample points that build the elevation profile.)

Michelson setup

In order to guide two beams to illuminate the sample and the reference object respectively, then superimpose the two reflected beams so that they interfere, a Michelson optical configuration [15] can be used (Fig. 2.8). In this setup, a light source illuminates the sample and the reference using a beam-splitter. The two reflected beams are back reflected and recombine after the beamsplitter. and propagate to a converging lens is often used to image them on a recording medium such as a camera sensor where they interfere.

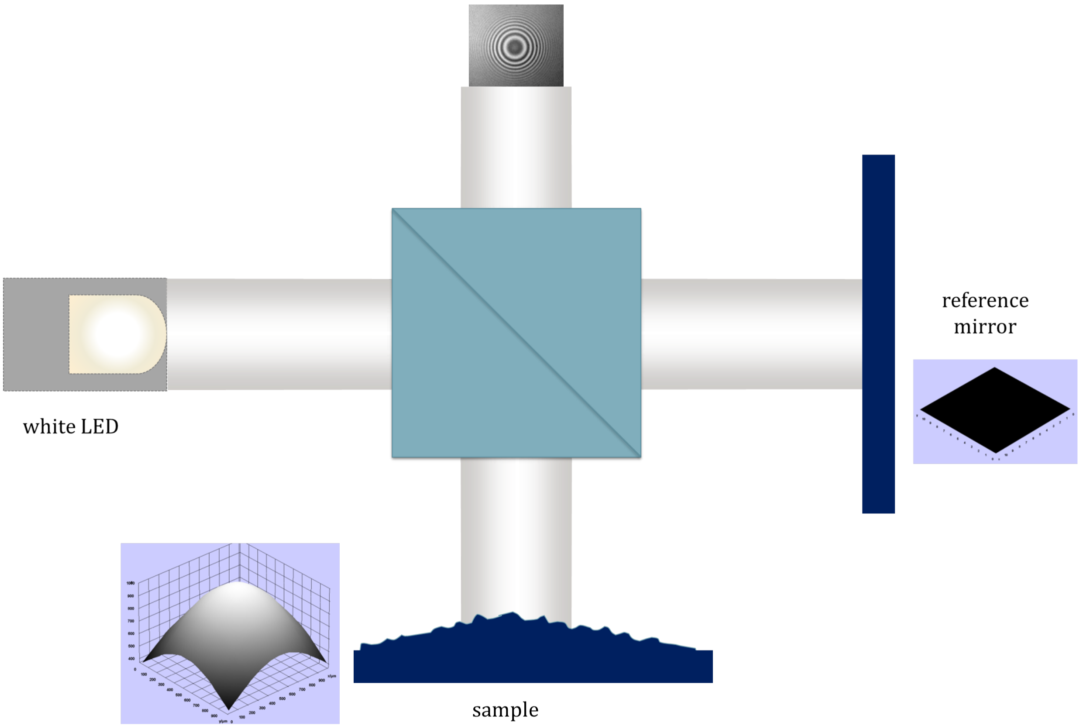


FIGURE 2.8: Basic arrangement of a Michelson interferometer. The initial beam is directed towards the sample and the reference using a beam-splitter. Then, the two reflected beams recombine in the beam-splitter where they interfere with one another.

The general interference equation (Eq. 2.12) can be simplifield to better descibe the operation of the interferometer. Assuming that the two interfering beams have initial phase difference equal to zero and the same polarization, the phase of the interference term is determined solely by the vector $\Delta g(\boldsymbol{r}) = \boldsymbol{k}_1\,\boldsymbol{r}_1 - \boldsymbol{k}_2\,\boldsymbol{r}_2$. It incorporates the direction and the current position of the two beams, giving thus information about their OPD. In a Michelson interferometer, the $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ vectors have equal amplitude since they refer to the same wavelength. Furthermore, $\boldsymbol{k}_i$ and $\boldsymbol{r}_i$ vectors are, under paraxial conditions, effectively collinear. Altogether, the phase vector can be simplified as follows:

$$\boldsymbol{k}_1\boldsymbol{r}_1 - \boldsymbol{k}_2\boldsymbol{r}_2 = k\,\hat{\boldsymbol{k}}_1\boldsymbol{r}_1 - k\,\hat{\boldsymbol{k}}_2\boldsymbol{r}_2 = k(r_1 - r_2) = k(z_1 - z_2) = k\;OPD \tag{2.17}$$

Substituting the above to the general interference equation (Eq. 2.12) results in:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2}\cdot P(\boldsymbol{r})\cdot\gamma_{12}(\tau)\cdot cos[k\;OPD] \tag{2.18}$$

The structure of an interferogram depends on the initial shape of the source wavefronts, the sample surface and the tilt between the sample and reference planes. The simplest case is two flat mirrors as "sample" and reference. Then, there is a combination of different interference patterns depending on if the illuminating wave fronts are plane or spherical and if the two mirrors are tilted with one another. For the ease of visualization, the two mirrors are shown together on a single plane (Fig. 2.9). The distances between the mirrors are the OPDs between corresponding points. Then, the illuminating wave fronts are shown as they propagate towards each mirror. Finally, the reflected wavefronts are shown. Their phase difference vector $\Delta g(\boldsymbol{r})$ which is also the phase of the interference term of the intensity modulation) underline the shape and direction of the interference pattern. More specifically, the fringes vary over an axis that is normal to the $\boldsymbol{k}_1 - \boldsymbol{k}_2$ vector and as it can be shown, the period of the intensity variation of the fringes (Λ) depends on the wavelength of the light source (λ_0) and the angle between the interfering wavefronts (θ, also tilt between mirrors) (Eq. 2.19).

$$\Lambda = \frac{\lambda_0}{2\,sin(\theta/2)} \tag{2.19}$$

- (Fig. 2.9(A))Illumination: Plane waves. Tilt between mirrors: $\theta = 0$. Then, $\Lambda = 0$ and the result is a constant intensity across the observation plane which depends on the OPD of the mirrors. That is, as the OPD of the mirrors change linearly, the observed intensity oscillates sinusoidally between black and white.

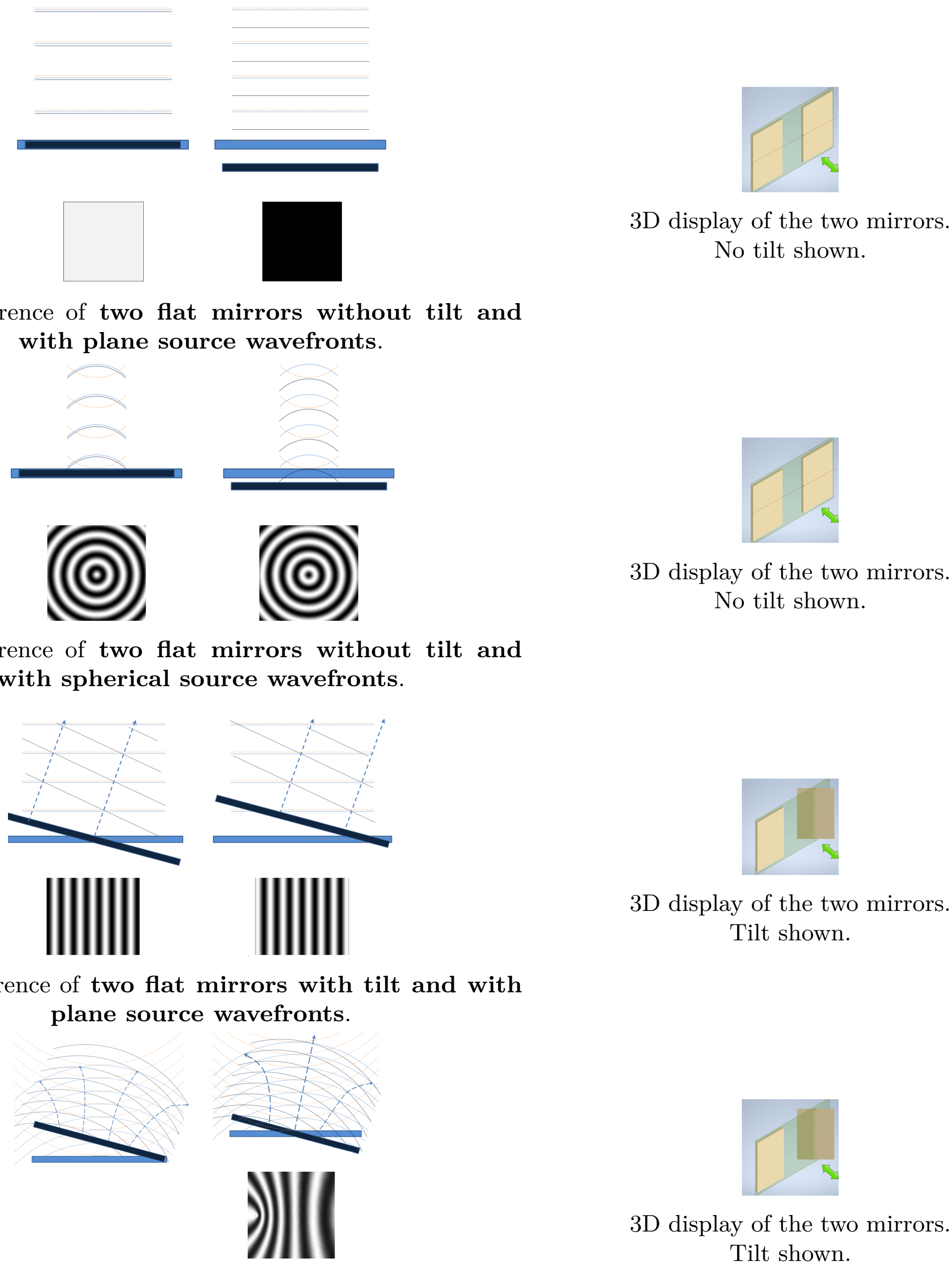
- (Fig. 2.9(B) Illumination: Spherical waves. Tilt between mirrors: $\theta = 0$. Then, for every point of interference, the angle between the component spherical waves is constant and consequently the result is equally spaced circular fringes. A linear increase of OPD between mirrors leads to linear increase of $\boldsymbol{r}_1$ and/or $\boldsymbol{r}_2$ equally across the observation plane. Therefore, the phase of the intensity modulation shifts sinusoidally and so the intensity for every point in the observation plane oscillates sinusoidally too.

- (Fig. 2.9(C) Illumination: Plane waves. Tilt between mirrors: $\theta > 0$. Then, for every point of interference, the angle between the component plane waves is constant and consequently the result is equally spaced linear fringes (normal to the $\boldsymbol{k}_1 - \boldsymbol{k}_2$ vector). A linear increase of OPD between mirrors leads to a lateral shift of the point where OPD is zero and to a linear increase of $\boldsymbol{r}_1$ and/or $\boldsymbol{r}_2$.

Therefore, the phase of the intensity modulation shifts sinusoidally and so the fringes appear to shift laterally in the observation plane.

- (Fig. 2.9(D) Illumination: Spherical waves. Tilt between mirrors: $\theta > 0$. Then, component spherical waves interfere with different angles across the observation plane and consequently the result is curved fringes but not equally spaced. As OPD changes, the intensity for every point in the observation plane oscillates sinusoidally.

Experimentally captured interferograms using a white LED as a source are shown in Fig. 2.10. Various samples from measurements of the current thesis are shown (spherical lens, mirror, polymer film - from top left to bottom). The images are in grayscale since the camera used is a monochrome one.



3D display of the two mirrors. No tilt shown.

(A) Interference of **two flat mirrors without tilt and with plane source wavefronts**.

3D display of the two mirrors. No tilt shown.

(B) Interference of **two flat mirrors without tilt and with spherical source wavefronts**.

3D display of the two mirrors. Tilt shown.

(C) Interference of **two flat mirrors with tilt and with plane source wavefronts**.

3D display of the two mirrors. Tilt shown.

(D) Interference of **two flat mirrors with tilt and with spherical source wavefronts**.

FIGURE 2.9: Representation of the action of mirrors in a Michelson interferometer. For visualization purposes the sample mirror is depicted as overlayed on the reference mirror. Illuminating wavefronts (orange dashed curves) propagate to and reflect from the mirrors (blue solid curves). Dashed arrows show points of constructive interference across the vertical axis. The resulted interference patterns shown under the drawings are the tranverse cross sections (left-right and over-under directions with respect to the paper)

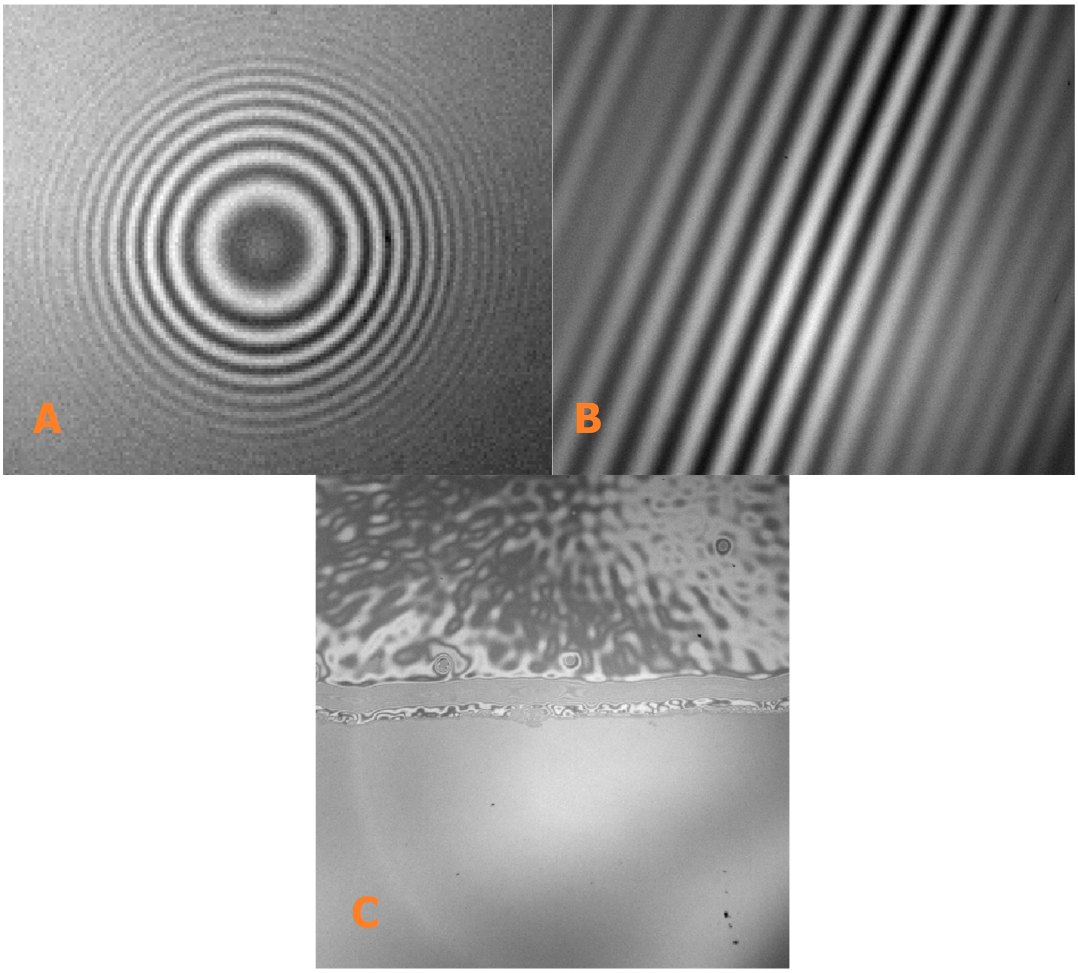


FIGURE 2.10: Various samples from measurements of the current thesis are shown (spherical lens (A), mirror (B), polymer film (C). The images are in grayscale since the camera used is a monochrome one.

Coherence length and profilometry applications

The source of the aforementioned waves can be either temporally coherent or not. The choice of the degree of temporal coherence depends on the specimen under study and specifically on whether the surface contains abrupt changes. An abrupt change is considered a step in the elevation more than half the wavelength of the light source ($\lambda/2$). As it will be explained below, with a low-coherence light source one can measure samples with steps more abrupt than $\lambda/2$, whereas with a high-coherence light source only smoother surfaces can be accurately measured.

As previously stated, when the OPD between the interferometer arms changes, the interference fringes either shift laterally or their intensity changes sinusoidally. For the sake of an example, let say there is a sample with a vertical step. Then, since the step in the elevation imposes also a step in the OPD of the sample arm, the interferogram will exhibit two interference patterns with a phase shift corresponding to the OPD. As the OPD increases, the phase shift increases too. Therefore, by measuring the phase shift of the interferogram, the OPD and consequently the step elevation (=OPD/2) can be measured indirectly.

When using a fully coherent light source, the extrema of the fringes always have the same intensity (a perfect cosine function). So, every phase shift $\Delta\phi$ gives the exact same fringe visibility as $\Delta\phi + 2m\pi, m = 1, 2, 3, \ldots$. For example a phase shift of π has the same interferogram with phase shifts of 3π, 5π, ... This results to an ambiguity for the deduction of the sample elevation, thus making a fully coherent source only applicable for measuring samples that are smooth without abrupt height changes larger that $\lambda/2$.

On the other hand, incoherent sources or partially coherent sources like white light lead to fringes with decreasing visibility as the OPD increases. This makes it possible to identify the zero order phase shift and extract the absolute phase diference without the 2π ambiguity.

The following subsections present the basic principles of measurement and profile analysis of different categories of interferometry with respect to the coherence degree of the light source and the specimen under study. In all of these categories, the implemented interferometer might be of any type, such as Michelson, Twyman-Green, Mirau etc.

2.2.3 Phase Shifting Interferometry

In this technique one typically uses a high coherence illumination source such as a laser. The concept behind phase-shifting interferometry (PSI) [16] is to apply a time-varying phase shift between the reference and sample wavefront. This can be achieved, for example, by mounting the reference optic on a linear transducer, such as a piezoelectric crystal (see 2.3). Assuming that the relative phase-shift ϕ changes linearly with time, the intensity at every point in the photodetector array, I, will change sinusoidally with ϕ(see Eq. 2.18) where the bias level, $I_1 + I_2$, and modulation amplitude, $2\sqrt{I_1 I_2} \cdot P(\boldsymbol{r}) \cdot \gamma_{12}(\tau)$, are unknown and the optical phase difference between the sample and reference, OPD, is the quantity to be measured. The OPD is related to the surface elevation, h, by $OPD = 4\pi h/\lambda$ where λ is the laser wavelength. Since there are three unknowns terms, measurement of at least three interferograms at known phase differences is needed to determine OPD. For example, by taking data at $\phi = 0, \pi/2, \pi, 3\pi/2$, one can compute the phase difference at every point by the simple formula:

$$OPD(x,y) = tan^{-1}\left[\frac{(I_4(x,y) - I_2(x,y))}{I_1(x,y) - I_3(x,y)}\right] \tag{2.20}$$

, where $I_1 \cdots I_4$ are the intensities measured at every photodetector element at phase shifts $\phi = 0 \cdots 3\pi/2$. Since the calculation is performed at every point, variations in intensity and photodetector sensitivity are not an issue.

Since the retrieved phase is bounded in the region $[0, 2\pi]$ the resulted heights of the sample will be wrapped in this region. Hence, this can be corrected by unwrapping the phase in $[0, +\infty]$: Wrapped phase = Modulo 2π (Unwrapped phase).

2.2.4 Low Coherence Interferometry

When the specimen under study is known to have abrupt changes in its surface elevation, Low Coherence Interferometry (LCI) is chosen. LCI is actually a family of many methods that use spectrally broadband light. The most common one is the White Light Interferometry (WLI) that uses visible broadband light. The unknown surface elevation is contained in the OPD term in the interference equation (Eq. 2.18). Finding the OPD can be done in more than one ways.

Coherence Scanning Interferometry

This method is used in the current thesis. Coherence Scanning Interferometry (CSI) [17, 18] produces a 2D+1 surface elevation map of the sample under study. It evaluates changes in interference signal visibility related to optical coherence. In the simplest conceptualisation, surface heights are inferred by noting where the interference fringe visibility is highest. Thus one defining feature of CSI instruments, with respect to PSI instruments, is that by design, interference fringes are only strongly observed over a narrow surface height range defined by the temporal coherence length of the source..

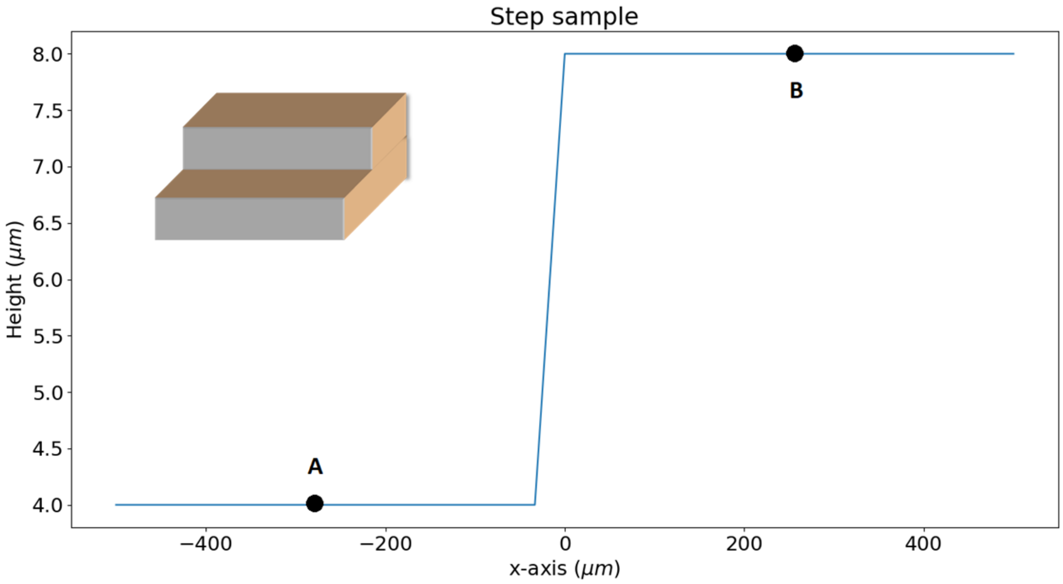


FIGURE 2.11: Sample with a single abrupt step

The interference signal is recorded while continuously changing the OPL of the reference arm (vertical scanning). This mechanical motion is accomplished with a high precision piezo actuator (see 2.3). In that way, a 3D cube (image stack) is formed, with width and depth the dimensions of the field of view and height the number of scans. Via the calibration of the piezo actuator, each scan movement of the reference plane is corresponded to a real displacement. Assuming that the reference plane defines the equal to zero refererence height, then when it is shifted by a distance equal to the height of a point of the sample, the OPD of reference and sample point is equal to zero. It is known that in zero OPD the intensity of the interferogram maximizes. Therefore, this results to a straightforward algorithm that relates the position of the interferogram maximum with the unknown height of the sample (see Fig. 2.11, 2.12). By doing this for every sample point (every pixel), then

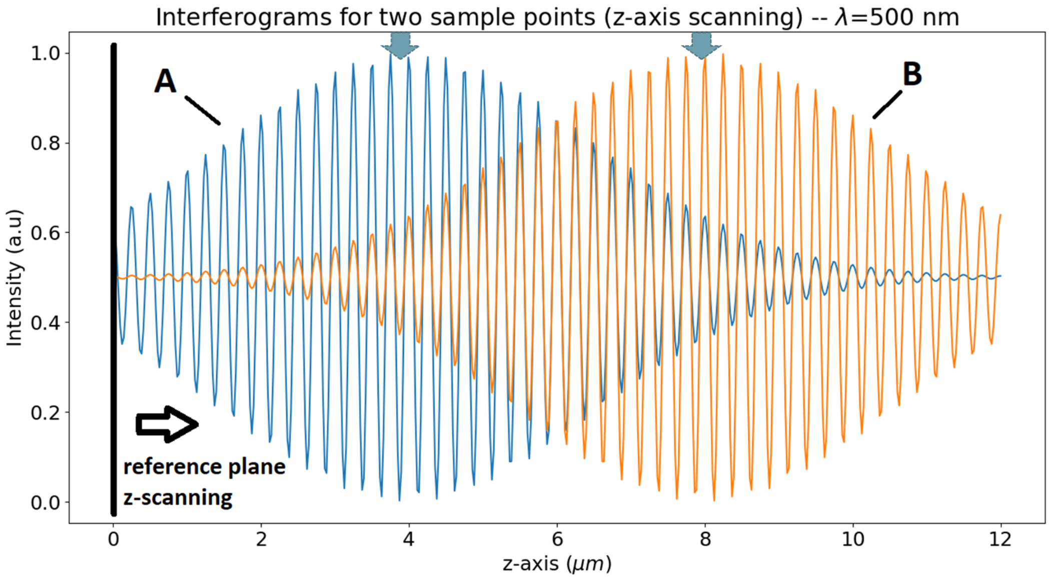


FIGURE 2.12: Corresponding interferograms of point A and B. Contrast of fringes maximizes when OPD of each point with the reference plane is equal to zero. By finding the OPL that the reference plane must travel in order to reach a zero OPD with a sample point, the elevation of the sample point can be deduced. It is half the traveld OPL.

the surface height profile of the entire sample is deduced, forming in this way a surface topographic map. The idendification of the interferogram maximum can be done in several ways like for example with peak fringe intensity detection, demodulation, centroid detection, FFT, wavelets and signal correlation. The method of peak fringe intensity detection that is implemented in the current thesis is presented in detail in Chapter 3, Sec. 3.2.

2.3 Automation and mechanical parts

The required repeated and discrete change of the OPL of the interferometer reference arm of the interferometer is controlled automatically by an electro-mechanical system (Fig:A.1).

Piezoelectric actuator

Specifically, the system consists of a high-precision piezo transducer or actuator for linear micromovement and its driving electronics. The functionality of such an actuator lies in the nature of piezo materials. These materials can convert electrical energy to mechanical energy, thus justifying their alternative designation as motors [19]. They are a special type of ceramics that expand or contract when an electrical charge is applied, generating motion and force. Conversely, piezo materials will also generate voltage when a mechanical stress is applied. When a linear micromovement is required, linear actuators are used.

The most common type is the *preloaded stacked actuators*, which consist of multiple layers of piezo elements on top of each other. This combines the effect of each element's expansion to produce a useful movement and force. The individual piezo elements have alternating polarity (Fig:2.13) [20], and the electrical field is applied parallel to the direction of polarization. The applied voltage for commonly used stacked piezos can vary in the range of 0-200 V (for low-voltage applications) and up to 1000 V (for high-voltage ones). In the current thesis, a low voltage piezo is used, specifically in the range of 0-100 V. It is important to note that in steady state operation (no movement, constant force), no current is flowing and no power is consumed.

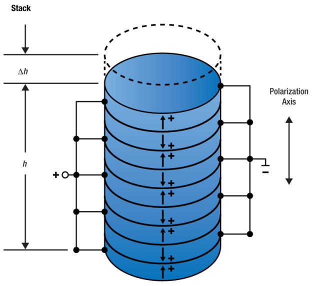


Figure 2.13: When a voltage is applied in the direction of polarization, displacement is produced. In a stacked piezo actuator, the total displacement is the sum of each layer's individual displacement.
Source: Thorlabs [20]

Concerning the total displacement (ΔL) generated by the piezo, it is generally between 0.1 and 0.15 percent of the actuator length. It can be estimated by the product of the applied voltage (V) to the number of the piezo layers (n) multiplied by the longitudinal piezo-electric coefficient (d):

$$\Delta L = n \cdot V \cdot d \tag{2.21}$$

The size of the actuators is typically small, in the range of a few mm. Nevertheless, they attain a force density in the range of $30N/mm^2$. In order to reach the full tensile force [3] when expanding, the piezo stack is compressed (preloaded) with a form of spring with a value greater than the applied tensile load. The compression is done because the actual tensile force is largely determined by the effectiveness of the lamination process of the piezo layers. Also, in order to prevent bending, shear or torsional forces, piezo stacks are mounted in a rigid cell.

[3] force needed to pull something apart

Displacement with and without feedback

For various reasons like mechanical drift, hysteresis and voltage fluctuations, piezo actuators do not reach their expected displacement *among different sessions.* However, these displacement variations can be significantly eradicated when a feedback loop subsystem is introduced incorporating a sensor that provides position data used for microcorrections. This is called a closed-loop piezo actuator in contrast with an open-loop one. Closed-loop provides higher precision, linearity and repeatibility (Fig:2.14) [21] at the expense of speed, complexity and cost. Several types of position sensors are used for feedback, ranging from piezoresistive strain gauges to capacitive sensors and incremental linear scales. A controller is feeded both by reference displacement value and actual value from the sensor and then sends a correction signal to an amplifier which in turn drives the piezo. Even though displacements in open-loop actuators are not known beforehand, they can be measured with a process of *calibration* through the collected interferogram images. Therefore, a mapping between number of piezo step or input voltage and actual displacement is created.

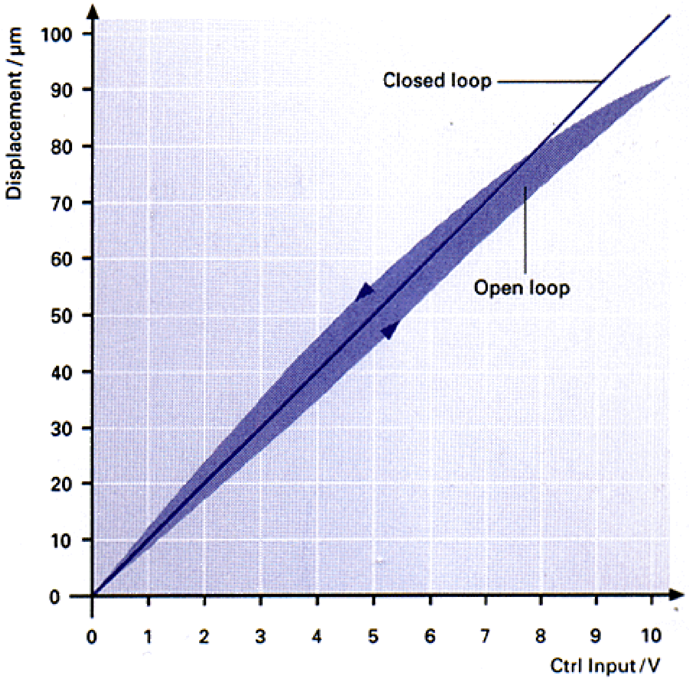


FIGURE 2.14: Open loop vs. closed loop performance of a typical PI piezo actuator.
Source: Physik Instrumente [21]

Chapter 3

Implementation

3.1 Optical setup and automation

Fig. 2.8 shows the layout of a Michelson interferometer. A camera placed in the interferometer exit will capture the interference fringes provided that the OPD of the interferometer arms is less than the coherence length of the light source. For incoherent white light illumination, where the coherence length is in the order of a few microns, this is proven to be tremendously tricky. The solution is to firstly image the surface of the reference mirror. By imaging the mirror, its distance from the camera is fixed. Once this is done, we focus the sample by moving the sample arm. Once the sample is sharply focused, the sample-reference OPD is close to zero. Only needs micro-adjustments in the position and tilt of the sample are then needed in order for the white light interference fringes to appear.

The imaging of the sample is performed using a telecentric system comprised by two lenses. Such a system receives parallel rays and produces also parallel rays in a way that every point of the sample is imaged with the same magnification, preventing thus the image to be distorted. In the current thesis, the lens closer to the beamsplitter is a photographic lens with focal length equal to 50 mm, while the lens closer to the camera is also a photographic lens but with focal length equal to 25 mm. This results in a magnification equal to $25/50 = 1/2$ that shrinks the image. Fig. 3.1 shows the addition of the imaging system.

The implementation of the optical setup is shown in Fig. 3.2(a). The basic skeleton of the setup is developed by Ioannis Orfanos for his master thesis [18]. The whole setup is a portable device than can be put in any flat table without needing any optical table or special vibration isolation systems. The chassis of the setup, interconnecting the interferometer arms and the camera-beamsplitter path, is placed onto a table on small sorbothane hemispherical dampeners. The mechanical development in this thesis concerns the expansion of the interferometer arm. In particular, the sample is mounted on a series of micro-positioning and tilt mechanisms attached in such a way that the sample can move and rotate in all 3D axes (x-y-z, with z-axis the axis vertical to the table).

Fig. 3.2(b) shows the sample holder (used to mount a spherical sample) attached to a tilt mechanism which in turn is attached to a z-axis positioning mechanism. Altogether these parts are mounted on the x-y-axes positioning mechanism which in turn is mounted on the device chassis. Fig. 3.2(c) shows the piezo actuator with the mounted reference mirror. The mirror is glued to a screw with a special epoxy glue which in turn is screwed in the piezo actuator external mount. Fig. 3.2(d) shows the box containing the automization electronics.

Fig. 3.1 also shows the automation system which concerns the synchronization between the interferogram capturing and the reference mirror displacement. Repeated and controlled changes of the OPL of the reference arm, needed for the interferogram

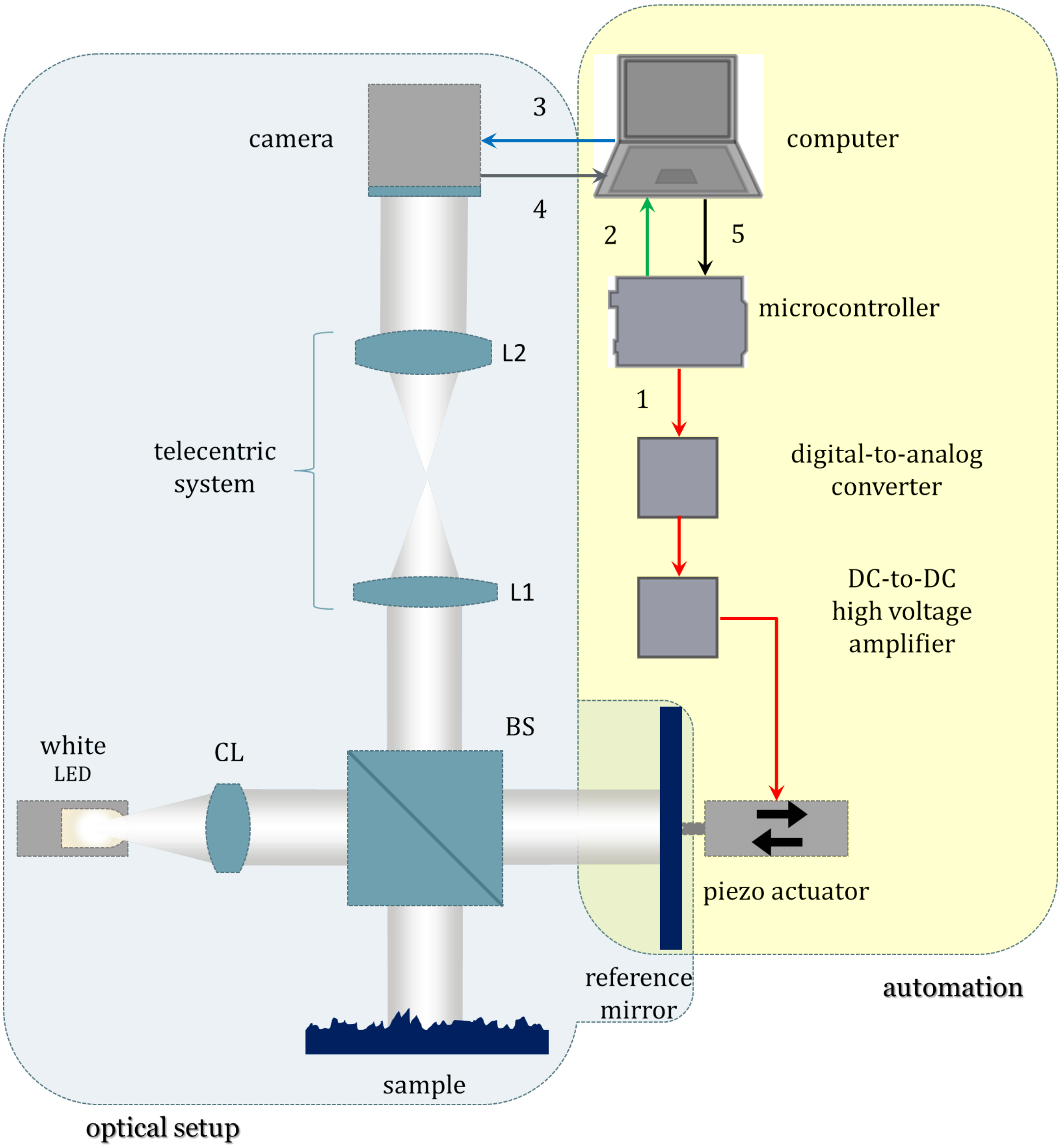


FIGURE 3.1: Michelson setup with the addition of the imaging system.

analysis, are realised by the piezoelectric actuator. Of course, in each step of the piezo, an image must be captured. Hence, it is easily understandable that a *turn-based cooperative "game"* must be played between the piezo and the camera. The piezo carries out a displacement and then the camera captures an image of the interferogram. Then a second displacement is made and a second image is captured. This procedure is followed until the total number of needed interferograms is achieved. In order to achieve synchronization, two seperate flows of signals happen alternately in each moment. The first flow travels from the *microcontroller (Arduino) towards the piezo* while the second one runs from the *computer to the camera and back*. Of course for these flows, in order to happen alternately, the computer and the Arduino must communicate with one another. For this purpose, the two devices communicate serially via the USB cable that connects them.

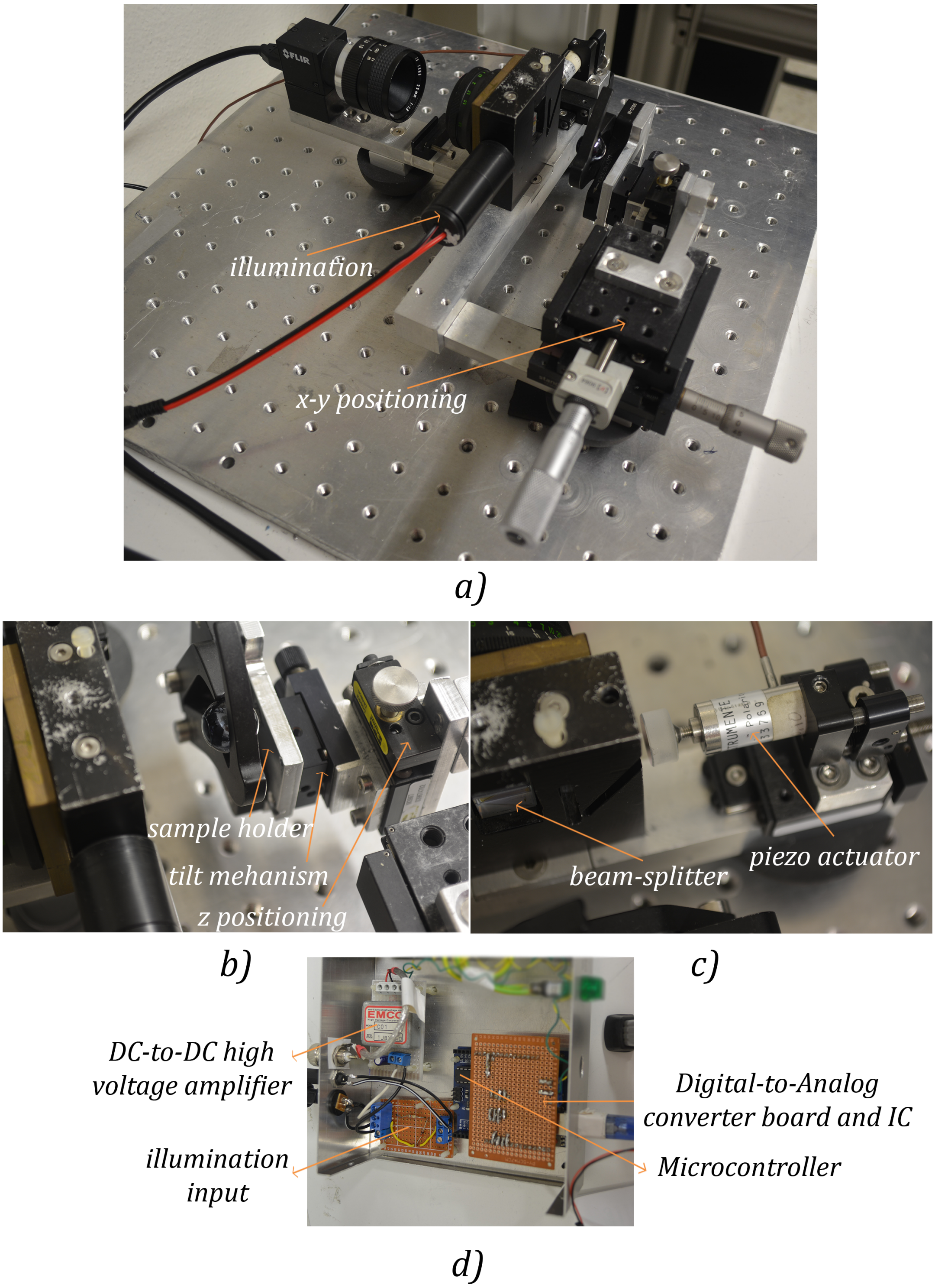


Figure 3.2: The implementation of the optical setup. Image (b) shows the sample holder with the tilt and z-axis adjustment mechanisms. shows the piezo actuator with the mounted reference mirror. Image (d) shows the box containing the automization electronics.

3.2 Profile analysis

PER PIXEL ANALYSIS

In this thesis, the algorithm of peak fringe visibility detection [22] is used with a few variations. In order to find the interferogram maximum visibility $V_{max}(x, y, z)$ for each position (x, y) on the sample surface, it is not safe to get the absolute maximum intensity $I_{max}(x, y, z)$ of the interferogram because this is sensitive to noise related errors. To mitigate such errors, the envelope of the interferogram signal is first calculated. Then by executing a mathematical fit of the envelope with a known function (e.g gaussian, or more precisely the $\gamma(\tau)$ functional distribution of the source), the position z' (or OPD) where the fringe visibility is maximized can be estimated. Below, steps towards the maximum visibility estimation are presented (see Fig. 3.3):

1. Extract the interferogram which corresponds to a single pixel. (A vertical array gets extracted from the 3D cube).

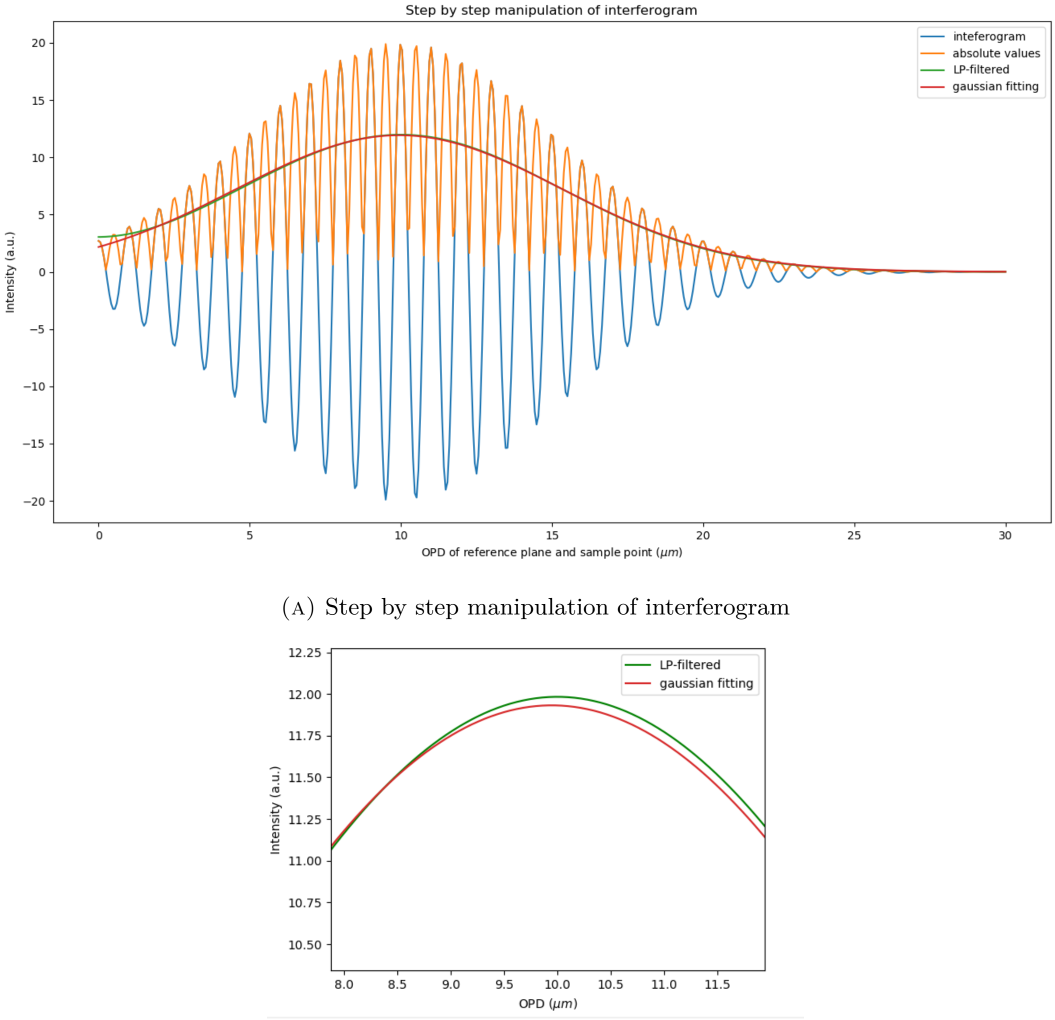
2. Remove DC bias from the interferogram signal. (Subtract the median of the interferogram from the latter.)

3. Get the absolute value of the signal.

4. Apply a low-pass filter (e.g gaussian) with such parameter(s) so that a the envelope of the signal is retrieved

5. Find a function (e.g gaussian) that best-fits with the envelope-like signal. For the gaussian case $Ae^{\frac{z-z_0}{w}} + B$, the function parameters are the offset z_0, width w, amplitude A and dc bias B. To help the best-fitting calculation get faster, a subset of these parameters can be provided as a first guess. A guess for z' can be the position for which the signal is at the maximum (argmax). Since, the signal has the zero value as a low boundary, the guess of B can be zero and A can be the maximum value of the signal. Regarding w, this depends on the coherence of the light source. Hence, it is constant for every interferogram and can be easily calculated.

6. The best-fit calculation yields all the optimized parameters of the gaussian function. The wanted parameter is the optimized z' which corresponds to the wanted elevation of the sample point. Since all the calculations are performed without taking into account any real unit, z' is with 'sample' units (acting as an index). Therefore, using the calibrated reference plane displacements, z' is associated to a real displacement and equivalently to a real elevation value. This is done via mathematical interpolation of the calibrated values and the calculated index.



(A) Step by step manipulation of interferogram

(B) Gaussian fit and LP-filtered signals

Figure 3.3

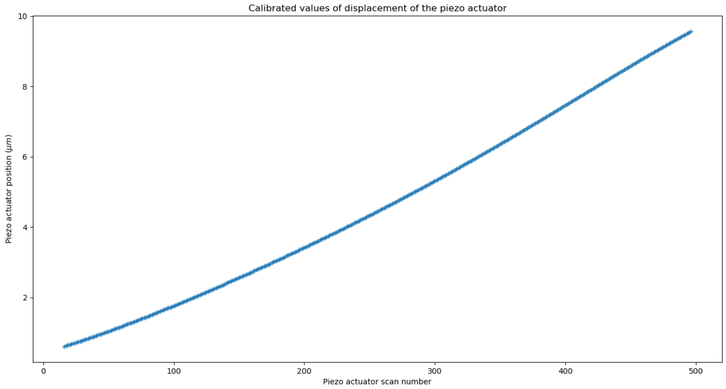


Figure 3.4: Association of piezo actuator scan movements to real displacements. Using the optimized mean value of the best-fit calculation as an argument, it is associated to a real displacement and equivalently to a real elevation value.

3.3 Post-processing profile analysis

It is quite probable that the profile analysis will not give immediately a useful surface map of the sample, since the profile might contain irregular single pixel bursts. Noise related issues and/or imperfections in the sample render specific points not capable of being measured by the used system. This degrades the optically comprehensible result and surface quality estimations such as surface roughness. For these reasons, a software post-processing tool has been made and is used as shown in Fig. 3.13.

The post-processing steps towards the cleanup of the profile are the following:

- At first, any NaN (not a number) values of the profile (arising from a failed z-axis interferogram analysis) are substituted by the result of the cubic interpolation of the neighbor values. Then an appropriate lower and upper threshold for the profile values is selected by the user by using both histogram and visual feedback.
- A region of interest is selected and cropped.
- For flat surfaces, the profile is leveled. That is, the whole profile is rotated in such a way that the in principle flat area will eventually have a slope equal to zero.
- A region of interest fis selected and surface quality estimations are extracted.

3.4 Surface texture parameters

The areal height parameters used are the following.

- M: Mean plane. The arithmetic average of a set of values. It is calculated by summing the data and dividing by the number of points.

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- Sp: Maximum peak height of the areal surface.

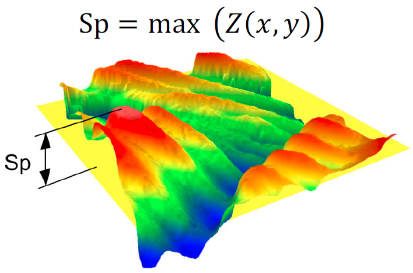


FIGURE 3.5: Maximum peak height of the areal surface
Source: Zygo [23]

- *Sv*: Maximum valley depth of the areal surface.

$$\mathrm{Sv} = \min\big(Z(x,y)\big)$$

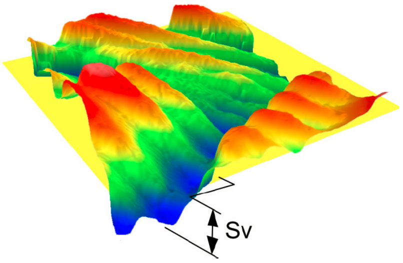


FIGURE 3.6: Maximum valley depth of the areal surface
Source: Zygo [23]

- *Sz*: Maximum height of the areal surface. It is the peak to valley height. (Also used as a flatness estimation for a flat mirror sample, given that the mirror profile is leveled.)

$$\mathrm{Sz} = \mathrm{Sp} + \mathrm{Sv}$$

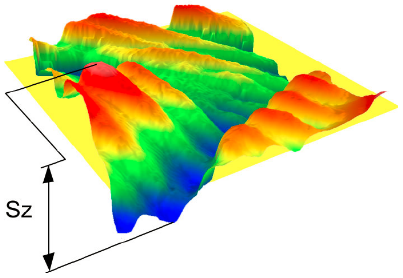


FIGURE 3.7: Maximum height of the areal surface
Source: Zygo [23]

- *Sa*: Roughness: Mean difference around mean plane.

$$\mathrm{Sa} = \frac{1}{A}\iint_{A} |Z\,(x,y)|\, dxdy$$

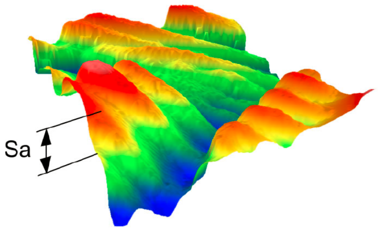


FIGURE 3.8: Roughness: Mean difference around mean plane
Source: Zygo [23]

- *Sr*: Roughness: Standard deviation around surface waviness. Waviness: smoothed surface in such a way that the low frequency oscillations dominate.

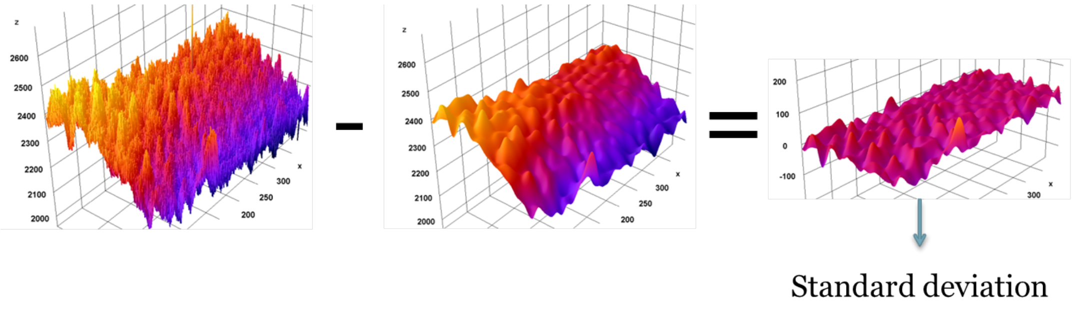
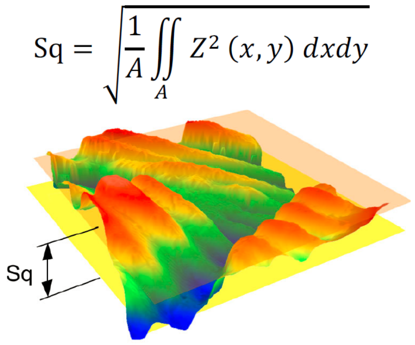


FIGURE 3.9: Roughness: Standard deviation around surface waviness

- *Sq*: Roughness: Root mean square difference around mean plane.



$$Sq = \sqrt{\frac{1}{A}\iint_{A} Z^{2}\,(x,y)\,dxdy}$$

FIGURE 3.10: Roughness: Root mean square difference around mean plane
Source: Zygo [23]

- *Ssk*: Skewness: Degree of symmetry of the surface heights around the mean plane. Ssk>0: peaks predominance, Ssk<0: valleys predominance.

- *Sku*: Kurtosis: Presence (Sku>3) of lack (Sku<3) of inordinately high peaks or deep valleys.

$$\mathrm{Ssk} = \frac{1}{S_q^3}\left[\frac{1}{A}\iint_A \left(Z(x,y)^3\right) dxdy\right]$$

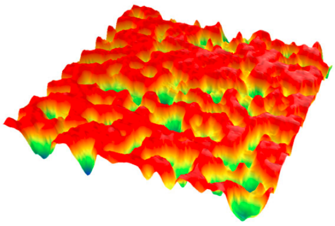


FIGURE 3.11: Skewness: Degree of symmetry of the surface heights around the mean plane
Source: Zygo [23]

$$\mathrm{Sku} = \frac{1}{S_q^4}\left[\frac{1}{A}\iint_A \left(Z(x,y)^4\right) dxdy\right]$$

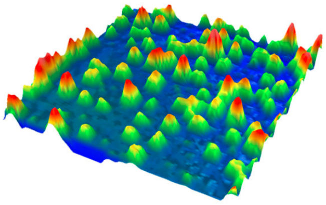


FIGURE 3.12: Kurtosis: Presence of lack of inordinately high peaks or deep valleys
Source: Zygo [23]

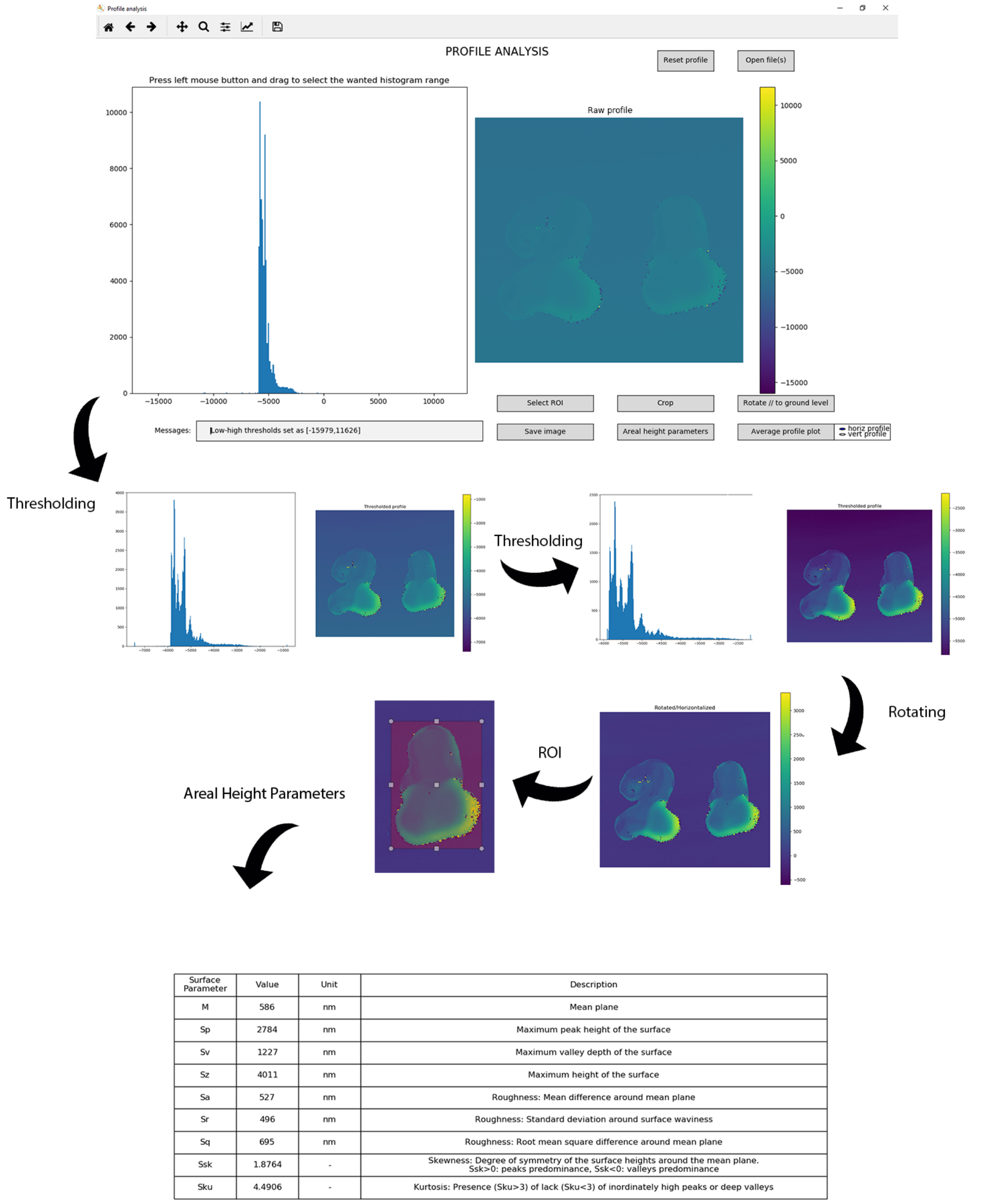
3.5 Assessment of samples

CURVED SAMPLES

In order to estimate the RoC of a spherical surface, the measured surface profile is fitted with a 2D paraboloid ($Z = \frac{(X-x_0)^2+(Y-y_0)^2}{2\,R} + z_0$). Additionally, the best-fit algorithm estimates the standard deviation of the error in the RoC measurement taking also into account the accuracy in the estimation of the transverse scale . The aforementioned steps are shown briefly in Fig. 3.14.

STEP SAMPLES

For the assessment of samples with abrupt changes like a butte or a dent a software analysis tool has been developed (see Fig. 3.15). For the situation of a step sample, this tool takes the average of the step profile across the horizontal or the vertical dimension. Then, it automatically detects the different levels of elevation and estimates the standard deviation of the step height. Sample points that exist in a region ±20% left and right of the step position do not participate in the standard deviation calculation. The standard deviation of the step is the sum of the two indidual standard deviations of the upper and lower levels. Moreover, the user can also subjectively make small adjustments to the elevation and location of the step.



Messages: Low-high thresholds set as [-15979,11626]

Surface Parameter	Value	Unit	Description
M	586	nm	Mean plane
Sp	2784	nm	Maximum peak height of the surface
Sv	1227	nm	Maximum valley depth of the surface
Sz	4011	nm	Maximum height of the surface
Sa	527	nm	Roughness: Mean difference around mean plane
Sr	496	nm	Roughness: Standard deviation around surface waviness
Sq	695	nm	Roughness: Root mean square difference around mean plane
Ssk	1.8764	-	Skewness: Degree of symmetry of the surface heights around the mean plane. Ssk>0: peaks predominance, Ssk<0: valleys predominance
Sku	4.4906	-	Kurtosis: Presence (Sku>3) of lack (Sku<3) of inordinately high peaks or deep valleys

FIGURE 3.13: By applying a lower and upper bound to the histogram of the profile, unwanted pixels are discarded. This might be an iterative procedure up to a point that further discarding is considered destructive. The sample is a drawn marker stamp on a glass slide.

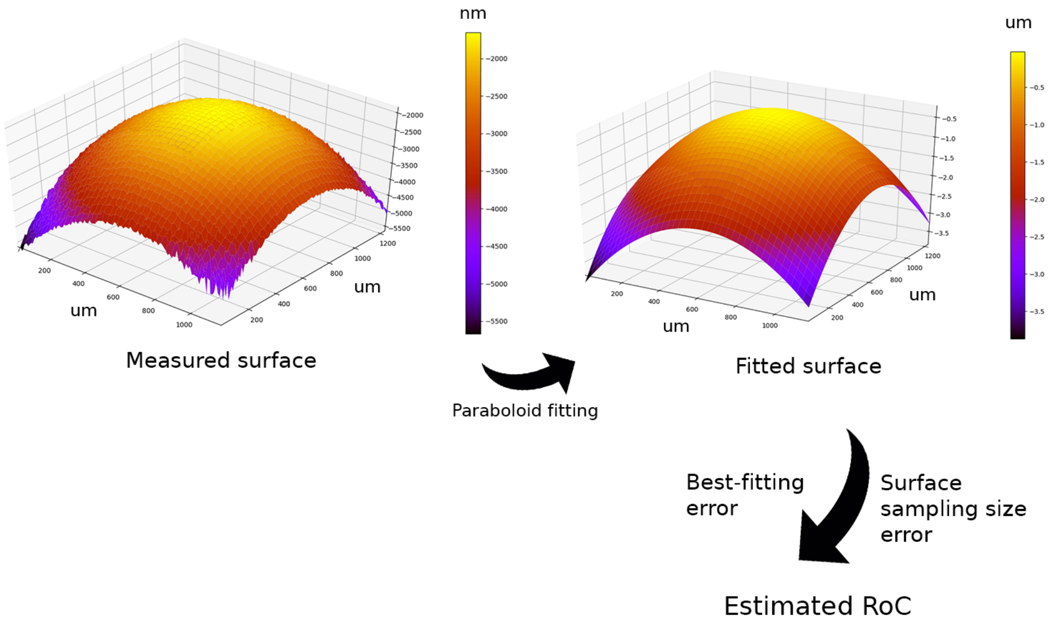


FIGURE 3.14: Assessment of the RoC of a paraboloidal surface. The shown profile is the front surface of a plano-convex lens.

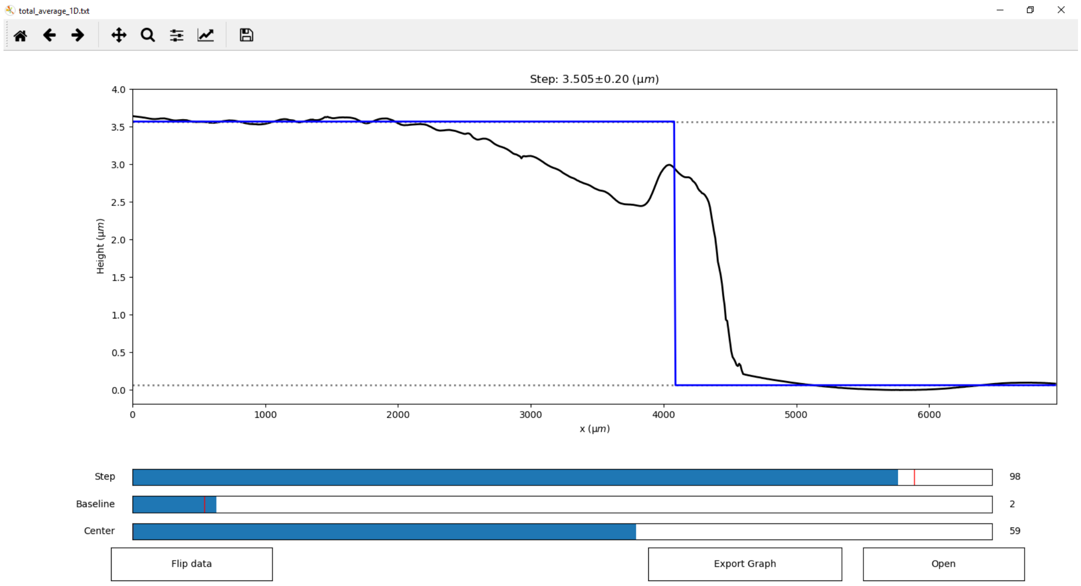
3.6 Calibration of vertical and transverse axes of the profiles

CALIBRATION OF VERTICAL AXIS

As stated in Sec. 2.3 and shown in Fig. 2.14, the piezo actuator moves in a non-linear fashion and also without a feedback system for corrections of the movements. For these reasons, the actual displacements need to be calibrated. The interference phenomenon is also useful in this context (see Eq. 2.12). The calibration technique that follows relies on the Fourier-transform method of phase-shift determination [24]. Since the terms $\Delta g\boldsymbol{r} + \Delta\phi_0$ incorporate the OPD of the two interferometer arms, it is evident that by adding a shift Δz to one of the arms, the phase of the interference pattern will shift in the axis of the added Δz (z-axis). Hence, the interferogram appears to shift transversely in the camera by the same phase shift as in the z-axis. The addition of a specific shift Δz in one arm results in an OPD equal to $2\Delta z$. Therefore, when one arm moves by Δz, the interferogram shifts by $2\Delta z$. The relation of the phase shift and the added Δz is straightforward:

$$\frac{\Delta\phi}{2\pi} = \frac{2\ \Delta z}{\lambda} \tag{3.1}$$

To perform the above, the interferometer must be configured with one flat mirror in each arm. For the addition of OPLs, the piezo actuator is utilized to scan iteratively and record the phase shift per scan. The finer the displacement step of the piezo actuator, the better. Likewise, the scan movements must be much smaller than the wavelength of the light source, which is chosen to be a laser source emitting at $670nm$.


total_average_1D.txt

Step: 3.505±0.20 (μm)

Step 98

Baseline 2

Center 59

FIGURE 3.15: Assessment of the step height of a cut polymer sample put on glass.

To measure each phase shift, the recordings of two consecutive interferograms must be compared. Since, the interferogram is a sinusoid, it can be easily visualized that each spatial shift imposes a shift of the sinusoidal spatial frequency in phase, due to the Fourier decomposition. This relation holds for every signal and it is expressed as (for 1D signals):

$$\mathcal{F}\{g(z-\Delta z)\} = e^{-i2\pi f \Delta z} G(f)$$

, where Δz : shift of g function (interferogram image), G : F.T of g.
This processes is parallelized using the following approach:

- An F.T is executed which produces a complex image result. This image contains information about the spatial frequencies participating in the formation of the original image. Particularly, when having a pure 2D sinusoidal waveform with some angle with the horizontal axis (tilt axis), the resulted F.T has two spots which correspond to the single spatial frequency of the signal, along the normal to the tilt axis. Specifically, the two spots are the equal amplitudes of the conjugate frequencies. Practically though, a realistic signal usually contains an offset value, lower frequencies that form the general envelope of the signal, medium frequencies that form the sinusoid and higher that exist due to noise. When the noise of the signal is uniform to some degree, the F.T result of the above is an image with three discernible spread spots.

- The spatial frequencies that contain the sinusoid information appear as spots located symmetrically to the zero spatial frequency.

- The peak position of one of two points is indetified and then the phase at this specific spatial frequency is retrieved as the angle of the complex exponential that corresponds to this frequency.

The calculated phases are bounded in the range $[0, 2\pi]$, thus they need to be unwrapped in the range $[0, \infty)$. Then, the differences between the consecutive pairs

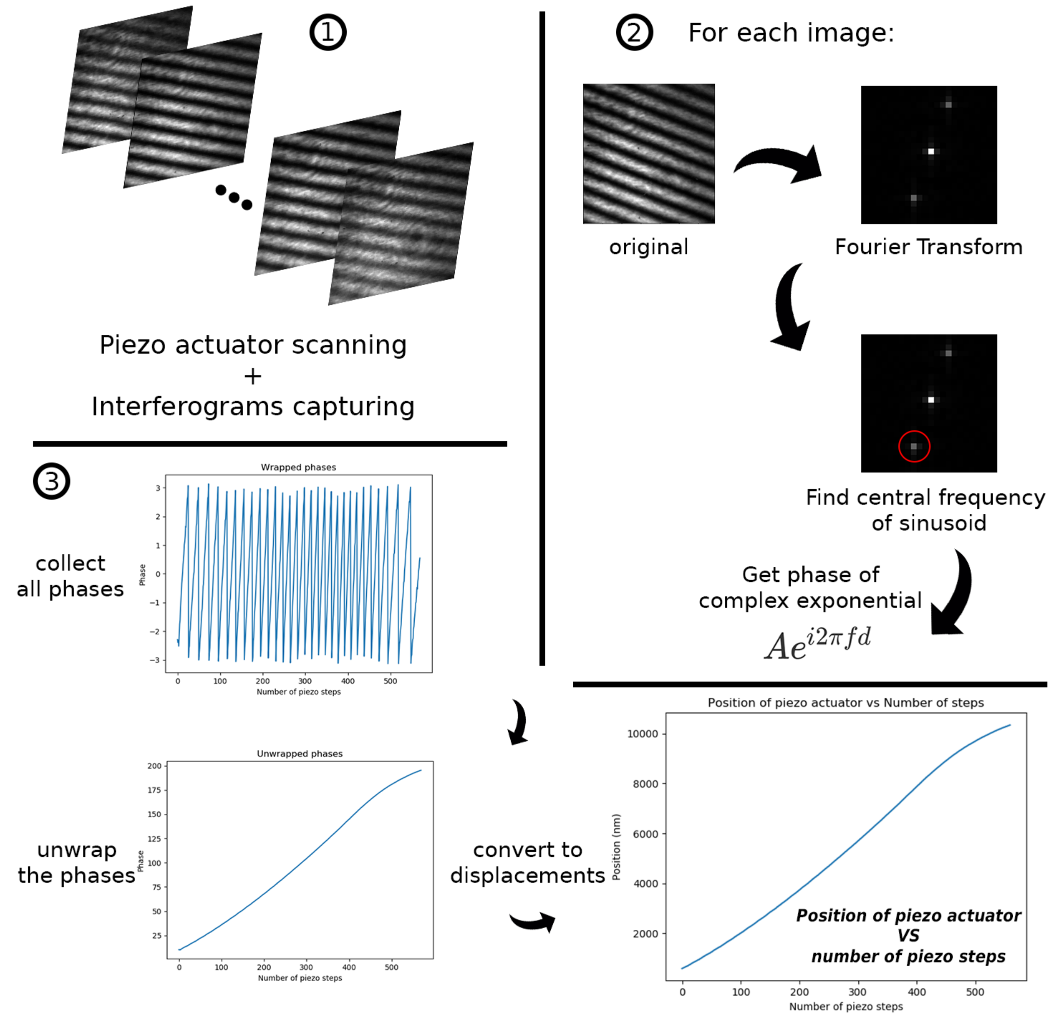


Figure 3.16: Procedure followed for the calibration of the displacements of the piezo actuator.

are calculated to obtain all the phase shifts. Using Eq. 3.1 the phase shifts are converted to OPL shifts of the reference arm or else to displacements of the piezo actuator. The final step is to create a mapping of the actual position of the piezo actuator with respect to the number of scans/steps.

The above procedure is shown with steps in Fig. 3.16.

Reproducibility and errors

To test the accuracy of the displacement of the piezo actuator per scan (positioning tolerance), calibration was repeated 10 times. Fig. 3.17 shows the maximum differences per scan which range from 11 to 44 nm nm with a median of 26 nm. Additionally, the differences between the 1st and 3rd quartiles of the samples per scan were calculated which range from 1 to 24 nm with a median of 11 nm. Assuming homoscedasticity in the variance of piezo step displacements, we can argue that the uncertainty of the piezo step displacement equals the median value of 26 nm.

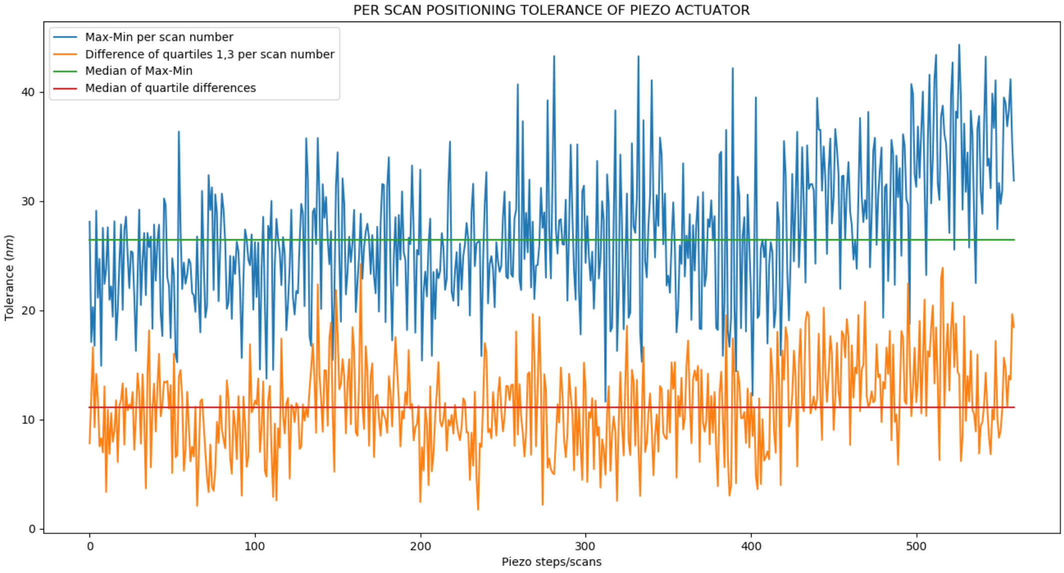


FIGURE 3.17: Accuracy of the displacement of the piezo actuator per scan (positioning uncertainty)

CALIBRATION OF TRANSVERSE AXIS

For the transverse calibration of the optical setup with the Michelson configuration, a precision reference micrometer scale imprinted upon a glass slide was used. The minor spacings of this scale have nominal distance of $10\mu m$, whereas the major ones of $50\mu m$. An image of this scale was captured by the current optical system. Measurements of the whole scale ($500\mu m$) show that it corresponds to 50 pixels on the camera sensor. Hence, the pixel size is $10\mu m$. Also the uncertainty (σ) of the pixel size must be found. In the captured image, it is clearly seen that the scale spacings are "sharp" even after the $500\mu m$ distance. Since, the measurement is determined by the intensity of light and there is no objective threshold so as to count or reject a pixel, let say the measured distance is $+1$ pixel, that is 51. In order for this to be true, the accumulated uncertainties per pixel should be equal to:

$$\frac{real \quad pixel \quad size}{distance \quad in \quad pixels} = \frac{10\mu m}{50} = 0.2\mu m \tag{3.2}$$

From this it holds that the calibrated pixel size for the optical setup with the Michelson configuration is $10 \pm 0.2\mu m$.

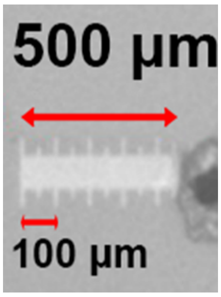


FIGURE 3.18: Image of the micrometer scale used for the transverse calibration of the optical setup withe the Michelson configuration.

Chapter 4

Experimental Results

The categories of the various measured samples is listed below:

1. Flat mirror
2. Stylus tip calibration reference sample
3. Lenses
4. Spin coated polymer films
5. Fingerprints
6. Marker stamps

4.1 Flat mirror

The first test for the reliability of the optical setup and the profile analysis methods is to measure a flat mirror. The selected sample is a round mirror (ME1-G01). The sample and the results are shown below:

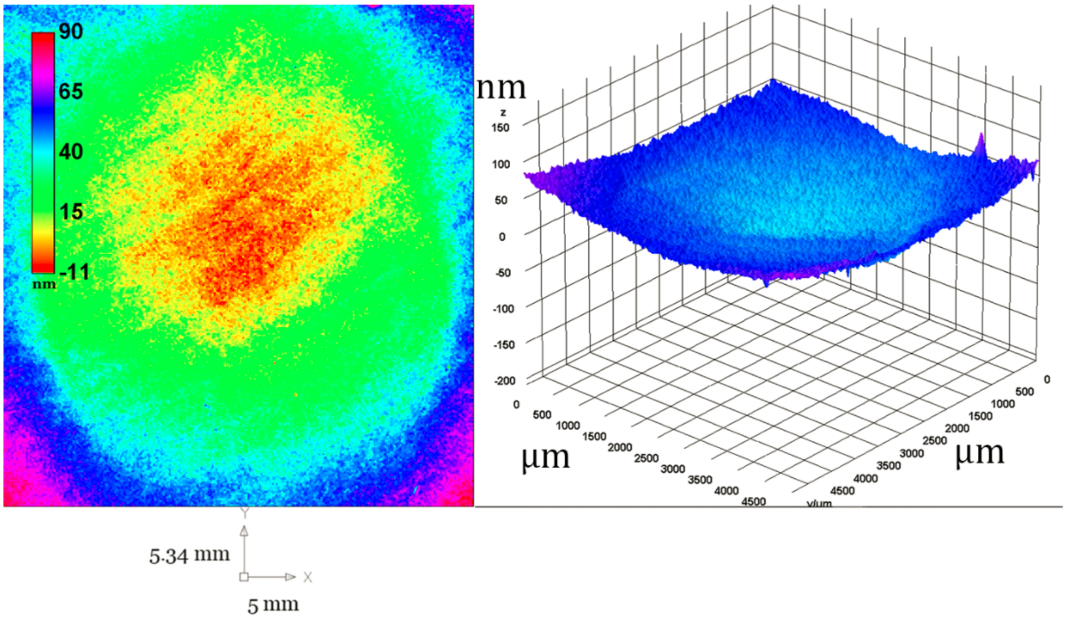


Figure 4.1: 2D and 3D surface profiles of the round mirror

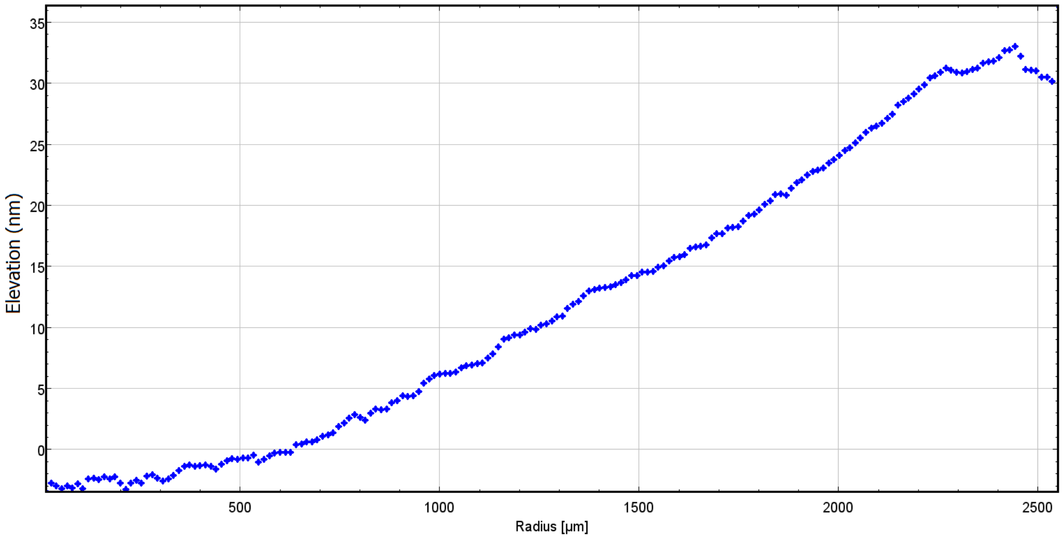


FIGURE 4.2: 1D radial average surface profile of the round mirror. Average elevation for a given distance from the center of a circle (center of spherical surface)

	M [nm]	Sp [nm]	Sv [nm]	Sz [nm]	Sa [nm]	Sr [nm]	Sq [nm]	Ssk	Sku
Mirror surface	0	154	228	382	12	7	15	1.62	3.80

4.2 Stylus tip calibration reference sample

A dent that has been engraved by a reliable stylus tip has been measured. The step elevation of the sample agrees with the tolerances of the manufacturer (9.6μm). The step uncertainty is the sum of the roughnesses of the top and bottom elevation levels. The sample and the results are shown below:

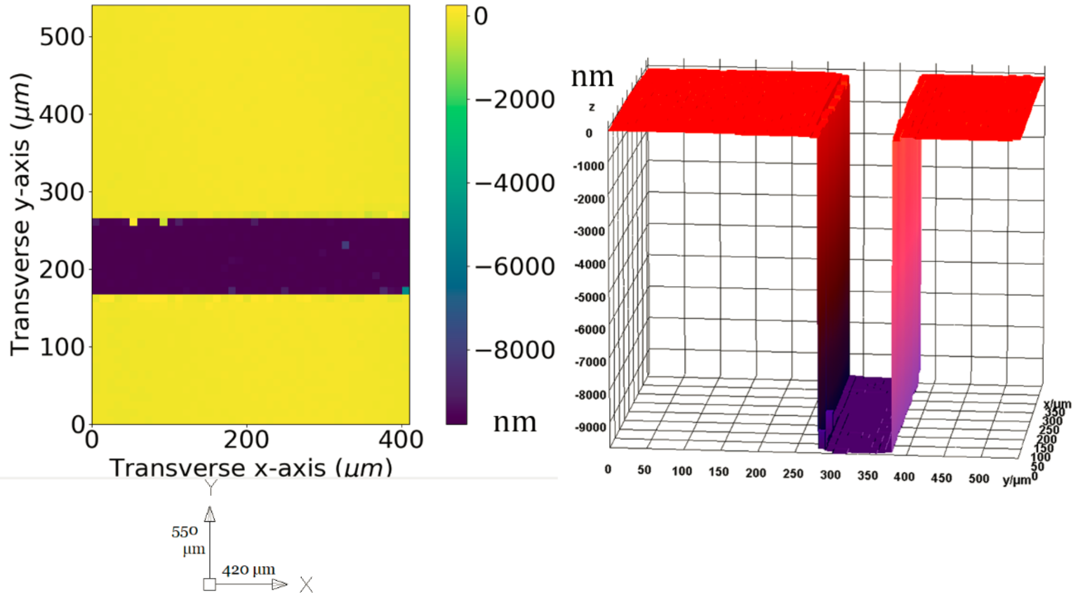


FIGURE 4.3: 2D and 3D surface profiles of the Stylus tip calibration reference sample

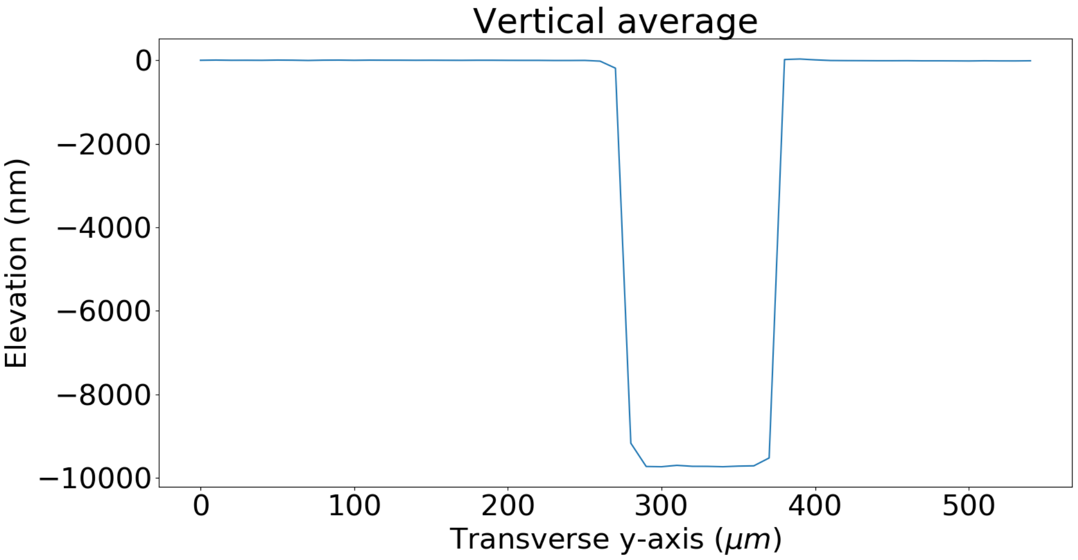


FIGURE 4.4: 1D vertical average surface profile of the Stylus tip calibration reference sample

	Sa [nm]	Sr [nm]	Sq [nm]
Top level	93	112	115
Bottom level	86	106	109
Step elevation [um]	9.7±0.2		

Table 4.1: Surface topography metrics for the sample engraved by a stylus tip

4.3 Spherical Lenses

Two types of lenses were measured in terms of their RoC. First, optical lenses always found on a laboratory, of RoC larger than $25mm$. Second, a set of contact lenses of nominal front surface RoC around $8mm$ were measured. For the CLs, the front surface curvature (FC) is calculated through the back surface curvature (BC), the back vertex power and the central thickness of the CL. Roughly the particular FCs and BCs are very close. Due to the very small RoC of the CLs, the Michelson configuration used yields mediocre results in terms of RoC uncertainty. For sample points further from the CL center, the OPD from the reference mirror increases in a quadratic fashion, thus producing denser and denser fringes. This means that after a while the fringes totally disappear. The samples and the results are shown below:

Lenses - #1 |Plano-convex lens : f = 200 mm (LA4102)

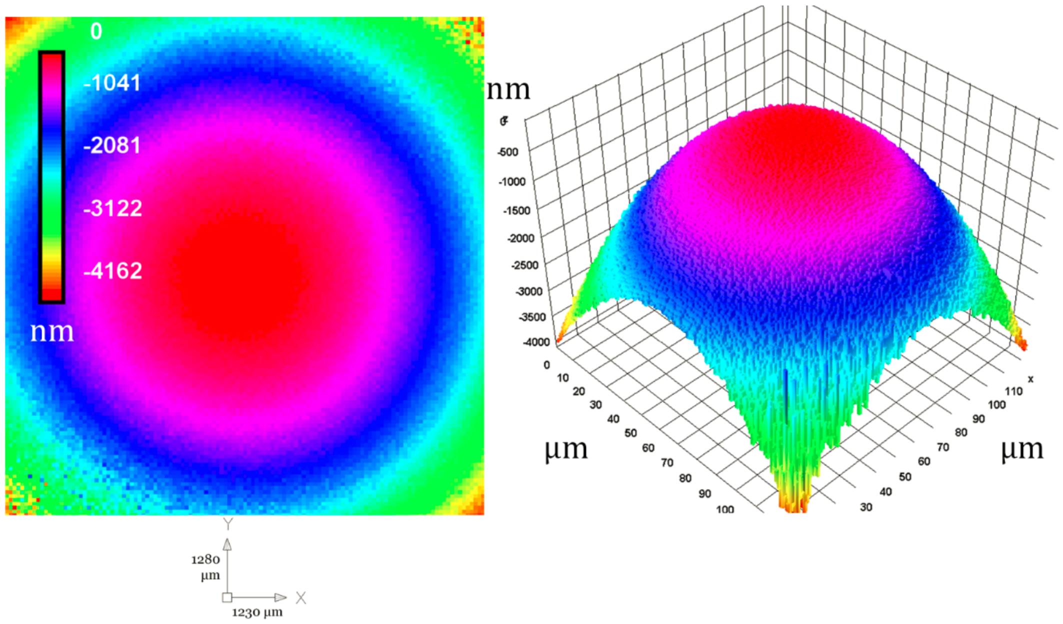


Figure 4.5: 2D and 3D surface profile of Lens #1 |Plano-convex lens : f = 200 mm (LA4102)

Above results show a relative error of RoC ≈ 9%. The theoretical estimations (see Appendix G.1) show relative error ≈ 1%.

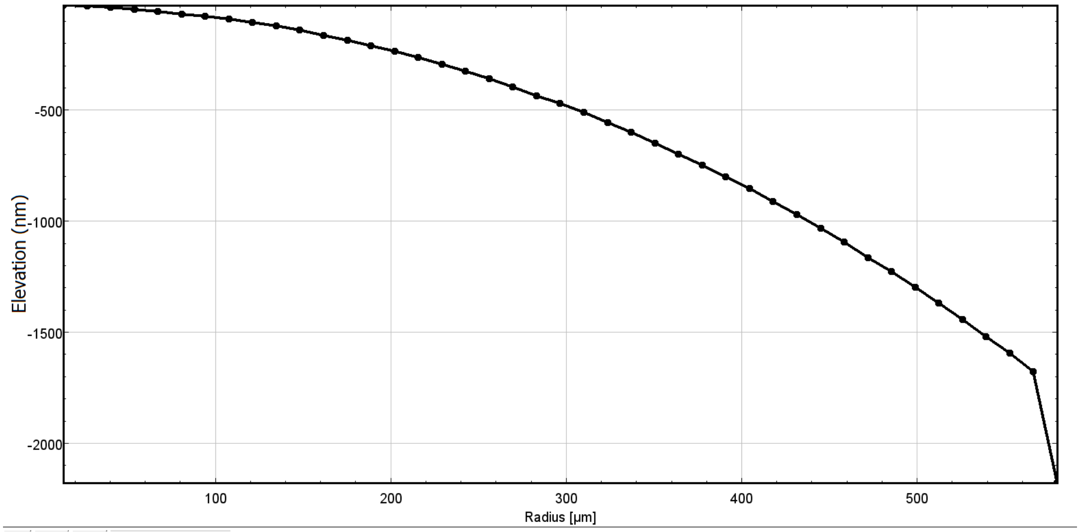


Figure 4.6: 1D radial average surface profile of Lens #1 |Plano-convex lens : f = 200 mm (LA4102). Average elevation for a given distance from the center of a circle (center of spherical surface)

Expected RoC	92.0 ±0.9 mm
Measured RoC	99.3 ± 4.0 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 of paraboloid fit$	99.3%

	Field of View	Total elevation
Measured area and elevation	1230 x 1280 μm^2	$\sim 4\mu m$

Table 4.2: Aggregated results for Lens #1 |Plano-convex lens : f = 200 mm (LA4102)

Lenses - #2 |Bi-convex lens (doublet) : f = 25 mm (AC127-025-A)

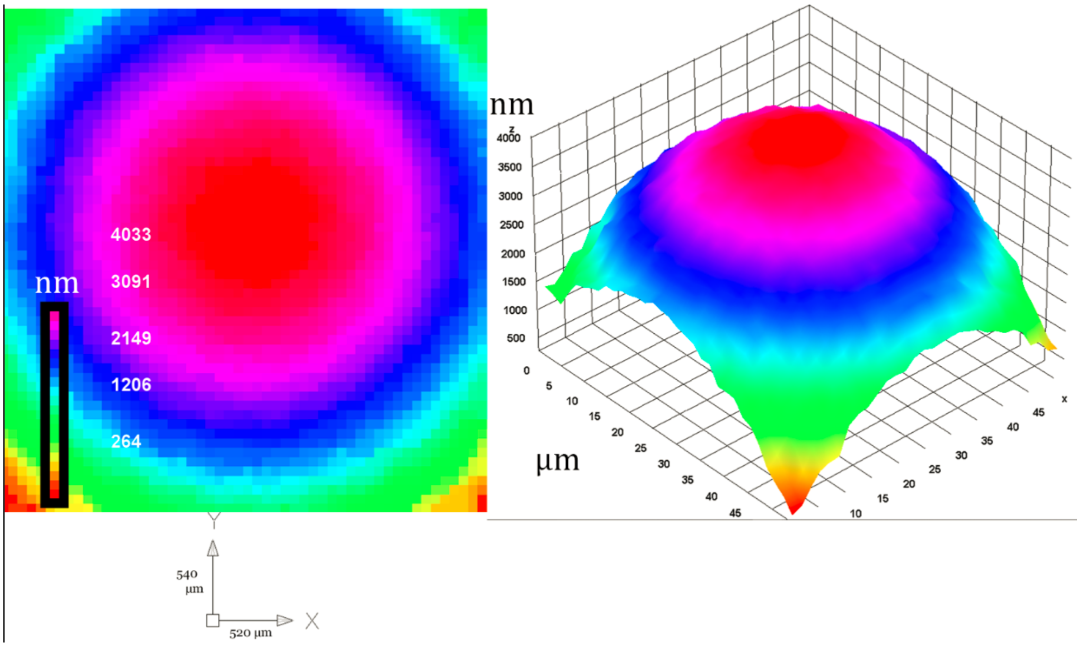


Figure 4.7: 2D and 3D surface profile of Lens #2 |Bi-convex lens (doublet) : f = 25 mm (AC127-025-A)

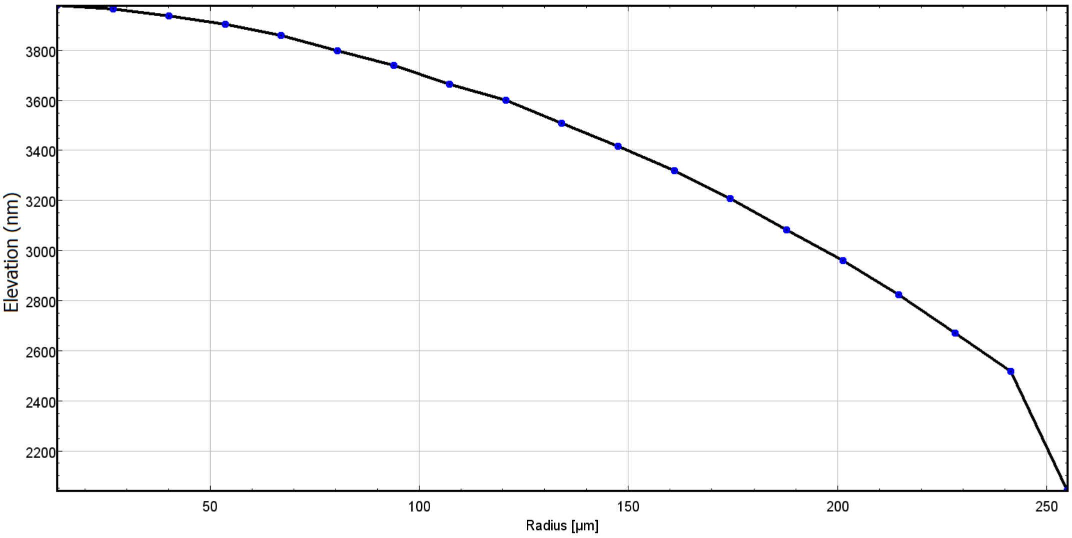


Figure 4.8: 1D radial average surface profile of Lens #2 |Bi-convex lens (doublet) : f = 25 mm (AC127-025-A). Average elevation for a given distance from the center of a circle (center of spherical surface)

Above results show relative error of RoC ≈ 12%. The theoretical estimations (see Appendix G.1) show relative error ≈ 5%.

Expected RoC	18.8 mm
Measured RoC	21.1 ± 1.0 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2ofparaboloidfit$	91.31%

	Field of View	Total elevation
Measured area and elevation	520 x 540 μm^2	$\sim 4\mu m$

Table 4.3: Aggregated results for Lens #2 |Bi-convex lens (doublet) : f = 25 mm (AC127-025-A)

Contact Lenses set (CL) – holder

For the stable holding of the CLs, a metallic ball of similar RoC (8 mm) was used and mounted on the sample holder of the interferometer (Fig. 4.9). 3 measurements of the ball holder were made. Below, the profile giving the best coefficient of determination (fitting quality) r^2, is shown.

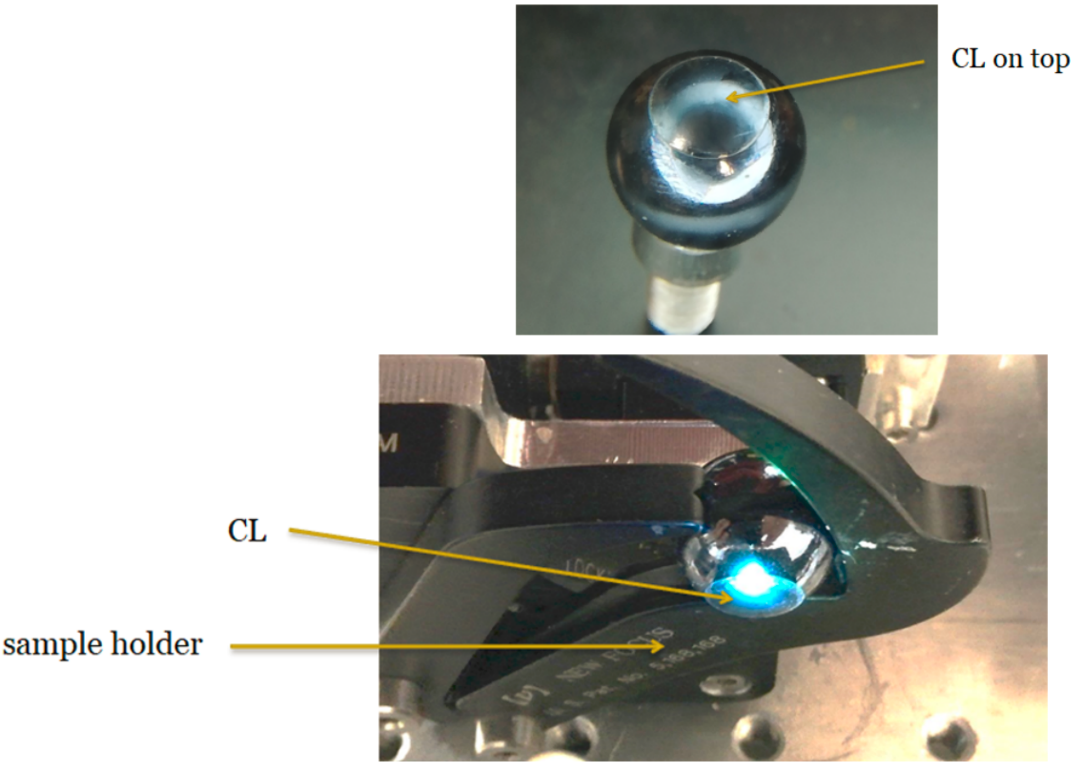


Figure 4.9: For the stable holding of the CLs, a metallic ball of similar RoC (8 mm) was used and mounted on the sample holder of the interferometer

Fig. F.2 shows the estimated nominal RoC of the front surface of the measured CLs of this thesis.

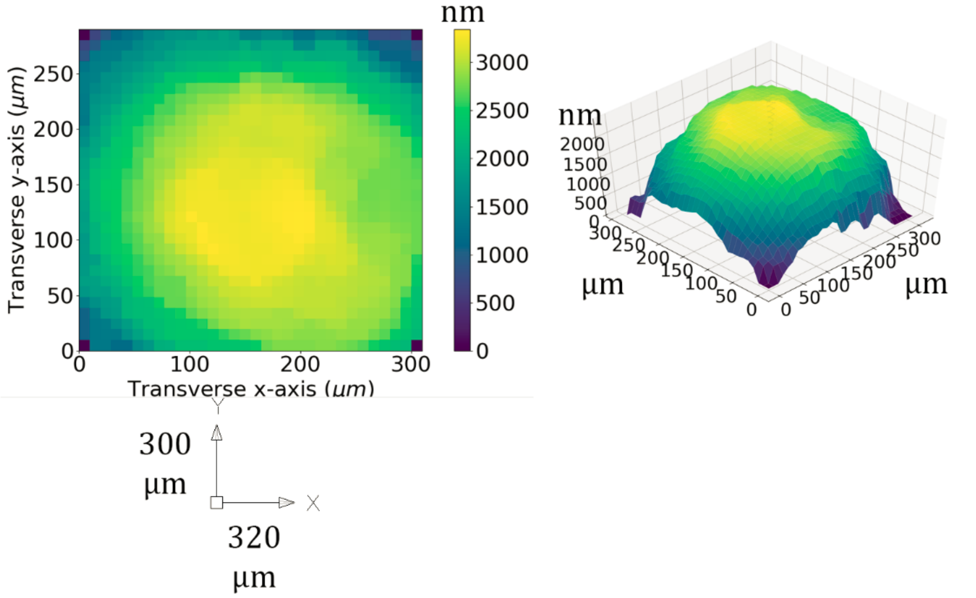


FIGURE 4.10: 2D and 3D surface profile of ball holder of CLs

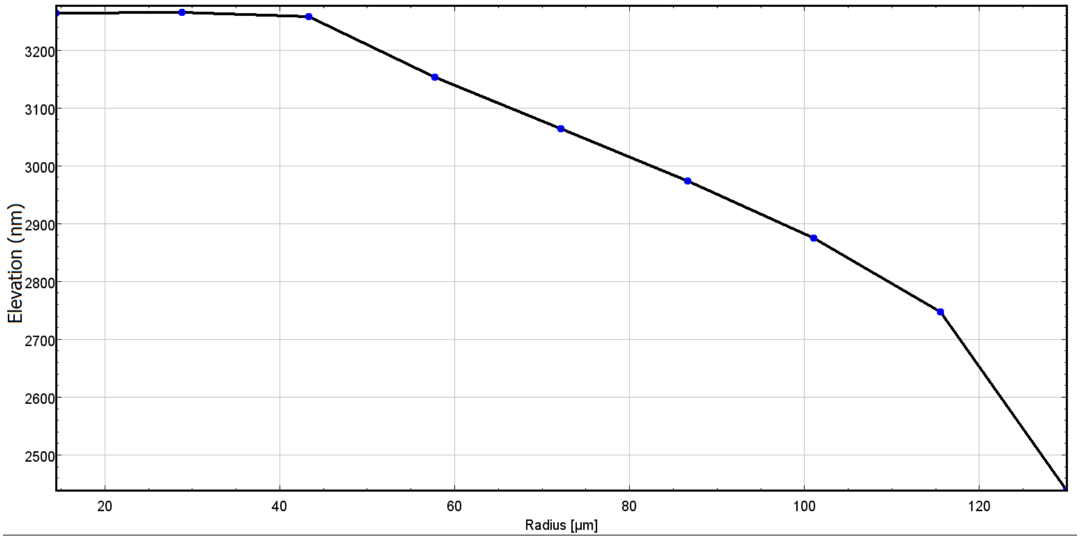


FIGURE 4.11: 1D radial average surface profile of ball holder of CLs. Average elevation for a given distance from the center of a circle (center of spherical surface)

Expected RoC	8.0 mm
Measured RoC	10.7 ± 0.6 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 of paraboloidfit$	78.37%

	Field of View	Total elevation
Measured area and elevation	320x 300μm^2	$\sim 2.75\mu m$

TABLE 4.4: Aggregated results for ball holder of CLs

Lenses - #3 |CL ($BC = 7.50$ mm)

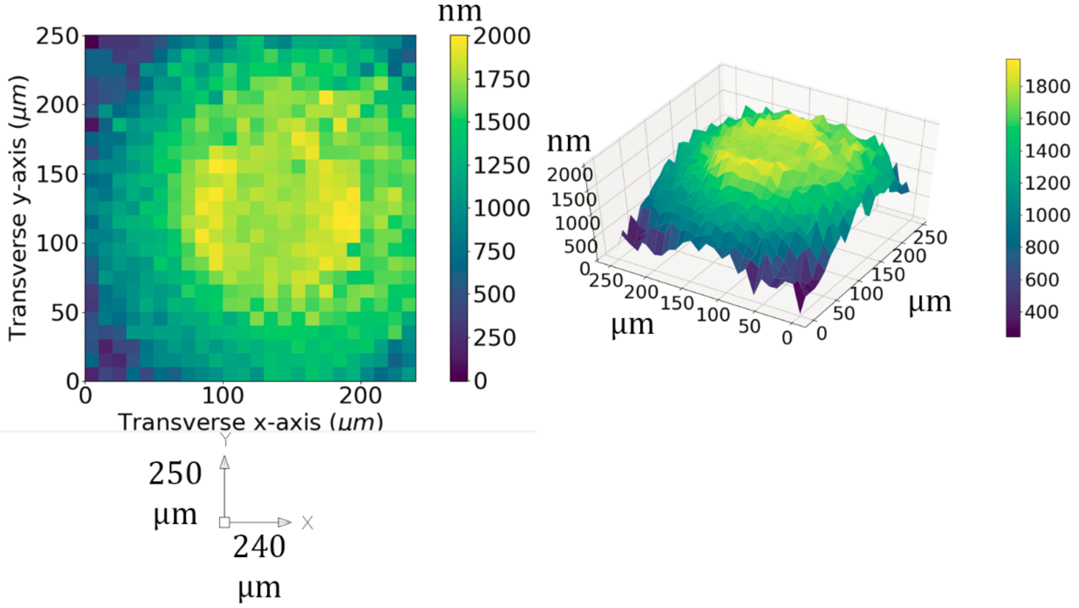


FIGURE 4.12: 2D and 3D surface profile of CL ($BC = 7.50$ mm)

Expected RoC	7.95 mm
Measured RoC	11.0 ± 1.0 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 of paraboloid fit$	56.61%

	Field of View	Total elevation
Measured area and elevation	240 x 250μm^2	$\sim 1.8\mu m$

TABLE 4.5: Aggregated results for CL ($BC = 7.50$ mm)

Lenses - #3 |CL ($BC = 7.70$ mm)

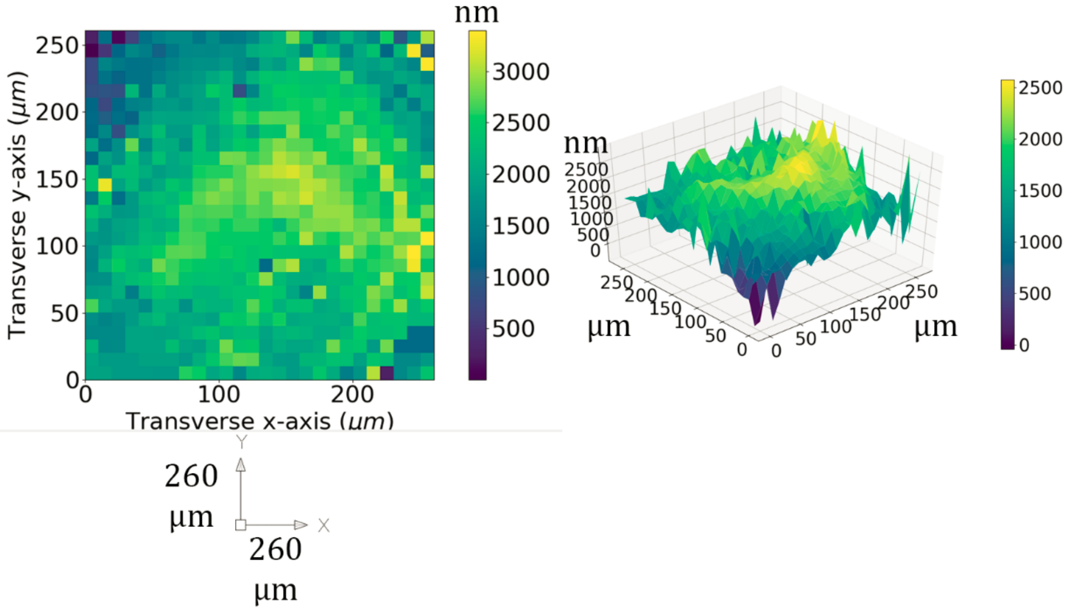


FIGURE 4.13: 2D and 3D surface profile of CL ($BC = 7.70$ mm)

Expected RoC	8.20 mm
Measured RoC	9.6 ± 1.1 mm
Surface sampling size	$10.0 \pm 0.2\ \mu m$
$r^2 of paraboloid fit$	25.63%

	Field of View	Total elevation
Measured area and elevation	260 x 260 μm^2	$\sim 2\mu m$

TABLE 4.6: Aggregated results for CL ($BC = 7.70$ mm)

Lenses - #3 |CL ($BC = 7.85$ mm)

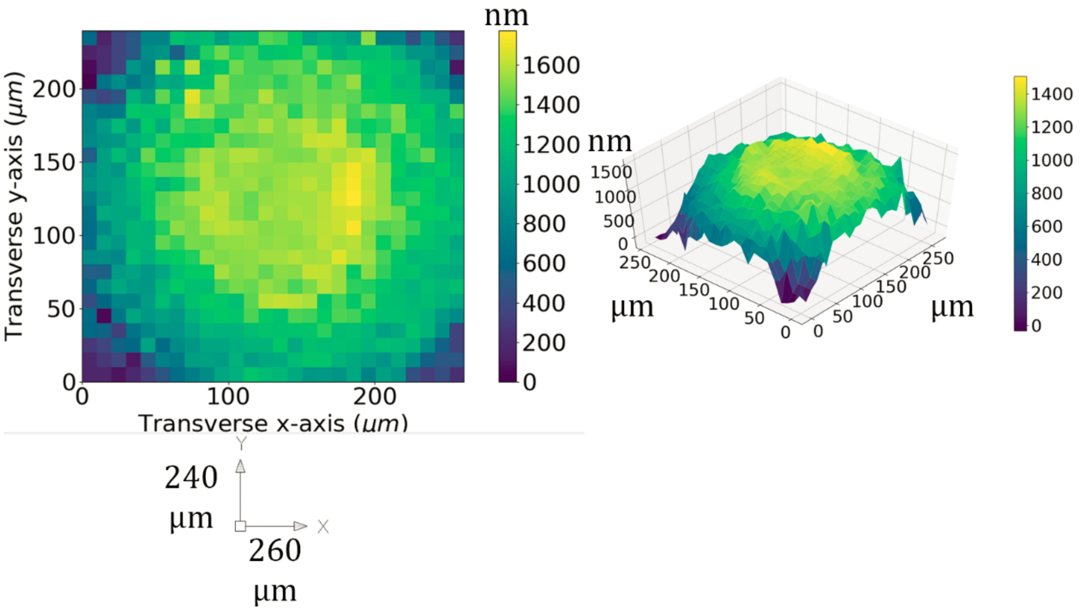


FIGURE 4.14: 2D and 3D surface profile of CL ($BC = 7.85$ mm)

Expected RoC	8.34 mm
Measured RoC	10.2 ± 0.7 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 ofparaboloidfit$	68.26%

	Field of View	Total elevation
Measured area and elevation	260 x 240 μm^2	$\sim 1.7 \mu m$

TABLE 4.7: Aggregated results for CL ($BC = 7.85$ mm)

Lenses - #3 |CL ($BC = 7.90$ mm)

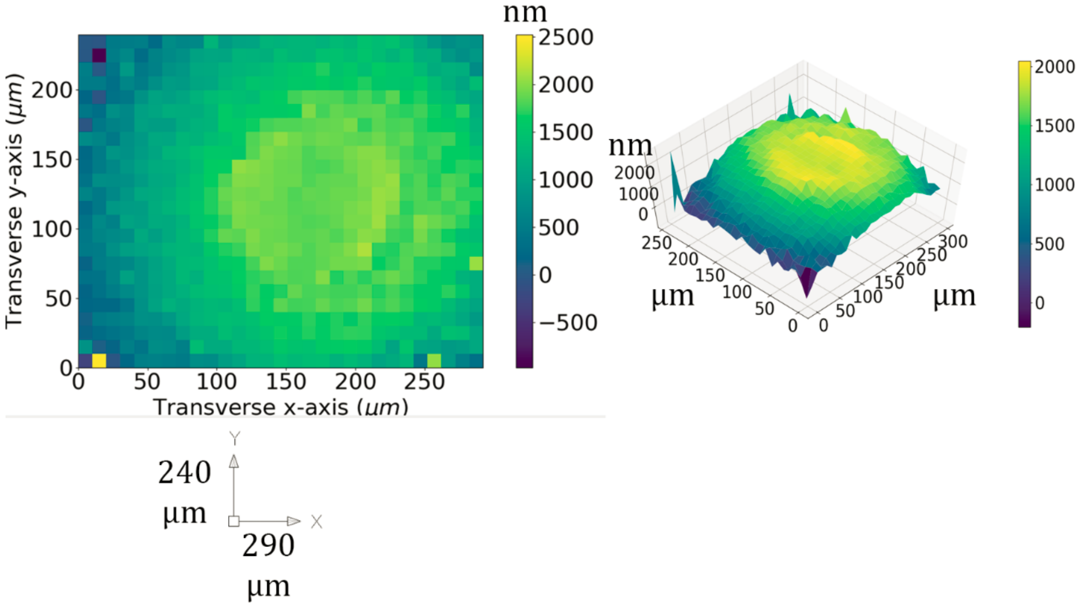


FIGURE 4.15: 2D and 3D surface profile of CL ($BC = 7.90$ mm)

Expected RoC	8.40 mm
Measured RoC	10.2 ± 0.7 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 of paraboloid fit$	71.42%

	Field of View	Total elevation
Measured area and elevation	290 x 240 μm^2	$\sim 2.2 \mu m$

TABLE 4.8: Aggregated results for CL ($BC = 7.90$ mm)

Lenses - #3 |CL (BC = 7.95 mm)

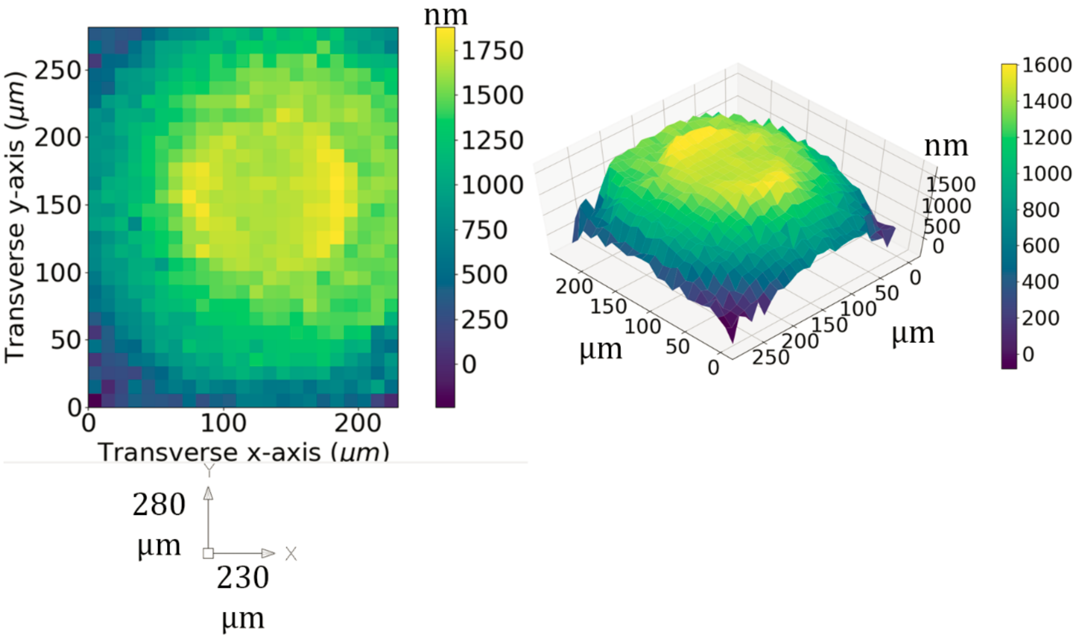


FIGURE 4.16: 2D and 3D surface profile of CL (BC = 7.95 mm)

Expected RoC	8.45 mm
Measured RoC	10.7 ± 0.6 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 ofparaboloidfit$	92.94%

	Field of View	Total elevation
Measured area and elevation	230 x 280 μm^2	$\sim 2\mu m$

TABLE 4.9: Aggregated results for CL (BC = 7.95 mm)

Lenses - #3 |CL ($BC = 8.20$ mm)

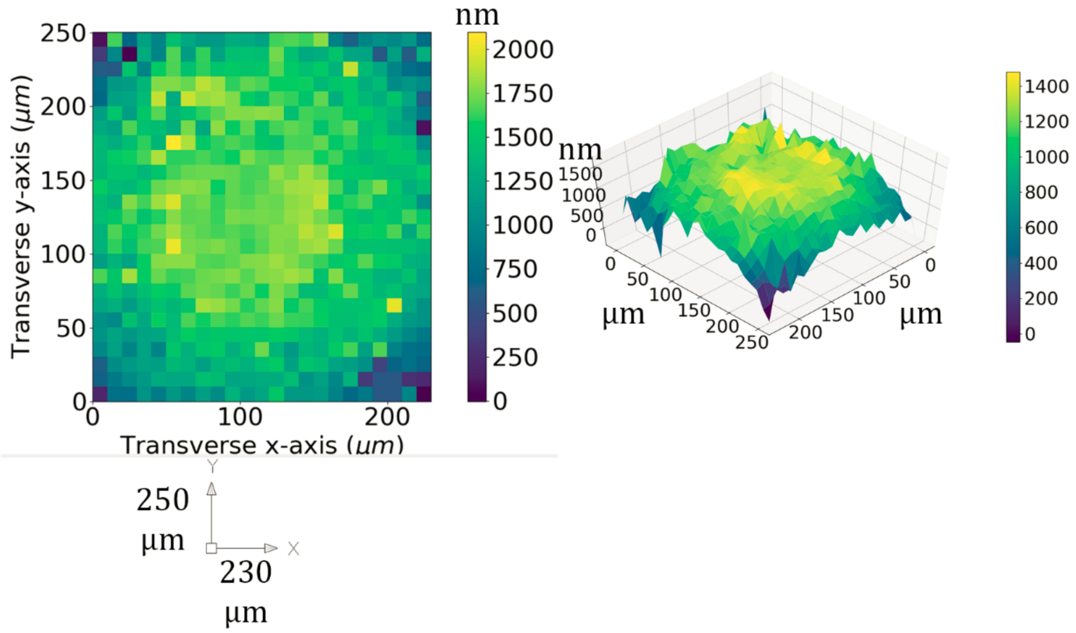


FIGURE 4.17: 2D and 3D surface profile of CL ($BC = 8.20$ mm)

Expected RoC	8.73 mm
Measured RoC	11.2 ± 0.9 mm
Surface sampling size	10.0 ± 0.2 μm
$r^2 of paraboloidfit$	53.50%

	Field of View	Total elevation
Measured area and elevation	230 x 250 μm^2	$\sim 1.2 \mu m$

TABLE 4.10: Aggregated results for CL ($BC = 8.20$ mm)

Lenses - #3 |CL set

Below, the aggregated results are displayed altogether.

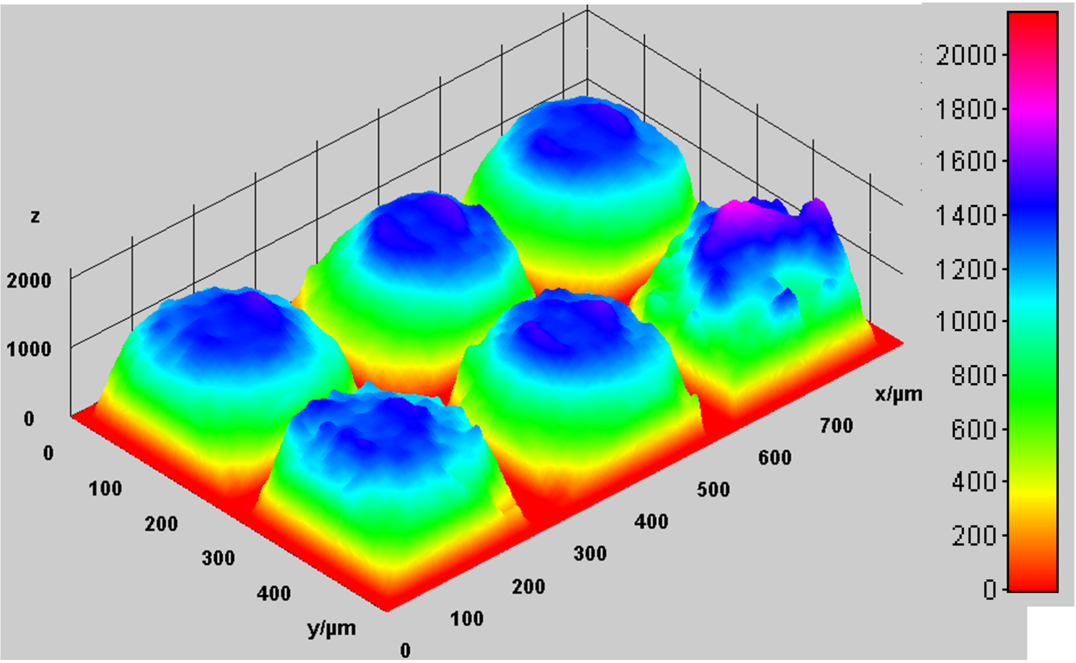


Figure 4.18: 3D surface plots for the six measured CLs altogether

BC [mm]	Expected FC [mm]	Measured FC [mm]	r^2	Measured area x elevation [μm]
7.50	7.95	11.0 ± 1.0	56.61 %	240 x 250 x 1.8
7.70	8.20	9.6 ± 1.1	25.63 %	260 x 260 x 2.0
7.85	8.34	10.2 ± 0.7	68.26 %	260 x 240 x 1.7
7.90	8.40	10.2 ± 0.7	71.43 %	290 x 240 x 2.2
7.95	8.45	10.2 ± 0.6	92.94 %	230 x 280 x 2.0
8.20	8.73	11.2 ± 0.9	53.50 %	230 x 250 x 1.2

Table 4.11: Aggregated results for CL set

Above results show relative errors of RoC that range from 15 to 38 %. The theoretical estimations (see Appendix G.1) show relative error ≈ 10%. This is due to the fact that, as shown in the previous table, the fittings do not present a good degree of fit (over 90 %), except for one CL.

4.4 Spin coated polymer films

The thin polymer films were prepared by the 2PP group in FORTH and were coated on top of typical microscope glass slides. They were transparent in visible light. Our goal was to measure the thickness of the polymer film. For this purpose, in the edge of the polymer layer, a step was created. For the calculation of nominal thickness, the glass surface was used as a reference. The thickness was estimated ignoring a bump appearing at the step edge since this was related to the step fabrication method. For all the surface topography metrics, the inordinately high peaks are ignored (e.g the 3 high peaks shown in Fig. 4.20 in the main polymer area). Because of the inclination of the polymer films, the mean plane metric (M) was calculated in the inclined polymer area, where as the rest metrics (roughness, etc.) were calculated in the leveled polymer area (plane rotation to meet the horizontal axes). The samples and the results are shown below:

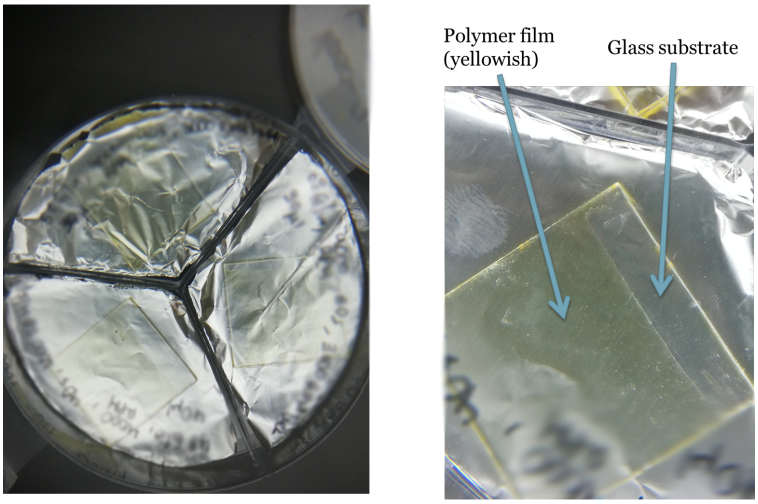


FIGURE 4.19: Polymer film on glass substrate with the fabricated step in the polymer edge

Polymer films - #1

	M [nm]	Sp [nm]	Sv [nm]	Sz [nm]	Sa [nm]	Sr [nm]	Sq [nm]	Ssk	Sku
Polymer film (Top level)	1473	191	185	376	49	32	60	1.53	2.72
Glass (Bottom level)	-5	262	318	581	31	19	38	1.54	2.83
Step elevation [μm]	1478								
Step uncertainty [μm]									

TABLE 4.12: Surface topography metrics for Polymer film - #1

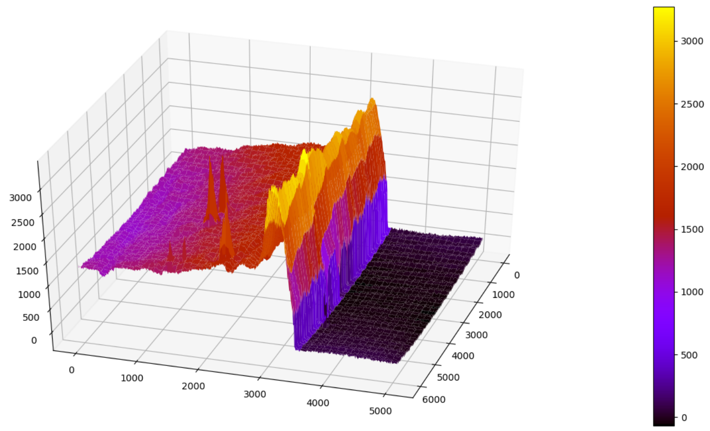


FIGURE 4.20: Surface profile (3D) of Polymer film - #1

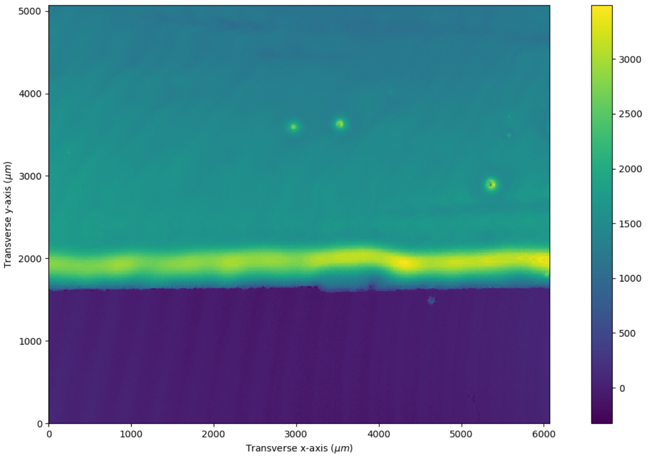


FIGURE 4.21: Surface profile (2D) of Polymer film - #1

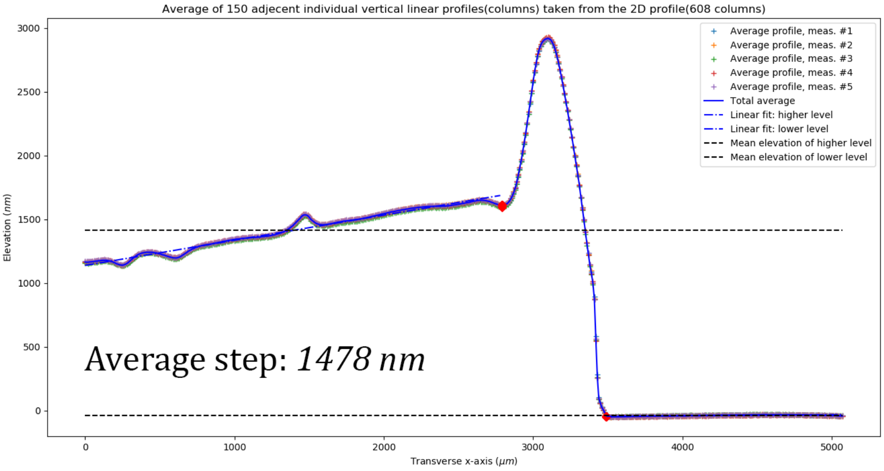


Figure 4.22: Surface profile (1D) of Polymer film - #1. Averaging over horizontal direction.

Polymer films - #2

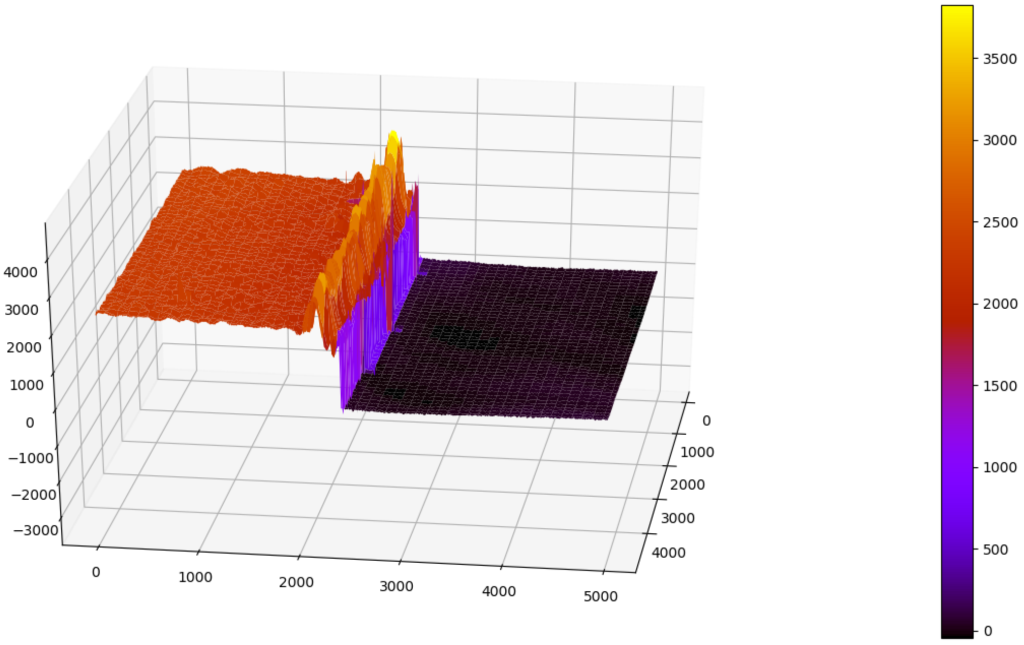


FIGURE 4.23: Surface profile (3D) of Polymer film - #2

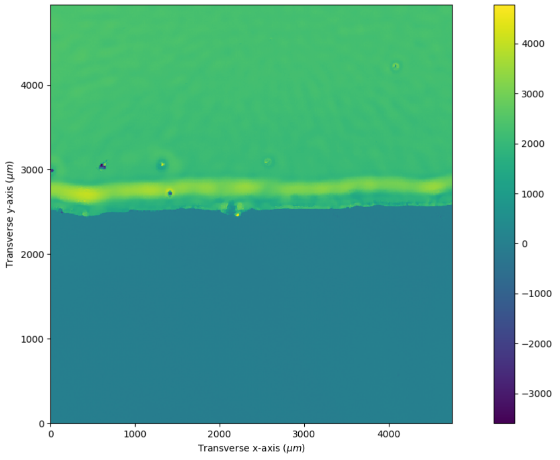


FIGURE 4.24: Surface profile (2D) of Polymer film - #2

4.5 Marker stamps

Just to add some more elegant samples and for purposes of visual indulgence, a few numbers were written with a marker on a glass slide. The samples and the results are shown below:

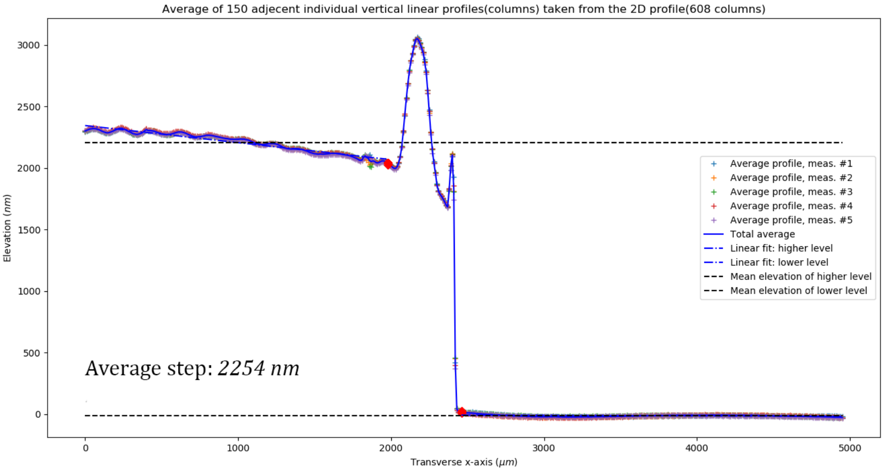


Figure 4.25: Surface profile (1D) of Polymer film - #2. Averaging over horizontal direction.

	M [nm]	Sp [nm]	Sv [nm]	Sz [nm]	Sa [nm]	Sr [nm]	Sq [nm]	Ssk	Sku
Polymer film (Top level)	2250	186	330	516	34	38	43	1.67	3.42
Glass (Bottom level)	-4	128	83	210	18	15	23	1.70	3.54
Step elevation [μm]	2254								
Step uncertainty [μm]									

Table 4.13: Surface topography metrics for Polymer film - #2

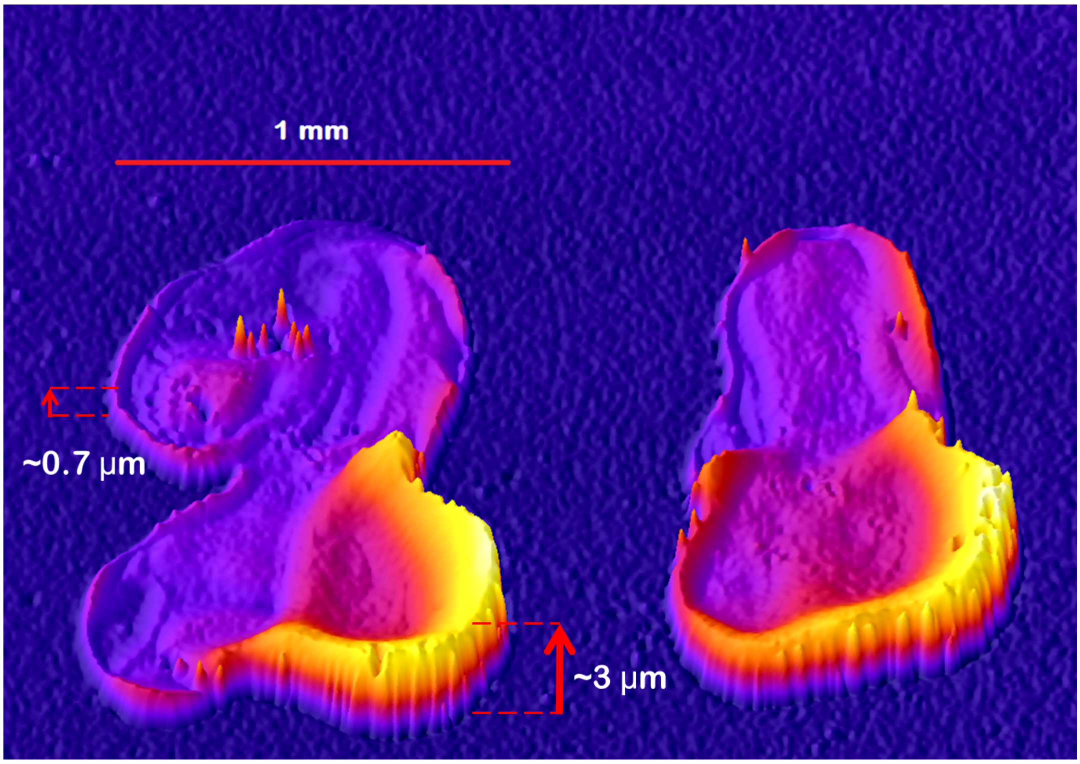


FIGURE 4.26: Surface profile (3D) of a marker stamp on a glass slide

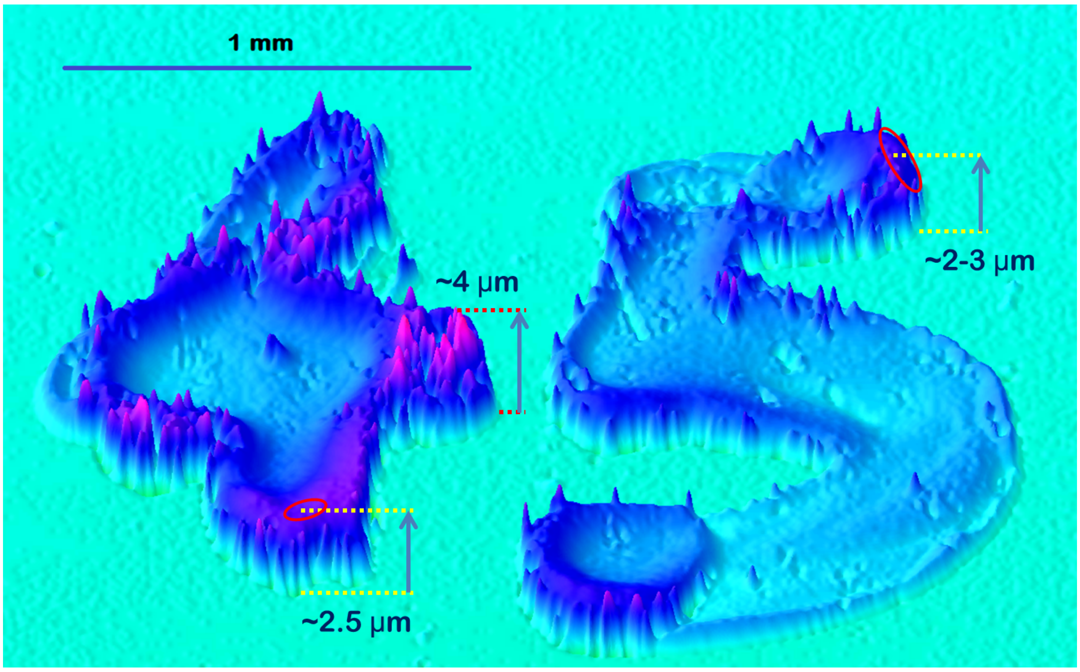


FIGURE 4.27: Surface profile (3D) of a marker stamp on a glass slide

Chapter 5

Conclusions and Future Work

Optical setup and results

The optical setup of the Michelson configuration is one of the simplest, most intuitive and straightforward interferometers. Utilizing PSI, it is often used for educational purposes in high schools rendering it relatively easy to build. The comprising parts are few in number and of course the most expensive ones are the optics. Depending on the application and the needed accuracy and precision, the reference mirror and beam splitter can raise the cost high enough. Regarding CSI, one extra part comes in which is the piezoelectric actuator. This device together with its driver consist the economic bottleneck.

In terms of the effectivenss of the Michelson interferometer, it is a robust device that yields very accurate results but only when it comes to samples that have small curvature. Usually, this device is used to test the flatness of optics in the order of a few nanometers. For curved samples, scientists resort to different solutions such as the Twymann-Green interferometer that adds a wavefront correcting lens in front of the curved sample.

Regarding the measured lenses, the results show relative errors in the estimation of the RoC of $\approx\;9,\;12,\;15-18\%$ for lenses with RoC nominally equal to $92,\;18.8,\;\approx 8$. On the other hand, theoretical estimations show relative errors $\approx\;1,\;5,\;10\%$. Fittings that present small residual errors, also have a large value of the coefficient of determination r^2. The results of the lenses with RoC equal to 92 and 18.8 mm show r^2 over 99%, whereas CLs exhibit poor fitting with a poor r^2.

The elevation resolution for the Michelson interferometer of this thesis is estimated to be ≈ 15 nm which is equal to the roughness of the used reference mirror.

Automation and analysis

The automation part is a one time startup cost that after being created, the measurements can be taken with a few clicks in the computer. As for the profile analysis and all the auxiliary computations that accompany the profile estimation, almost every procedure is integrated to the GUI.

Regarding the speed of calculations, the profile analysis is quite slow in order to yield the best possible result with the specific methodology. Attempts to speed up the analysis without compensating the results have been made and are yet to be completed.

Future work

- Build and test of the Twyman-Green setup for the measurements of curved samples with large curvature. Attempts have been made with forthcoming results.
- Speed up the profile analysis calculations. Many modifications have been tested but are not fully debugged with the most important of them to be the parallelization of calculations that can reduce the calculation time up to 4 times.
- Irrelevantly of optical elements, latent fingerprints are about to be measured for forensic purposes.

Appendices

Appendix A

Automation

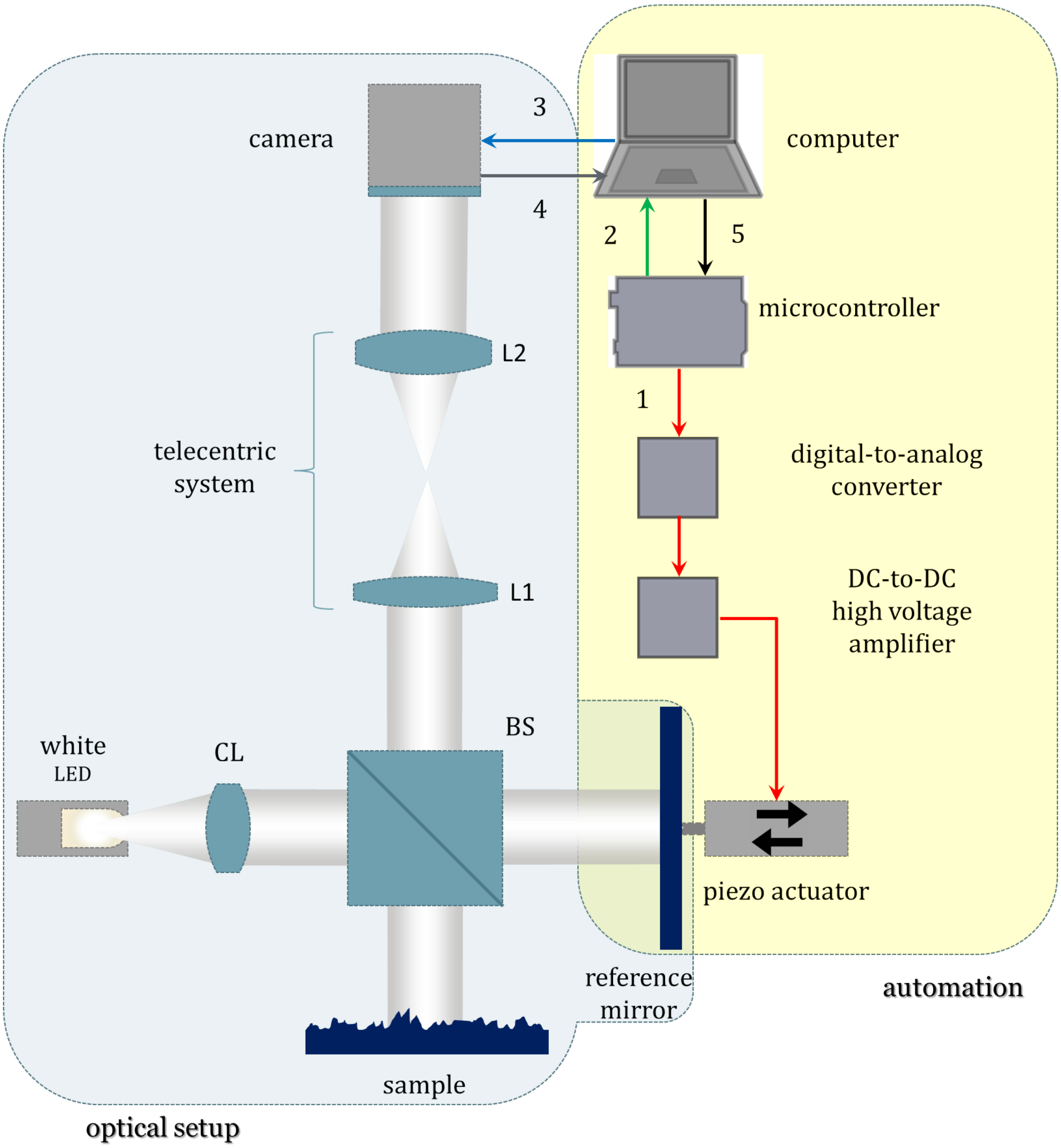


FIGURE A.1: Automated movement and image capturing system: Numbers correspond to the sequence of signal transmission in a loop.

Repeated and controlled changes of the OPL of the reference arm, needed for the interferogram analysis, are realised by the piezoelectric actuator. Of course, in each step of the piezo, an image must be captured. Hence, it is easily understandable that a *turn-based cooperative "game"* must be played between the piezo and the camera. The piezo carries out a displacement and then the camera captures an image of the interferogram. Then a second displacement is made and a second image is captured. This procedure is followed until the total number of needed interferograms is achieved. In order to achieve synchronization, two seperate flows of signals happen alternately in each moment. The first flow travels from the *Arduino towards the piezo* while the second one runs from the *computer to the camera and back.* Of course for these flows, in order to happen alternately, the computer and the Arduino must communicate with one another. For this purpose, the two devices communicate serially via the USB cable that connects them.

Piezoelectric Actuator

In the current thesis, a piezo actuator, named as PI P-840.1 [25], is used and it attains a nominal travel range of $15\mu m$ with a nominal resolution of $0.15nm$. Its required input voltage ranges from 0 to 100 V. It can sustain a pull force of 50 N and a push force of 1000 N. Specifically, in the optical arrangement, the pull force is used since the piezo pulls the attached mirror, increasing the OPL of the reference arm. As seen in Fig:A.2, on the upper part of the piezo there is a threaded hole which accomodates a screw on the top of which the reference mirror is mounted with epoxy glue.

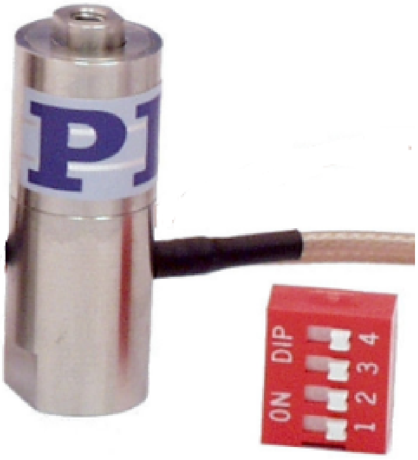


FIGURE A.2: Used piezoelectric actuator with a dip switch for size comparison. The threaded hole on the upper part, accomodates a screw on the top of which the reference mirror is mounted with epoxy glue.

Source: Phsyic Instrumente [25]

Driving the piezo

The piezo actuator is operated in open-loop mode. So, a calibration is operated ideally before each single measurement and a mapping between number of piezo step and actual displacement is created. Taking into consideration the aforementioned travel range and resolution, it is evident that the piezo can maximally provide $100,000$ steps of $0.15nm$ each. However, the capturing time and the volume of data of $100,000$ images for a measurement of a single sample will be certainly significantly high.

Having knowledge of each individual displacement of the piezo is certainly the optimal solution. Though, operating with a quite smaller amount of images, *thus bigger piezo steps*, is not a problem. This is because having an adequate sampling of actual piezo displacements, a "new" (virtual) piezo displacement can be interpolated mathematically [1]. For this reason, in the current thesis, only a set of at max 1000 images is obtained which correspond to 1000 displacements of the piezo. So, in order to have a better accuracy of just 1000 displacements, "new" displacement samples are interpolated.

CONTROLLING INPUT VOLTAGE (Controlling piezo displacement)

The whole procedure is controlled by a computer, which means that the start of the communication is digital while the end (piezo) is analog. Additionally, the computer can provide control signals at levels like for example 5 volts, while the piezo needs 0-100 volts. One obvious solution to this is firstly the use of a device that converts digital control signal from the computer to analog driving signal to piezo. This is called a *Digital-to-Analog Converter* (DAC). Secondly, the low-voltage analog signal will be amplified by a *high-voltage DC-to-DC amplifier.* In order to do n displacements, the piezo must receive n discrete voltage signals, thus making necessary the use of a DAC with a $log_2 n$ -bit capacity. Lastly, the sequence of the signals and the synchronization of them with the camera signals must be operated by an external microcontroller.

Arduino and DAC communication

The signals to be headed to the DAC integrated chip are generated in the Arduino board [26]. The latter is a programmable circuit board that contains a microcontroller chip and a set of input and output pins for communication with other devices. The microcontroller chip, which consists of one CPU (processor core) along with memory and programmable input/output peripherals, is the core of the board that coordinates and executes the necessary instructions that are loaded from the computer through an Integrated Development Environment (IDE). The IDE runs in the computer and converts code written in C/C++ to binary instructions that are stored in the registers of the microcontroller CPU.

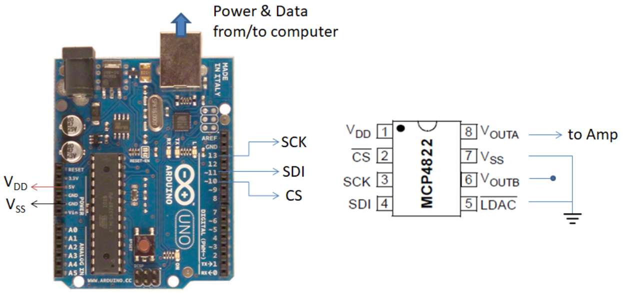


FIGURE A.3: SPI bus between Arduino and DAC.
fontsize1012 Source: Arduino [26], MCP [27]

Both the DAC and the Arduino board support a Serial Peripheral Interface (SPI) which allows their communication. For this specific task, the Arduino is defined

[1] meaning to be approximated by a known function the output of which traverses through all the real measured sample points with the least residual error

as the master of the communication and the DAC as the slave. SPI consists of 4 channels: (MOSI) Master Output Slave Input, for data output from master; (MISO) Master Input Slave Output, for data output from slave; (SS) Slave Select, output from master, selecting among many slaves if more than one; (SCK) Serial Clock, output from master, coordinating the communication with clock pulses. SPI delivers one byte per clock cycle. When prompted from the computer, the Arduino sends to the DAC via the MOSI channel, a binary command encoding the voltage that must be applied to the amplifier input. The MISO channel is not used of course as no information is sent from the DAC to the Arduino. Analytically, the correspondingly connected pins of the Arduino and the DAC are shown in Fig:A.3. The corresponding MOSI pin of the DAC is named SDI (Serial Data Input), while the SS pin is named CS (Chip Select).

In the current thesis, a 12-bit MCP 4822 integrated chip [27] is used as a DAC, allowing a total of 4096 output signals and specifically from 0-4095 mV (1 mV resolution). All the input pins of the device are connected to the Arduino microcontroller and the one output pin is connected to the high-voltage amplifier. Specifically, the DAC is supplied with power from the V_{DD} pin which is connected to the 5 V output pin of the Arduino. The input supply voltage is relative to the V_{SS} pin, which is connected to the GND (ground) pin of the Arduino, and can range from 2.7 V to 5.5 V. Independently of the SPI, the DAC has also a pin named LDAC. When this pin is tied to low (V_{SS}), both output pins are updated at the same time with their input register contents (where the encoded voltage command is stored).

Regarding the realisation of the wiring of the Arduino and the DAC, a perfboard[2] accomodating the DAC is plugged using conducting headers on top of the Arduino board (in the two I/O pin arrays of the board).

DAC to Amplifier to Piezo Actuator

The output signal of the DAC acts as the input for the DC-to-DC high voltage ampplifier. The amplifier is a power converter that steps up voltage while stepping down current from its input to its output. This is done in a way that the output power ($P = V \cdot I$) is conserved in a stable value. After the voltage signal is boosted it is applied to the piezo actuator.

Controlling the camera

The camera used in this thesis is the FLIR's Blackfly® S BFS-U3-13Y3M (Fig:A.4) [28]. It is a monochrome camera, thus resulting in a gray scale image. Each pixel from the 1.3 M in total is capable of measuring 1024 different levels of light intensity[3] (10-bit pixel depth). The pixel size is $4.8\mu m$x$4.8\mu m$.

Incident photons are converted to voltage via photodiodes in each pixel and are read out of them in manners specified by the hardware logic embedded on the whole image sensor along with software instructions that are stored in programmable chips. This is called a firmware. Firmware forms the very first low-level bridge between the physical and the hardware world. So, in order for someone to access the images stored in the camera, he or she can write a piece of software. But because it is not

[2] thin, rigid sheet with holes pre-drilled at standard intervals across a grid for prototyping electronic circuits

[3] actual saturation capacity of an individual sensor well is 6034 electrons, though the Analog-to-Digital converters of the whole sensor are only 10-bit in depth meaning 1024 different intensity levels

Figure A.4: Camera used in thesis.
Source: Flir [28]

always useful to build something from scratch, a set of subroutine definitions, communication protocols and software tools is provided by the manufacturing company of the camera. These definitions, protocols and tools make up the building blocks of the interface/bridge between hardware and software. For this reason, this is called an Application Programming Interface (API). For the shake of usefulness and ease from the perspective of the programmer, this API is bundled together with a set of software tools that can be used to create and develop applications based on the building blocks of the API. This extra software tools are called a Software Development Kit (SDK). Therefore to wrap up, when the programmer needs to do a specific task on the camera, he or she uses functions from the SDK that make calls to the API requesting data from the firmware of the camera.

For the Blackfly® S camera, the API and SDK (named as Spinnaker) are available in C++ and Python programming languages [29]. All of the code written in this thesis for the direct management of the camera is in Python. The Spinnaker SDK is accompanied with a Python library, named as PySpin, for user-friendly configuration and run of the camera. This library gathers a set of functions used for configuration of acquisition and trigger mode, image pixel format, image size, exposure time, gain, gamma value, binning and execution of the tasks of beginning and finishing the image acquisition.

Camera and Piezo Actuator Synchronization

As it was stated at the beginning of this section, the alternation of the Arduino → piezo flow and the computer ↔ camera flow is controlled automatically by an exchange of information between the computer and the Arduino. The corresponding timing diagram between each flow of information is shown in Fig:A.5.

To start with, for this specific synchronization task, Arduino functions as a Finite-State Machine (FSM). It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some external inputs. The Arduino FSM in this current situation consists of two basic states: one for provoking a piezo displacement (`STATE_MOVE_PIEZO`) and one for waiting until the computer acknowledges for the next piezo displacement (`STATE_WAIT_ACK`). There is also the state during which the Arduino waits to read from the serial buffer of intercommunication the exact number of steps that the piezo must do (`STATE_READ_STEPS`). This is also the initializing state, so when there is no transfer of information between the computer and the Arduino, the latter stays permanently there. The last two states are the `STATE_READ_INCR` where the Arduino receives the increment (or decrement in the current situation) of voltage that it must

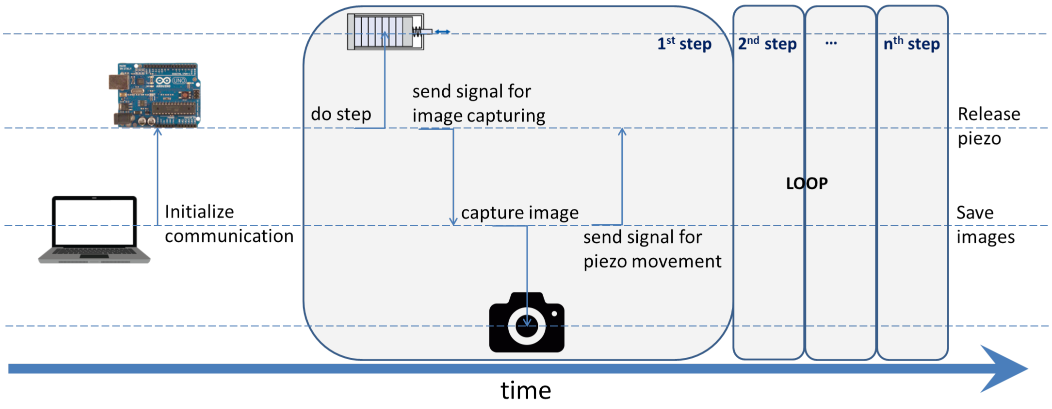


Figure A.5: Timing diagram of piezo actuator-Arduino-computer-camera interaction. In each loop, the piezo attains a single displacement and an image of the interferogram is captured and saved in memory.

apply to the DAC in each step and the `STATE_PREPARE` where the piezo is set on its maximum displacement, supplied with approximately $4.096 \cdot 20$ V.

ACQUISITION OF INTERFEROGRAMS

The procedure of acquiring a set of interferogram images is described below, using two code scripts that interact with one another. The script from the Arduino IDE is loaded and run first, to establish the connection between the Arduino and the computer. After, it will wait for commands from the computer in order to move the piezo. These commands will be realised with running the Python script that also commands the camera for image capturing. The person operating the device has to do only 3 tiny steps to begin the procedure:

a. Plug the Arduino board to the computer.

b. Upload the right script to the board.

c. When script is loaded, run the Python script.

Hereafter, exchange of information and commands happen automatically (see Fig. A.5 for flowchart):

1. Arduino waits for number of piezo steps.
2. Computer sends number of piezo steps —Arduino reads and then waits for voltage increment.
3. Computer sends voltage increment and waits for acknowledgement that the Arduino has transmitted the movement signal —Arduino reads and sets the piezo to its maximum displacement.
4. Arduino decreases the piezo displacement and sends acknowledgement signal to the computer.

5. Arduino waits for acknowledgement for the next piezo movement and calculates the next applied voltage —Computer triggers image acquisition and saves image in an image stack in memory.

6. Computer sends acknowledgement signal to the Arduino for the next piezo movement and waits for acknowledgement for the next image capture.

7. Steps 4–6 repeated until required number of interferograms is reached.

8. Computer saves image stack from memory to disk —Arduino sets the piezo free of any applied voltage and reinstates itself to the initialization state waiting to read number of piezo steps in order to begin a new cycle of interferogram capturing.

Appendix B

Computer-Arduino intercommunication

B.1 Python script: Computer transmits/receives to/from Camera and Arduino

For starters, a group of necessary packages/libraries are imported in order for the program to run:

- package `'serial'` : uses the USB connection with the serial protocol
- library `'camera_library' (clb)` : customized implementation of the Spinnaker API
- library `'rw_config' (cfg)` : reads and writes to a special file ('config.ini') which holds camera configurations for different occasions
- package `'imageio'` : used for saving and loading of image files
- package `'numpy'` : used for depicting images as 2D arrays

The main Python file, that is the computer file that the user runs and is used as the authority in this specific intercommunication task, is named `'sync_PiezoCam.py'`.

The **basic executed operations** are the following:

```python
"""
Initialize/Establish connection between camera and computer.
Nodes and settings (image height,width) for camera manipulation are pulled from the camera API implementation.
"""
cam_nodes, cam_settings = init_camera()

"""
Establish serial connection on port COM4 with baud rate 9600.
"""
ser = serial.Serial('COM4', 9600)

"""
Set the number of piezo steps and increment of DAC output voltage
"""
max_steps = 4096 # minimum step: 1 mV, maximum step: 4096 mV (ref. DAC output)
steps = 600
increment = round(max_steps/steps)

"""
Store bytes of information to the serial buffer. Transfer from buffer to Arduino (flush) after each distinct information.
```

```python
"""
ser.write(encode_int2fixednumbytes(steps)) # Send number of steps to arduino
ser.flush()
ser.write(encode_int2fixednumbytes(increment)) # Send voltage increment to arduino
ser.flush()

"""
Create a stack of 2D arrays in which images will be loaded to later.
"""
image_stack = np.zeros([height, rows, cols])

"""
Start the image acquisition engine.
"""
clb.begin_acquisition(cam)

"""
The basic loop of communication.
"""
# extra : rounding of increment may lead to extra steps
for i in range(steps-extra):
  # Read the value of voltage that the Arduino just sent to the DAC.
  # (Wait until the buffer is loaded and flushed by the Arduino)
    rvalue = ser.readline().split(b'\r')[0]

    # omit : a few first interferograms are not captured due to DC-DC amp irregularities when starting.
    if i+1 > omit:
      # Software triggering to the camera to acquire an image.
        image_stack[i-omit-1] = clb.grab_next_image_by_trigger(cam)

    # Send acknowldge to the Arduino to continue with the next piezo displacement.
    ser.write(b'ok\n')
    ser.flush()
```

B.2 Arduino C/C++ script: Arduino transmits/receives to/from Computer

Arduino is in constant communication with both the computer and the DAC device (MCP4822). Therefore it makes use of the built-in `'Serial'` function for Computer-Arduino intercom and an `SPI` library for Arduino-DAC intercom along with a custom library named `MCPDAC` for easy manipulation of the DAC pins and internal registers.

The main Arduino file that the user runs is named `'control_piezo.ino'`. An Arduino file generally consists of two basic functions for starters, named *'setup'* and *loop*. The former is executed once and in the current task it opens the serial connection from the side of Arduino and initializes the DAC device. The latter is the implementation of the FSM (look: A). More specifically, in each loop, a state-specific branch of code is run eternally until the Arduino disconnects from the computer by unplugging the USB cable or by uploading an idle code.

All the executed operations are the following:

```cpp
// MCPDAC relies on SPI.
#include <SPI.h>
#include "MCPDAC.h"

```

```
String incr;
String steps;
unsigned int num_steps;
unsigned int volt_incr;
int ctr;

typedef enum e_State {
  STATE_READ_STEPS, // wait for number of steps from the serial
  STATE_READ_INCR, // wait for voltage increment from the serial
  STATE_PREPARE,
  STATE_MOVE_PZT,   // moving PZT
  STATE_WAIT_ACK    // waiting acknowledge from message
} eState;

void setup()
{
  Serial.begin(9600);

  // CS on pin 10, no LDAC pin (tie it to ground).
  MCPDAC.begin(10);

  // Set the gain to "HIGH" mode - 0 to 4096mV.
  MCPDAC.setGain(CHANNEL_A,GAIN_HIGH);

  // Do not shut down channel A, but shut down channel B.
  MCPDAC.shutdown(CHANNEL_A,false);
  MCPDAC.shutdown(CHANNEL_B,true);
}

void loop()
{
  static unsigned int volts = 4096; # initialized voltage of DAC output; piezo on max displacement
  static int max_strlen = 5; # number of steps and voltage increment info received from computer is exactly 5 bytes
  static int ascend = 0; # boolean, piezo movement direction; ascent --> increasing voltage of DAC output
  static int stp = 0;
  static eState LoopState = STATE_READ_STEPS; // stores the current state; initialized with state_read_steps; static variables are never overwritten
  eState NextState = LoopState; # initialize next state variable
  static char a = 'a'; static char d = 'd'; char s; # for serial printing

  switch (LoopState) {
    case STATE_READ_STEPS:
      // wait until the receiver buffer of Arduino has exactly 5 bytes of info
      if (Serial.available() == max_strlen) {
        // get the number of steps from the serial
        steps = Serial.readStringUntil('\n');
        num_steps = decode_string2int(steps);
        NextState = STATE_READ_INCR;
        break;
      }else{
        NextState = STATE_READ_STEPS;
        break;
      }
    case STATE_READ_INCR:
      if (Serial.available() == max_strlen) {
        ctr = 0;
        // get the voltage increment from the serial
      incr = Serial.readStringUntil('\n');
      volt_incr = decode_string2int(incr);
      NextState = STATE_PREPARE;
      break;
    }else{
      NextState = STATE_READ_INCR;
      break;
    }
  case STATE_PREPARE:
    volts = 4096;
    MCPDAC.setVoltage(CHANNEL_A, volts&0x0fff);
    // wait a little until the piezo reaches its max displacement
    delay(5000);
    NextState = STATE_MOVE_PZT;
    break;
  case STATE_MOVE_PZT:
    ctr++;
    // Set the voltage of channel A. Voltage integer is stored in a
specific register of the DAC, linked by the hexadecimal address
    MCPDAC.setVoltage(CHANNEL_A, volts&0x0fff);
    delay(10);
    // construct ACK (with piezo movement direction and DAC voltage
info) to be sent to computer
    s = (ascend == 1) ? a : d ;
    Serial.print(s);
    Serial.print("_");
    Serial.println(volts); // send signal to python
    Serial.flush(); // wait for data to be transmitted
    NextState = STATE_WAIT_ACK;
    break;
  case STATE_WAIT_ACK:
  // ACK by computer is exactly 3 bytes
    if (Serial.available() == 3 && (Serial.readStringUntil('\n')).
equals("ok")) {
      // no state change while no acknowledge
      unsigned int volt_plus = volts + volt_incr;
      unsigned int volt_minus = volts - volt_incr;

      // prepare next DAC voltage, depending on current piezo movement
 direction
      if (ascend == 1){
        if ( (volt_plus >=0) && (volt_plus <= 4096) ){
          volts += volt_incr;
        }else{
          ascend = 0;
          volts -= volt_incr;
        }
      }else{
        if ( (volt_minus >=0) && (volt_minus <= 4096) ){
          volts -= volt_incr;
        }else{
          ascend = 1;
          volts += volt_incr;
        }
      }
      NextState = STATE_MOVE_PZT;
      break;
    // only when done with stepping, relax piezo
    }else if (num_steps+1 == ctr) {
      volts = 0;
      MCPDAC.setVoltage(CHANNEL_A, volts&0x0fff);
      NextState = STATE_READ_STEPS;
      break;
      }else{
        NextState = STATE_WAIT_ACK;
        break;
      }
  }

  // define the state for the next loop
  LoopState = NextState;
}

// discard possible zeros from the input string and convert the string to integer
int decode_string2int(String str)
{
  while (str[0] == '0') {
    str.remove(0,1);
  }
  return str.toInt();
}
```

Appendix C

Interferogram and Profile analysis

C.1 Python script: Profile calculation for CSI

The extraction of a surface profile is done following the method of CSI. The whole process is discussed in Sec. 3.2. The **basic operations** are the following:

```python
"""
Load interferogram files and create a stack
"""
"""
...
"""
"""
Load calibration file
"""
"""
...
"""

def gaus(x,a,x0,sigma,c):
    return a*np.exp(-(x-x0)**2/(2*sigma**2))+c

"""
PER PIXEL ANALYSIS
"""
for i in range(rows):
        for j in range(cols):
        # extract interferogram in vertical axis (normal to the surface)
            z = image_stack[:,i,j]
            zm = z - np.median(z)
            za = np.abs(zm)
        # take the "envelope"; sigma is selected as a good trade-off with visual criteria.
            zg = gaussian_filter1d(za, sigma=30)

        # make a first guess for the gaussian fitting of the "envelope"
            nmax = np.argmax(zg)
            amp = zg[nmax]

            try:
            # statistically 60 pixels is a good guess for the specific white light source used
                sigma = 60
                c = np.median(zg)
                guess = [amp,nmax,sigma,c]
                popt, _ = curve_fit(gaus,n,zg,p0=guess)
                peak = popt[1]
            # the surface elevation is the interpolated value of the calibration array with input
            # the calculated peak of the best-fit.
                profile[i,j] = curve(peak)
            except:
                profile[i,j] = None

"""
Save profile
"""
"""
...
"""
```

C.2 Python script: Profile calculation for SRWLI

This process is presented in Sec. E. The **basic operations** are the following:

```

"""
Load images of interference, reference, sample and background.
Load array containing the calibrated values for the decomposed wavelengths.
"""

# compute wavenumber array
k = 2*np.pi/lamda

# smooth the following signals before using them to isolate the interference cosine term
Is_g = gaussian_filter1d(Is, sigma=3, axis=1)
Ir_g = gaussian_filter1d(Ir, sigma=3, axis=1)
Ib_g = gaussian_filter1d(Ib, sigma=3, axis=1)

# calculate interference cosine term --> 'Ipr'
eps = np.finfo(np.float64).eps
nom = 2.0*np.sqrt(Is_g)*np.sqrt(Ir_g)
nom[np.where(nom==0)] = eps
Ipr = (Itot-Is_g-Ir_g-Ib_g)/nom

# EXTRA PRE-PROCESSING
# extract waviness --> 'Iprmodg2'
Ipr_g = gaussian_filter1d(Ipr, sigma=11, axis=1)
Ipr_mod = Ipr - Ipr_g
Ipr_modg = gaussian_filter1d(Ipr_mod, sigma=3, axis=1)
Iprmodg2 = Ipr_modg - np.median(Ipr_modg)

# Linear resampling (factor=1) of the wavenumber samples
num_resample = cols
if num_resample % 2 == 0:
    num_resample -= 1
k_new = np.linspace(k[0], k[-1], num_resample)

# calculate frequency range and samples
dk = abs(k_new[0] - k_new[1])
Fn = 1/(2*dk)
Nfreqs_f = num_resample
f_axis = np.linspace(0, Fn, Nfreqs_f//2+1)

# initialize profile array
z = np.zeros(rows)
```

```

"""
Iterate procedure for each row of the image
"""
for i in range(rows):
  # resampled waviness signal
    f_rsmpl = interp1d(k, Iprmodg2[i], kind='cubic')
    Ipr_new = f_rsmpl(k_new)

    # FT calculation
    fourier = np.fft.fft(Ipr_new, n=Nfreqs_f)
    fourier = np.abs(fourier)
    fourier = np.fft.fftshift(fourier)
    fourier = fourier[Nfreqs_f//2:]

    # Upsampling of fourier signal to "increase" resolution
    Nf_times = 20
    f_axisnew = np.linspace(f_axis[0], f_axis[-1], Nfreqs_f*Nf_times)
    f_rsmpl2 = interp1d(f_axis, fourier, kind='cubic')
    fourier_new = f_rsmpl2(f_axisnew)

    # smooth fourier signal to eradicate noise for peak detection
    sigma_f = Nf_times*1      #Nf_times/4
    fourier_g = gaussian_filter(fourier_new, sigma=sigma_f)
    argmax_idx = np.argmax(fourier_g)
    # calculate surface elevation
    z[i] = np.pi*f_axisnew[argmax_idx]/um
```

C.3 Python script: Piezo actuator calibration

This process is explained thoroughly in Sec. 3.6. The **basic operations** are the following:

```
"""
Load interferogram files and create a stack
"""
"""
...
"""

"""
PHASE CALCULATION FROM EACH INTERFEROGRAM
"""
phases[0] = calc_phase(image)

for j in range(1,listlen):
    image = ((io.imread(file_list[j])).astype(float))
    # odd dimensions of image --> Frequency 0 of Fourier axes is exactly on the center of the image
    image = im2odd_dim(image)
    phases[j] = calc_phase(image)

"""
CALCULATE DISPLACEMENT FROM PHASE SHIFTS
"""
uwphases = np.unwrap(phases)
dp = np.diff(uwphases)

# Phase difference converted to displacement of fringes.
# In michelson interferometers, displacement of the sample is equal to
# half the optical path difference
```

```
lamda = float(input('Enter the peak wavelength of the source (nm)...\n')) # in nm
displacements = dp/(2*np.pi)*lamda/2 # in nm
displacements = np.concatenate([[0],displacements]) # zero as the first position

incremental = np.cumsum(displacements)
grad = np.gradient(incremental)
grad = reject_outliers(grad, m = 2.)
dp_mean = np.mean(grad)

incremental = - incremental if dp_mean < 0 else incremental

"""
Save Calibration file
"""
...
"""
```

C.4 Python script: Profile post-processing

This process is explained in Sec. 3.3. It requires user visual feedback and for this reason a small GUI has been built. User events signal callback functions that do the basic operations of the GUI. The **callback functions** are the following:

```
"""
Create Matplotlib window with widgets for buttons, figures, axes, span selectors and ROI selectors.
"""
"""
...
"""

def line_select_callback(eclick, erelease):
    global ROI

    if toggle_selector_RS.active:
        x1, y1 = int(round(eclick.xdata)), int(round(eclick.ydata))
        x2, y2 = int(round(erelease.xdata)), int(round(erelease.ydata))
        ROI = [(x1,y1), (x2,y2)]

def enable_ROIsel(event):
    global ts_id

    del ROI[:]

    toggle_selector_RS.set_active(True)
    toggle_selector_RS.set_visible(True)
    ts_id = fig.canvas.mpl_connect('key_press_event', toggle_selector)

def toggle_selector(event):
    if event.key=='enter' and toggle_selector_RS.active:
#         print(' RectangleSelector deactivated.')
        textbox.set_val('ROI selected')
        toggle_selector_RS.set_visible(False)
        toggle_selector_RS.update()
        toggle_selector_RS.set_active(False)
        fig.canvas.mpl_disconnect(ts_id)

def onselect(xmin, xmax):
    global flag_save, thres, unrotated
```

```python

    thres[thres<xmin] = xmin
    thres[thres>xmax] = xmax
    ax1.cla()
    ax1.hist(thres.ravel(), bins=255)
    ax1.set_title('Press left mouse button and drag to select the wanted histogram range')
    ax2.imshow(thres, cmap=plt.cm.viridis)
    ax2.set_title('Thresholded profile')
    unrotated = 'thres'
    textbox.set_val('Low-high thresholds set as [%d,%d]'%(xmin,xmax))
    cbar2.set_clim(vmin=np.amin(thres),vmax=np.amax(thres))
    cbar2.draw_all()
    flag_save = 1

def crop(event):
    global thres, flag_save

    (x1,y1), (x2,y2) = ROI
    thres = thres[y1:y2,x1:x2]

    ax1.cla()
    ax1.hist(thres.ravel(), bins=255)
    ax1.set_title('Press left mouse button and drag to select the wanted histogram range')
    ax2.imshow(thres)
    ax2.set_title('Cropped profile')
    cbar2.set_clim(vmin=np.amin(thres),vmax=np.amax(thres))
    cbar2.draw_all()
    textbox.set_val('Image cropped')
    flag_save = 3

def rotate_image(event):
    global flat_profile, flag_save, unrotated

    (x1,y1), (x2,y2) = ROI
    if unrotated == 'thres':
        flat_profile = rotate(thres, [y1, y2], [x1, x2])
        im = ax1.imshow(thres)
        ax1.axis('off')
        ax1.set_title('Original/Thresholded')
        cbar1 = fig.colorbar(im, ax=ax1)
        cbar1.set_clim(vmin=np.amin(thres),vmax=np.amax(thres))
        cbar1.draw_all()
    elif unrotated == 'raw':
        flat_profile = rotate(prof, [y1, y2], [x1, x2])
        im = ax1.imshow(prof)
        ax1.axis('off')
        ax1.set_title('Original/Raw')
        cbar1 = fig.colorbar(im, ax=ax1)
        cbar1.set_clim(vmin=np.amin(prof),vmax=np.amax(prof))
        cbar1.draw_all()
    ax2.imshow(flat_profile)
    ax2.set_title('Rotated/Horizontalized')
    cbar2.set_clim(vmin=np.amin(flat_profile),vmax=np.amax(flat_profile))
    cbar2.draw_all()
    textbox.set_val('Image rotated')
    flag_save = 2

def compute_parameters(event):
    global fig2

    (x1,y1), (x2,y2) = ROI
    area = flat_profile[y1:y2,x1:x2]
#    area = thres[y1:y2,x1:x2]
#    area = prof[y1:y2,x1:x2]

    # Mean plane
    M = np.mean(area)

    # Distances of the sufrace from the mean plane
    abs_diff = np.abs(area - M)

#    # Flatness deviation
#    flt =

    # Maximum peak height of the surface
    Sp = np.amax(area-M)

    # Maximum valley depth of the surface
    Sv = np.amax(M-area)

    # Maximum height of the surface
    Sz = Sp + Sv

    # Average roughness ( Mean difference around mean plane )
    Sa = np.mean(abs_diff)

    # Roughness (Standard deviation of difference(smoothed surface, surface))
    filt = gaussian_filter(area, 11)
    diff = filt - area
    Sr = np.std(diff)

    # Root mean square roughness
    Sq = np.sqrt(np.mean(abs_diff**2))

    # Skewness (Degree of symmetry of the surface heights around the mean plane)
    Ssk = 1/Sq**3*(np.mean(abs_diff**3))

    # Kurtosis (Presence of lack of inordinately high peaks or deep valleys)
    Sku = 1/Sq**4*np.mean(abs_diff**4)

    col_labels = ['Surface\nParameter','Value','Unit','Description']
    table_data = np.array([['M','%.0f'%M,'nm','Mean plane'], ['Sp','%.0f'%Sp,'nm','Maximum peak height of the surface'],
                           ['Sv','%.0f'%Sv,'nm','Maximum valley depth of the surface'], ['Sz','%.0f'%Sz,'nm','Maximum height of the surface'],
                           ['Sa','%.0f'%Sa,'nm','Roughness: Mean difference around mean plane'],
                           ['Sr','%.0f'%Sr,'nm','Roughness: Standard deviation around surface waviness'],
                           ['Sq','%.0f'%Sq,'nm','Roughness: Root mean square difference around mean plane'],
                           ['Ssk','%.4f'%Ssk,'-','Skewness: Degree of symmetry of the surface heights around the mean plane.\n'+
                                         'Ssk>0: peaks predominance, Ssk<0: valleys predominance'],
                           ['Sku','%.4f'%Sku,'-','Kurtosis: Presence (Sku>3) of lack (Sku<3) of '+
                                         'inordinately high peaks or deep valleys']])

    # MATPLOTLIB TABLE (NOT SELECTABLE TEXT)

    fig2, ax3 =plt.subplots(1,1)
    figManager = plt.get_current_fig_manager()
    figManager.window.showMaximized()
    fig2.canvas.set_window_title('Areal Height Parameters')
    ax3.axis('tight')
    ax3.axis('off')
    tparam = ax3.table(cellText=table_data,colLabels=col_labels,colWidths=[0.1,0.1,0.1,0.75],
                       cellLoc='center',colLoc='center',loc='center')
    tparam.auto_set_font_size(False)
    tparam.set_fontsize(12)
    tparam.scale(1,1.6)

def average_pplot(event):
    (x1,y1), (x2,y2) = ROI
    area = flat_profile[y1:y2,x1:x2]

    ax1.cla()
    if direction == 'horiz profile':
        avg = np.average(area, axis=0)
        ax1.plot(avg)
        ax1.set_title('Average horizontal profile of ROI')
    else:
        avg = np.average(area, axis=1)
        ax1.plot(avg)
        ax1.set_title('Average vertical profile of ROI')
```

C.5 Python script: Radius of Curvature assessment (Curved profile)

This process is explained in Sec. 3.5. The **basic operations** are the following:

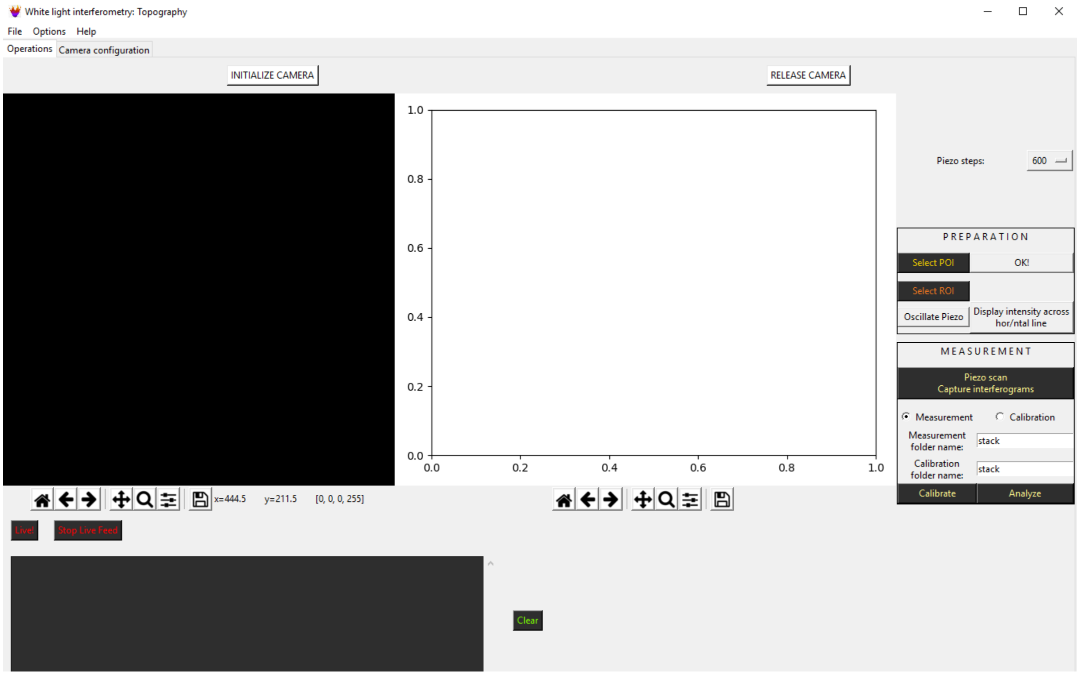
```
"""
Load profile image
"""
"""
...
"""

# values in um
# calibrated pixel size range
px_range = [9.8, 10.2]
px_mean = 10.0
px_uc = 0.2
sigma_z = 0.028

sel = '0'

while sel != '1' and sel != '2':
    print('NOMINAL AND MEASURED RADIUS OF CURVATURE COMPARISON\n\n')
    print('Enter \'1\' if nominal ROC is known...\n')
    print('Enter \'2\' if refractive index and focal distance of lens are known...\n')

    sel = input()

if sel == '1':
    ROC = float(input('Enter ROC (mm)\n'))
    f = float(input('Enter nominal focal distance (mm)\n'))
elif sel == '2':
    n_lens = float(input('Enter nominal refractive index of the lens...\n'))
    f = float(input('Enter nominal focal distance (mm)\n'))
    ROC = abs(f*(n_lens-1.0))

"""
BEST-FIT CALCULATION OF 1) 2D PROFILE WITH A PARABOLOID
             2) 1D PROFILE PASSING THROUGH CENTER WITH A PARABOLA
FOR UPPER AND LOWER LIMIT OF PIXEL SIZE
"""

for i in range(2):
    px = px_range[i]

    fringearea_gf = gaussian_filter(fringearea, sigma=1)

    if f < 0:
        [r0], [c0] = np.where(fringearea_gf == np.amin(fringearea_gf))
        Z = fringearea - fringearea[r0,c0]
        radius_c = ROC*1000
    elif f > 0:
        [r0], [c0] = np.where(fringearea_gf == np.amax(fringearea_gf))
        Z = fringearea - fringearea[r0,c0]
        radius_c = - ROC*1000


    X = np.arange(c)*px
    Y = np.arange(r)*px
    X, Y = np.meshgrid(X, Y)

    guess = [radius_c, px*r0, px*c0, 0.0]

    Z = Z/1000 # now in um

    po, pc = curve_fit(paraboloid, (X,Y), Z.ravel(), p0=guess)
    paraboloid_fit = paraboloid((X,Y), *po).reshape(np.shape(Z))
    perr = np.sqrt(np.diag(pc))

    po_list.append(po)
    pc_list.append(pc)
    perr_list.append(perr)

    if f < 0:
        [r0], [c0] = np.where(fringearea_gf == np.amin(fringearea_gf))
        z = fringearea[r0]-np.amin(fringearea[r0])
        radius_c = ROC*1000
    elif f > 0:
        [r0], [c0] = np.where(fringearea_gf == np.amax(fringearea_gf))
        z = fringearea[r0]-np.amax(fringearea[r0])
        radius_c = - ROC*1000

    guess2 = [radius_c, px*c0]

    z = z/1000 # now in um

    x = np.arange(c)*px
    s_z = sigma_z*np.ones(c)
    po2, pc2 = curve_fit(parabola, x, z, p0=guess2, sigma=s_z , absolute_sigma=False)
    perr2 = np.sqrt(np.diag(pc2))

```

```
        po2_list.append(po2)
        pc2_list.append(pc2)
        perr2_list.append(perr2)

    """
    STATISTICS
    """

        residuals = Z.ravel() - paraboloid((X,Y), *po)
        ss_res = np.sum(residuals**2)
        ss_tot = np.sum((Z.ravel()-np.mean(Z.ravel()))**2)
        r_squared = 1 - (ss_res / ss_tot)

        r_squared_list.append(r_squared)

"""
RESULTS: ROC AND UNCERTAINTY IN MEASUREMENT
"""
# values in mm
error_pxmin = perr_list[0][0]/1000
error_pxmax = perr_list[1][0]/1000
ROCmeas_pxmin = abs(po_list[0][0])/1000
ROCmeas_pxmax = abs(po_list[1][0])/1000
minROC = ROCmeas_pxmin - error_pxmin
maxROC = ROCmeas_pxmax + error_pxmax
ROC_meas = (maxROC + minROC)/2
ROC_error = (maxROC - minROC)/2
```

Appendix D

Integrated Graphical User Interface (GUI)

For the shake of operational simplicity of the whole procedure from the interferogram capturing to the profile analysis, a minimalistic GUI was developed (Fig:D.1, D.2). The requirements of the preparation for measurement are such that many procedures has to be done in parallel. For this reason, the code of this GUI was built strongly based on the "multiprocessing" package of Python. To run the GUI, simply run the "WLtopo_mp.py" script from the Windows command prompt (some IDEs may be problematic with running multi-processing modules in full functionality).



Piezo steps: 600

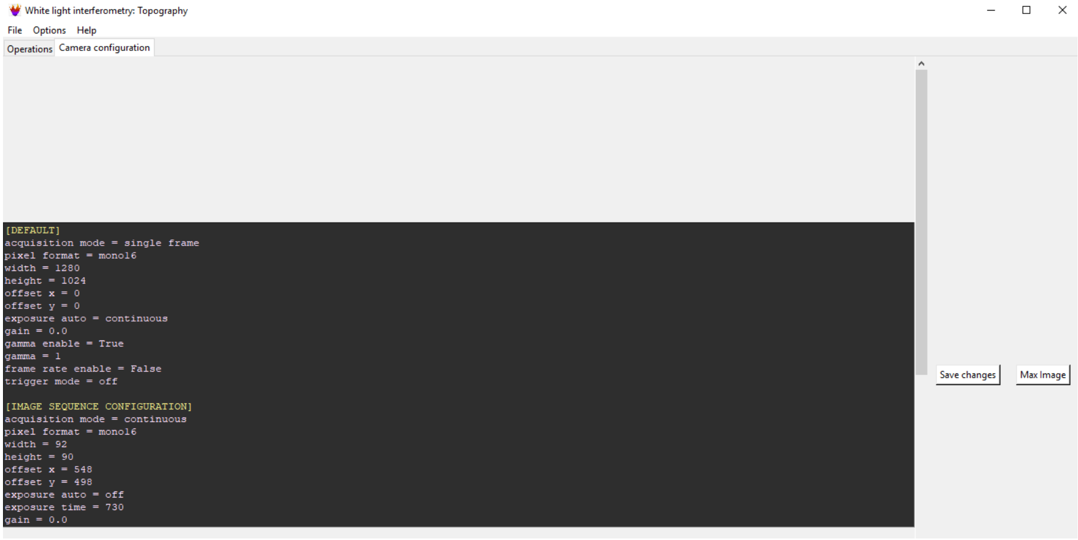
● Measurement ○ Calibration

Measurement folder name: stack

Calibration folder name: stack

x=444.5 y=211.5 [0, 0, 0, 255]

Figure D.1: GUI main window: operations and graph planes for results.



```
[DEFAULT]
acquisition mode = single frame
pixel format = mono16
width = 1280
height = 1024
offset x = 0
offset y = 0
exposure auto = continuous
gain = 0.0
gamma enable = True
gamma = 1
frame rate enable = False
trigger mode = off

[IMAGE SEQUENCE CONFIGURATION]
acquisition mode = continuous
pixel format = mono16
width = 92
height = 90
offset x = 548
offset y = 498
exposure auto = off
exposure time = 730
gain = 0.0
```

Figure D.2: GUI camera configuration window: all basic options needed for the camera to run.

–FUNCTIONS–

CAMERA CONFIGURATION TAB

Before taking a measurement, many things must be done in order, so as to achieve the best possible measurement. For starters, the user connects the camera, initializes it and configures it according to the "config.ini" file. A copy of it is stored in an editable text box in the "camera confiration tab". The user can change every single camera option of each section. There are three sections: "*Default*", which uses the maximum possible image size, "*Image Sequence Configuration*", which uses 16-bit pixel depth and internal software trigger for capturing each image and "*Live View Configuration*", which does not use any trigger but has 8-bit pixel depth for displaying purposes. Any other camera option is irrelevant to the occasion used and freely changes by just overwriting and saving the corresponding values.

LIVE VIEW

On the left of the "Operations" tab, the "Live View" window displays continuously the selected region of interest of the sample. Press "Live!" and "Stop Live feed" to start and stop respectively the live feed.

PREPERATION FRAME

Select ROI

All operations here have the purpose of setting up the sample holder of the interferometer in the best possible position in the x,y,z axes, with the most important being the axis parallel to the optical one.

- Click on the "Select ROI" button.
- Click on the "Live View" WIndow on the wanted top left corner of the ROI and release on the bottom right one. A box selection has been made.
- Modify the selection by clicking and releasing around the box.
- When ready, press enter.

Display intensity across horizontal line

In some cases when seeking interference fringes, it might be useful to have an auxiliary live plot of just a line across the image to search for deviations in the constant light intensity that may indicate the existence of fringes. In that case:

- Press "Display intensity across horizontal line". Light intensity vs horizontal image axis plot is displayed on the right window/plane in synchronization with the live view feed on the left window.
- Press the same button to stop the plotting.

Oscillate Piezo

WLSI method requires the existence of a well shaped interference pattern per sample point (signal per pixel) from which the argmax will be easily determined, inferring thus the height of this sample point. So to avoid marginal situations where the argmax is located very close to the beginning or the end of the "per pixel signal", an auxiliary function can be used.

The user can select multiple Points of Interest (POI):

- Click on the "Select POI" button.
- Click on the "Live View" window on the wanted POI. Press enter.
- If more points of interest are needed, select each one and press enter.
- When every POI is selected, to finish the process click on the "OK" button next to the "Select POI" button.
- Press "Live!" to start the live feed.
- Press "Oscillate Piezo" to start the back and forth oscillation of the piezo. In each movement of the piezo,

MEASUREMENT FRAME

Piezo scan – Capture interferograms

Following preperation, for the WLSI measurement and calibration of the piezo movements, a piezo scanning must be done for each case:

- Type a measurement/calibration folder name (which will contain the interferograms) on the corresponding text field.
- Select "Measurement" or "Calibration" from the radio buttons.
- Click on "Piezo scan – Capture interferograms" button to execute the image acquisition. [Note: The calibration process requires the proper use of the optical arrangement (Coherent light source, optical flat/mirror in reference arm)]

Calibrate

Before the first measurement, a calibration must be done regarding the piezo movements, that is the steps and real distance made correspondence. [Note: The calibration process requires the proper use of the optical arrangement (Coherent light source, optical flat/mirror in sample arm)]. After the aqcuisition of the images as described above, the user will:

- Click on the "Calibrate" button. When the process is done, a text file containing the piezo displacements between each scan will be saved on the root folder with name "Mapping_Steps_Displacement_<piezo steps>_<file increment number>.txt"

Analyze

The final step to the whole process for obtaining a 2D profile of a specimen is the analysis of the measurement interferograms:

- Click on the "Analyze" button.
- Select the interferograms image set from the emerging prompt window.
- When the profile is calculated, it will be saved on the root folder with name "raw_profile_<interferograms image set folder name>" in text and tiff format.
- A new window will emerge for *post processing* of the profile. These functions are *image cropping*, *histogram thresholding*, *average line profiling* and *rotation of the profile plane parallel to the ground.*

–PARALLELIZATION–

The workload specifically for live continuous image display and synchronized feature plotting is beyond the capabilities of usual single-threaded script/modules. In this way, each new operation starts only when the previous in order has finished. To overcome this, a multi-processing/multi-threading approach was implemented which allows the parallel execution of at least two different operations.

Two extra processes are created when running the GUI script "WLtopo_mp.py". More specifically, there exist a process for *GUI design and display* which actually creates a child process for *camera manipulation* purposes and another for *piezo oscillation* purposes. In order to have synchronized camera displaying, piezo movement and feature plotting, these three processes must communicate with each other. Every time some data are needed to be plotted either on "Live View" window or in the adjecent plane (feature plotting), the camera manipulation process transfer these data to a shared queue. Then, the GUI process reads the data from the queue and displays/plots them accordingly. The whole operation happens in a datum per datum manner to keep a synchronized data "production" and "consuption".

Appendix E

SRWLI profile analysis - FT method

Spectrally-resolved white-light interferometry (SRWLI) [30, 31, 32] produces a 1D+1 surface elevation map of the sample under study. Eq. 2.18 shows that the interference pattern can change in a sinusoidal fashion either by changing the OPD or by varying the wavelength of the interfering beams. The latter is utilized by SRWLI. It is a technique that analyses the spectral modulation that results from the interference between a sample and a reference wavefront when an interferometric device is illuminated with a continuous, broad-spectrum light source. The spectral interferogram, which is formed at the exit plane of a spectrometer, encodes the phase as a function of wavenumber. If a reflective sample replaces one of the mirrors of the interferometer, the phase in the spectral domain contains information on its profile. The sample is imaged in a narrow slit so as to isolate a single line of the profile to be measured. This slit acts as the entrance slit of the spectrometer. The interfering beams after the slit are spectrally analysed using a diffraction grating in an imaging configuration.

The spectral decomposition of each point of the single sample line is then recorded across the horizontal arrays (rows) of the camera sensor. This is done for each sample point, thus eventually ending up to a 2D spectral interferogram. The resulted interferogram is a function of intensity distribution versus individual wavelengths on the horizontal axis and sample points on the vertical one (columns). Since the wavelength term in the interference equation represents the period of the sinusoidal function, it holds that the smaller the wavelength the larger the spatial frequency of the fringes in that region.

For each row, the interference equation holds as each row is the intensity distribution versus wavelengths. The unknown OPD can be calculated in various ways. If each interferogram row is transformed linearly as intensity distribution versus wavenumber, the OPD acts as the slope of the phase of the interference term ($phase = OPD \cdot k$, where k the independent variable), then it is clear that the unknown OPD term is related to the slope of the phase term. Then, by phase retrieving techniques the OPD can be deducted and thus the surface elevation.

A second technique is to calculate the dominant carrier frequency of the interference signal as it encompasses the unknown OPD in the following manner:

A typical cosine funtion is written in terms of frequency and the running independent variable (time or space). By comparing the frequency terms of the typical cosine and the interference cosine functions it can be shown that $OPD = 2 \cdot h = \pi f$, where h is the unknown surface height for each sample point. The dominant carrier frequency is found via the Fourier Transform technique. This specific algorithm was developed in this thesis as an act of collaboration for the thesis of the until recently fellow master student Matina Vlachou [31].

1. The measured interferogram I_{tot} is a function of intensity versus wavelengths. In order to calculate the dominant carrier frequency of it, the interferogram is transformed in a function of intensity versus wavenumbers.

2. To isolate the cosine term of the interference equation, four measurements have to be made: The captured interferogram I_{tot}, the individual captures of the two beams I_1 and I_2 and also a background capture with no illumination I_b for noise eradication. The signal under examination thus becomes the expression: $\frac{I_{tot}-I_1-I_2-I_b}{2\sqrt{I_1 I_2}}$.

3. As an extra pre-processing step and in order to clean up the wanted cosine term, the waviness of the signal is extracted so as to better isolate the dominant carrier frequency.

4. *From this point on, the operations are done iteratively for each row of the 2D image.* The FT algorithm requires equispaced samples as an input, so the linearity of the wavenumber distribution is ensured by linear resampling of the wavenumber samples. A new resampled waviness signal is calculated per row.

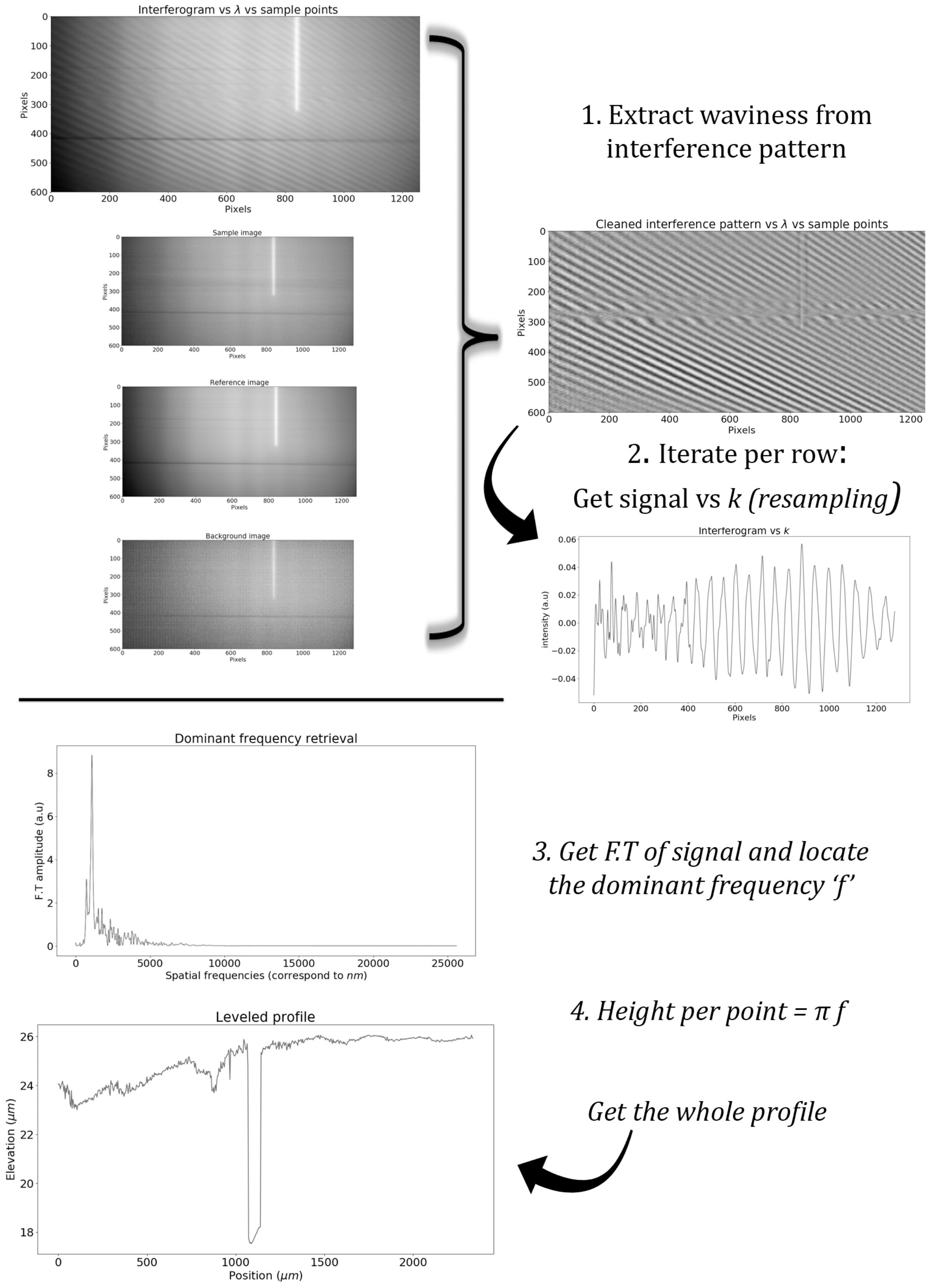
5. An FT is computed for the row signal.

6. To further increase the resolution in the spatial frequency domain, fourier signal is upsampled by a factor of 20.

7. The peak of the fourier and the corresponded spatial frequency signal are detected.

8. The unknown surface height of the single sample point is then calculated as: $h = f \cdot \pi$

The basic operations of the Python script are located to the Appendix E and the whole procedure is visualized in the following figure (Fig. E.1). Results of this method are presented in [31].



1. Extract waviness from interference pattern

2. Iterate per row:
Get signal vs *k (resampling)*

3. Get F.T of signal and locate the dominant frequency ‘f’

4. Height per point = π f

Get the whole profile

Figure E.1: SRWLI profile analysis - FT method. Sample used: sample engraved with a stylus tip (9.6μm nominal height)

Appendix F

Estimation of nominal value of RoC of front surface of CLs

In the prescription details of the CLs, the RoC of the front surface (FC) is not given. For this reason, it must be calculated from other quantities. In particular, usually the quantities given by the manufacturer of the CLs are the back vertex power (BVP), the RoC of the back surface (BC), the central thickness, the refractive index (RI) and of course the diameter.

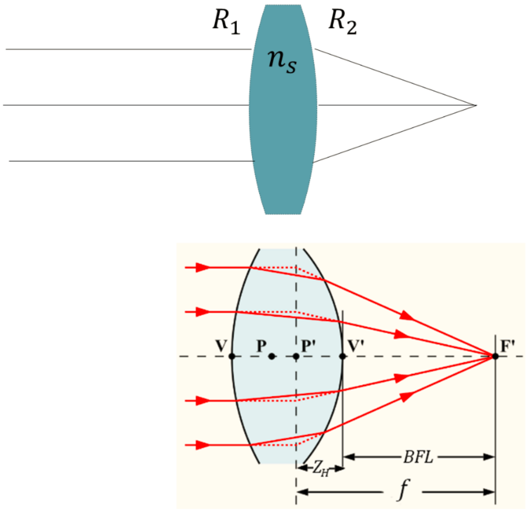


FIGURE F.1: A thick lens and its principal (P,P') and vertex points (V,V'). Principal plane Z_H: distance from V' to P'

The optical power of a lens is the ability to converge or diverge light. For a thick lens, it is defined as:

$$P = P_1 + P_2 - \frac{d}{n} P_1 P_2 = \frac{n_s}{f} \tag{F.1}$$

, where d : central thickness, n_s: RI of the lens, n: RI of the medium and P_1 and P_2 the power of the front and back surface respectively:

$$P_1 = \frac{n_s - n}{|R_1|}, \quad P_2 = \frac{n - n_s}{-|R_2|} \tag{F.2}$$

BVP is the optical power related to the back focal length (BFL):

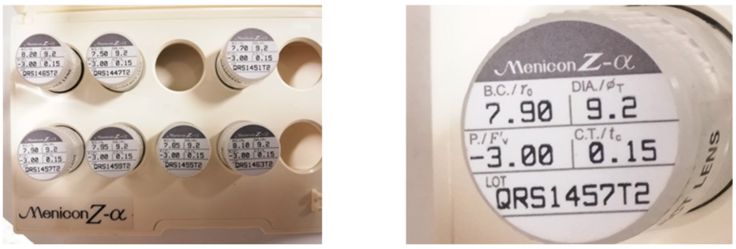
$$P_V' = \frac{n_s}{BFL} \tag{F.3}$$

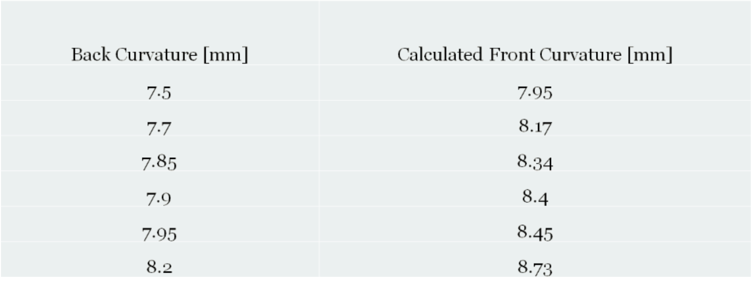
, where $BFL = f - Z_H$

From these we can conclude that:

$$P_V' = \frac{P_1}{1 - \frac{d}{n} P_1} + P_2 \tag{F.4}$$

Hence, in order to calculate the nominal FC, the above equation is solved for FC. Finally, the following table contains the calculated FCs for the six CLs that was measured in this thesis [33]:





Back Curvature [mm]	Calculated Front Curvature [mm]
7.5	7.95
7.7	8.17
7.85	8.34
7.9	8.4
7.95	8.45
8.2	8.73

Figure F.2: Calculated front surface nominal RoC for the measured CLs

Source: Menicon [33]

Appendix G

Uncertainty propagation

Every measurement is not perfect in terms of the accuracy and precision. Accuracy is relevant to the degree that the measurement device produces errors in the output, that is the deviation from the "real" ideal value. Hence for it to be measured, a reference measurement from another reliable device is needed. On the other hand, precision is relevant to the uncertainty or else variability that lies in the output value. The errors and uncertainties of the input parameters as well as of the the device itself produce (propagate) errors and uncertainties in the final result. Below, the general formulas for the uncertainty propagation are described. The output of the device produces some observed values, but often these values need to be fitted to a model. Then, an extra uncertainty is produced which originates from the deviation of the estimates from the observed values.

G.1 Uncertainty propagation: RoC

MAX FRINGE AREA IN SPHERICAL SURFACE MEASUREMENTS

As mentioned in the corresponding section, the RoC (hereby R) of the measured lenses is estimated by fitting a paraboloid.

The height of the spherical surface z is given by:

$$z = \frac{x^2 + y^2}{2R} \tag{G.1}$$

while R is given by:

$$R = \frac{x^2 + y^2}{2z} \tag{G.2}$$

Before, estimating the uncertainty of R, we can estimate the largest area (x by y) in which we can observe interference fringes. The period of the intensity variation of the fringes Λ depends on the wavelength of the light source λ and the angle between the interfering wavefronts θ:

$$\Lambda = \frac{\lambda}{2\,sin(\theta/2)} \tag{G.3}$$

Reflected wavefronts from the spherical surface form an increasing angle with respect to the reference mirror while moving further from the center of the spherical surface. This angle can be estimated by the rate of change of the height on the spherical surface (h) over the distance from the center (x, y), or equivalently with respect to x only:

$$tan(\theta) \approx \frac{dz}{dx} = \frac{x}{R} \tag{G.4}$$

For small angles, $sin(\theta) = tan(\theta) = \theta$. Hence, by combining Eq. G.3, G.4 we have:

$$\frac{\lambda}{\theta} = \frac{\lambda \cdot R}{x} \tag{G.5}$$

In order to adequately sample the interferogram, we constrain Λ to be ≥ 4 pixels or $\geq 4\ \sigma_x$, where σ_x refers to the sampling uncertainty of x. This means that the maximum distance from the center to observe fringes is equal to:

$$x_{max} = \frac{\lambda \cdot R}{4\ \sigma_x} = y_{max} \tag{G.6}$$

From Fig. G.1 we can deduce that if $x = x_{xmax}$, then the biggest height of the spherical surface that can be measured is given by the relation:

$$z_{max} = -\frac{\sqrt{4r^2 - x^2} - 2r}{2} \tag{G.7}$$

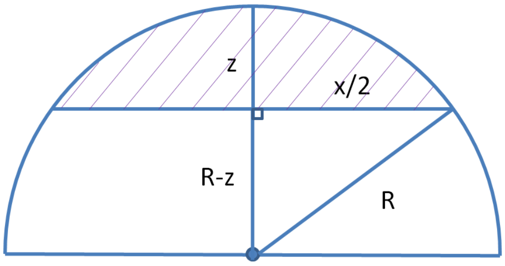


FIGURE G.1: 1D surface profile of a spherical surface. Purple lines show the measured area. Relation among height (z), area (x) of measurement and RoC (R) of the spherical surface.

RoC UNCERTAINTY

The relative uncertainty of R ($\Delta R/R$) is estimated as:

$$\frac{\Delta R}{R} = \sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2 + \left(\frac{\Delta z}{z}\right)^2} \tag{G.8}$$

By substituting $x = y = x_{max} = y_{max}$, $z = z_{max}$ and knowing that the uncertainties $\Delta x = \Delta y = 10\ \mu m$, $\Delta z = 0.015\ \mu m$ and also spanning R from 5 to 100 mm, we get the following relative uncertainty of R with respect to an increasing R:

Hence, from the theoretical approach we expect the relative uncertainties in the estimation of R to be ≈ 10%, 5% and 1% for the lenses with R ≈ 8, 18.8 and 92 mm.

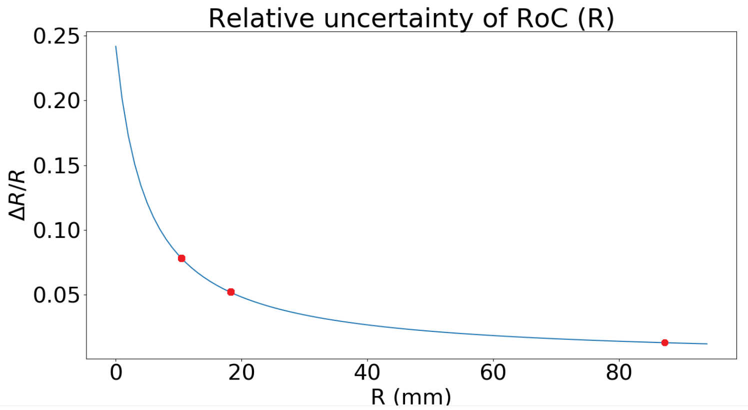


FIGURE G.2: Relative uncertainty of R with respect to an increasing R (5 to 100 mm). Red dots correspond to the nominal R of the measured lenses.

Bibliography

[1] J S Lundeen at English Wikipedia / Public domain. Interference finite coherence. https://commons.wikimedia.org/wiki/File:Interference_finite_coherence.png.

[2] Richard Leach and Stuart T. Smith. *Basics of Precision Engineering.* CRC Press, 1 edition, March 2018.

[3] University of Cambridge. Doitpoms - surface topography. https://www.doitpoms.ac.uk/tlplib/tribology/topography.php?printable=1.

[4] G. Binnig, C. F. Quate, and Ch. Gerber. Atomic force microscope. *Phys. Rev. Lett.*, 56:930–933, Mar 1986. https://link.aps.org/doi/10.1103/PhysRevLett.56.930.

[5] NT-MDT Spectrum Instruments. Contact atomic force microscopy. https://www.ntmdt-si.com/resources/spm-principles/atomic-force-microscopy/contact-afm/constant-height-mode.

[6] Y. Martin, C. C. Williams, and H. K. Wickramasinghe. Atomic force microscope–force mapping and profiling on a sub 100-åscale. *Journal of Applied Physics*, 61, January 1987.

[7] P. Demircioglu, Ismail Bogrekci, Numan Durakbasa, Neslihan Demir, and U. Kose. *Surface Topological Investigation of Seal Bushing by Using Chromium Plating Process*, pages 98–106. 05 2019.

[8] Debbie J. Stokes. Principles and practice of variable pressure environmental scanning electron microscopy (vp-esem). *Chichester: John Wiley & Sons*, 2008.

[9] NanoScience Instruments. Scanning electron microscope. https://www.nanoscience.com/techniques/scanning-electron-microscopy.

[10] Emil Wolf. *Introduction to the theory of coherence and polarization of light.* Cambridge University Press, 2007.

[11] Lecture notes: Notes D.G. Papazoglou, Wave Optics and Optical Metrology, MSc Optics and Vision. pages: 180–205.

[12] Elias N. Glytsis. Optical engineering: Spatial & temporal coherence. Lecture notes.

[13] Space-time analogy in synthetic coherence functions applied to optical tomography and profilometry. Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Relation-between-temporal-coherence-and-spatial-coherence_fig1_303311587 [accessed 26 Feb, 2020].

[14] F. Zernike. The concept of degree of coherence and its application to optical problems. *Physica*, 5(8):785–795, Aug 1938.

[15] Samuel J. Ling, Jeff Sanny, and William Moebs. *University Physics Volume 3*. OpenStax, Sep 2016. https://openstax.org/books/university-physics-volume-3/pages/3-5-the-michelson-interferometer.

[16] James C. Wyant. Phase shifting interferometry. Lecture notes: University of Arizona James C. Wyant College of Optical Sciences, Course: Optics 505.

[17] Peter de Groot. *Coherence Scanning Interferometry*, pages 187–208. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[18] Ioannis Orphanos. White light optical profilometry. Master's thesis, University of Crete, Department of Materials Science and Technology, 2008.

[19] PI Ceramic GmbH. Displacement modes of piezoelectric actuators, 2020. https://www.piceramic.com/en/piezo-technology/properties-piezo-actuators/displacement-modes.

[20] Thorlabs. Piezoelectric basic background and operation tutorial. https://www.thorlabs.com/NewGroupPage9_PF.cfm?Guide=10&Category_ID =220&ObjectGroup_ID=5030.

[21] Physik Instrumente. Open loop vs. closed loop performance of a typical pi piezo actuator. https://www.pi-usa.us/en/products/piezo-motors-stages-actuators/piezo-motion-control-tutorial/tutorial-4-32/.

[22] Akira Hirabayashi, Hidemitsu Ogawa, and Katsuichi Kitagawa. Fast surface profiler by white-light interferometry by use of a new algorithm based on sampling theory. *Appl. Opt.*, 41(23):4876–4883, Aug 2002. http://ao.osa.org/abstract.cfm?URI=ao-41-23-4876.

[23] Zygo$^{\mathrm{TM}}$. Ametek$^{\mathrm{TM}}$ Ultra Precision Technologies. *Mx^{TM}Surface Texture Parameters technical manual*, April 2018.

[24] Kenneth A. Goldberg and Jeffrey Bokor. Fourier-transform method of phase-shift determination. *Appl. Opt.*, 40(17):2886–2894, Jun 2001. http://ao.osa.org/abstract.cfm?URI=ao-40-17-2886.

[25] Physik Instrumente. *Preloaded Piezo Actuators datasheet.*

[26] Arduino. Introduction at arduino.cc. https://www.arduino.cc/en/Guide/Introduction.

[27] Michrochip. *MCP4821/MCP4822 datasheet.*

[28] Flir. Blackfly s bfs-u3-13y3m-c. https://www.flir.com/products/blackfly-s-usb3/?model=BFS-U3-13Y3M-C.

[29] Point Grey. Spinnaker application programming interface, Sep 2019. http://softwareservices.ptgrey.com/Spinnaker/latest/index.html.

[30] J. Calatroni, A.L. Guerrero, C. Sáinz, and R. Escalona. Spectrally-resolved white-light interferometry as a profilometry tool. *Optics & Laser Technology*, 28(7):485 – 489, 1996.

[31] Matina Vlachou. High resolution optical topography. Master's thesis, University of Crete, Department of Materials Science and Technology, October 2019.

[32] Paris Panagiotopoulos. Spectroscopic interferometry applied to dynamic profilometry. Master's thesis, University of Crete, Interdepartmental Postgraduate Programme in Optics and Vision, 2007.

[33] Menicon. Gas permeable lenses: Menicon z-α. https://www.menicon.com/pro/our-products/gp-lens/menicon-z-a/.