university of groningen / faculty of science and engineering

Automation of the mapping process of a compound eye with GRACE and Voronoi correlation

IEM Master Thesis

Giamfranco Bottone (S3798232)

g.bottone@student.rug.nl

First supervisor: Prof. Dr. Doekele Stavenga

Second supervisor: Dr. Mauricio Muñoz Arias

March 5, 2020

Abstract

The compound eye of arthropods has been a source of inspiration for several biomimicry apparatuses. The biology field as well as several engineering fields are aiming to obtain a better information of the house fly's eye acuity zone. The ultimate goal is to develop a mathematical model, simulation or test that can gather such information of the compound eye. Furthermore, that would open new possibilities, as the analysis could be broaden to even more insect species such as butterflies and introduce novel technologies, i.e. 3D cameras with spherical vision. The present report aims to tackle the current problems of the mapping process of the compound eye at the University of Groningen (RUG). The process used at RUG incorporates a GUI, which is believed it is impacting the flexibility of the algorithm as well as constraining the user customization depending on the required task. The reasons that have led to this preliminary diagnostic are concurrent failures, which mainly are difficulties to properly locate the house fly's pseudopupil; therefore, impacting in the centering of the fly inside the region of interest, as well as miscommunication with the ARDUINO and STANDA motors. Therefore, this research introduces a new automatized process, where the user intervention is minimized and the pseudopupil detection for autocentering have been highly improved. Moreover, a new approach for the image correlation based on voronoi is proposed. These proposed algorithms lead to interesting results as well as several recommendations for further research.

Table of Contents

1. Introduction

The compound eye may be the most interesting characteristic of an arthropod. Often related to insects, compound eyes are also present in other groups of arthropods such as crustaceans, centipedes, millipedes and horse crabs (Nilsson & Kelber, 2007). Compound eyes consist of thousands ommatidia and it is its architecture that sets the compound eye apart, due to their ability to adapt their design according to its location in the eye. This adaptability provides insects and other arthropods with a unique spatial resolution. Figure 1 presents a diagram of the ommatidia in a compound eye.

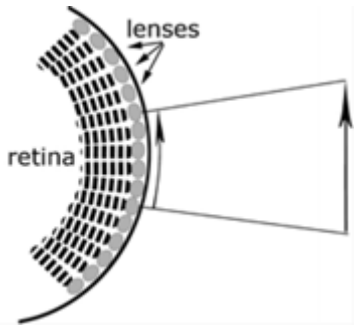


Figure 1. Ommatidia representation; spatial resolution is achieved through small lenses linked to each ommatidium(Buschbeck & Friedrich, 2008)

Scoping to insects, it is well known that these thousands of ommatidia (or small imaging units) are responsible for their remarkable large field of view (FOV)(Cheng, Cao, Zhang, & Hao, 2019).The compound eye spatial acuity is an area of interest for engineering mimicking as the insects FOV could improve blind navigating system, aircraft collision avoidance, general sensing, development of cameras, among other specialized fields by enhancing their reachable field. Therefore, understanding how an insect maps the world is of high interest.

In order to register properly the angular direction in which the different ommatidia are pointing, a map of the insect facets (an ommatidium face) has to be performed. The Goniometric Research Apparatus for Compound Eyes (GRACE) is an advanced apparatus used to microscopically capture images of compound eyes of arthropods. By applying image processing techniques to the images captured, the individual facets can be determined. Therefore, theoretically, a mosaic of the eye could be built.

By capturing several pictures at a different azimuth and elevation position an algorithm for stitching can be elaborated. However, guidance parameters for centering the photo have to be carefully identified. The centering guidance is considered to be the pseudopupil, as it represents the superposition of the virtual images in the center of curvature of the eye. Moreover, house-flies are of high interest for such mapping as the light phenomena do not fade away in time, phenomena that are present in butterflies.

The current mosaic building algorithms are not giving an accurate representation of the facet position. Therefore, in order to deeply understand the housefly acuity zone, a new approach to the matching of the images will be tested in this research as well as an optimal scanning path code that can maximize the quality and quantity of pictures taken.

2. Problem Analysis

In this section key concepts will be provided in order to understand the problem context. Furthermore, once the problem context has been introduced the problem statement will be presented.

2.1. Key concepts

Image Channels
Two classifications exist: RGB, which is the abbreviation for Red, Green, Blue, and the CMYK, for Cyan, Magenta, Yellow, Black. Those classifications are used for specific functions, the CMYK is used for designing colors that will be printed, as printers only recognizes those four pallets of color. On the other hand, the RGB files work with a model that is based on human vision.

As the research will focus on light phenomena, the model to be used is RGB. This is a way of tracking the amount of Red, Green and Blue light of each pixel.

Image Mosaicking
Image mosaicking refers to the construction of a unitary image which is constituted by several images taken in the same scene (RW.ERROR - Unable to find reference:6). There are two fundamental methods for image mosaicking:

- **Direct method:** In a direct method all the pixel's intensities are compared with each other. Measuring the contribution of each pixel in the image (Kota & Jonnalagadda, 2016)
- **Feature-based method:** In a feature-based method the comparison of the images is done by characteristic points (local descriptors) (Kota & Jonnalagadda, 2016)

Zero Mean Normalized Cross Correlation (ZNCC)
Zero mean normalized cross correlation (ZNCC) is an image mosaicking technique that enhances the widely used correlation formula Normalized Cross Correlation given by Eq. (1). ZNCC and NCC are widely used given the robustness. ZNCC is defined as, let I be the image under examination, of size W · H pixels, T the template sub-image, of size M · N. With mean values of $\overline{Mı}$ and $\overline{\overline{Mıc}}$, given by Eq. (2) (Di Stefano, Mattoccia, & Tombari, 2005).

$$NCC = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}[I(x+i,y+i)]*[T(i,j)]}{\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}\|[I(x+i,y+i)-]\|^2}*\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}\|[T(i,j)]\|^2}} \quad (1)$$

$$ZNCC(x,y) = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}\left[I(x+i,y+i)-\overline{Mı}\left(I_c(x,y)\right)\right]*[T(i,j)-\overline{Mıc}(T)]}{\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}\left\|\left[I(x+i,y+i)-\overline{Mı}\left(I_c(x,y)\right)\right]\right\|^2}*\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}\|[T(i,j)-\overline{Mıc}(T)]\|^2}} \quad (2)$$

Voronoi
The partitioning of a plane into several regions, which contain the points that are only closest to the "seed" of the region is called a Voronoi diagram or Dirichlet Tessellation. Once a point is not closest to the region at hand, a boundary is formed; therefore, partitioning the plane in multiple polygons. In order to construct a Voronoi diagram, the "seeds" or "generators"

are needed, these are the reference points and centroid of the polygons. Figure 3 gives an example of a Voronoi diagram (Dobrin, 2005).

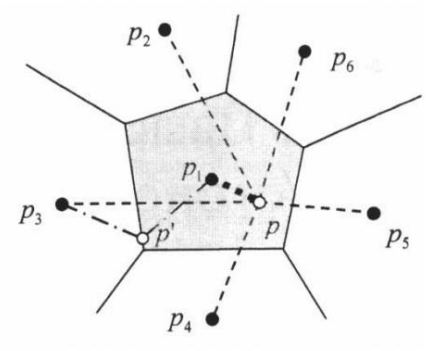


Figure 3. Voronoi diagram. P points represent the "seeds" or "generator" points. The shadowed P1 polygon represent the set of point closest to P1 and only to P1. Moreover, p' represent the boundary where that set of points is separated by the same length to P1 and P3. Therefore, regions share common vertex points (Dobrin, 2005).

The Ommatidium

An ommatidium is the building block of a compound eye (Stavenga, 1979). At the top of the ommatidium, the facet lens is located, which is followed by a crystalline cone in honeybee eyes and a so-called pseudo-cone in fly eyes. Surrounding the cone, there are the primary pigment cells. The pigments in the primary pigment cells are ommochromes and many arthropods use it for yellow, red or brown coloration of the eye (Rubin, Miserez, & Waite, 2010). The secondary and tertiary pigment cells surround the whole ommatidial unit from surface to base, and enswathe the lens and photoreceptor array in a deep coating of pteridine and ommochrome screening pigments (Tomlinson, 2012). The crucial element of a photoreceptor is the rhabdomere, a cylindrical organelle that contains the visual pigment. The tips of the rhabdomeres coincide with the focal plane of the facet lens. At the bottom of the photoreceptor cells are the axons. The axons transmit the signals to the nervous system. Figure 4 shows an individual ommatidium of a house-fly, which is the insect to be studied. A represents the retinula-cell axon, Cc the cone, L the corneal facet lens, Pc the pigment cells, Rh a rhabdomere and Rc the retinular cell.

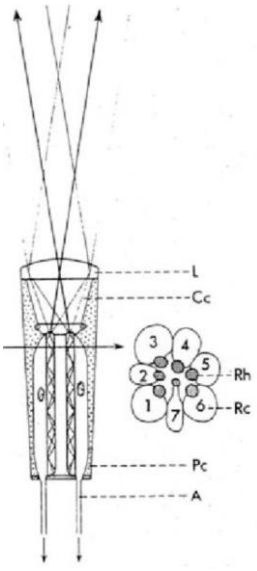


Figure 4. Ommatidium and cross section of a house-fly (Bitsakos & Fermüller, 2006).

Lattice of the Compound Eye (House-fly)

The representation of each facet lens center can be seen as a two dimensional lattice which is of the centred rectangular type (Stavenga, 1979). Commonly, the shape of the pattern is constant throughout the eye; however, some variations occurs. When moving from anteromedial (front and toward the middle) to posterolateral (posterior and lateral) the facet lens mosaic gradually changes from a hexagonal into a square facet lens pattern, and then into another hexagonal pattern (Stavenga, 1979). Figure 5 and 6 give a graphical representation of the house fly pattern.

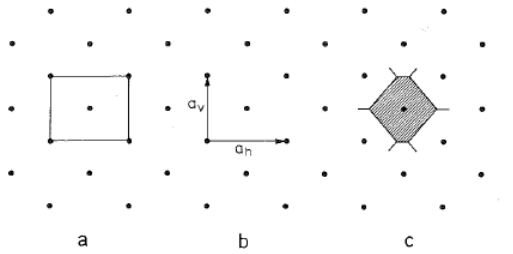


Figure 5. a) Is the so called unit cell; which is a rectangle with a lattice point as center and one in each corner, c) cell containing all points closer to the central lattice point that any other lattice point. (Stavenga, 1979)

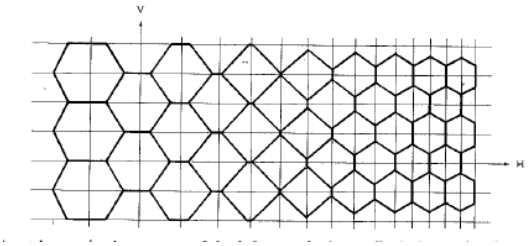


Figure 6. Facets pattern variation from anteromedial to posterolateral (Stavenga, 1979)

Inter-Ommatidial Angle

The acuity of compound eyes is determined by the interommatidial angles, optical quality, and rhabdom dimensions (Land, 1997). Optical quality is the process when an object becomes sharper the image contrast decreases until there nothing left to detect, the maximum resolution is dependent to the optics of the individual. Moreover, when details are smaller than the width of a rhabdom they are lost. In addition, The acuity is affected by three structural and two environmental features: a) angular space between receptors, b) quality of the optics, c) diameter of the photoreceptors; for the latter a) the amount of light b) motion. However, maximum acuity can be computed by knowing the interommatidial angle. In a compound eye the angular space between receptors refers to the interommatidial angles. Figure 7 shows the interommatidial angle.

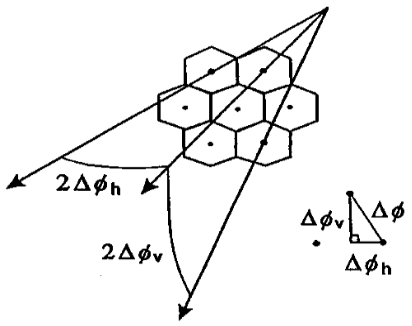


Figure 7. Describing inter ommatidial angles in the lattice pattern (Land, 1997)

Knowing the interommatidial angle $\Delta\phi$ the computation of the acuity is possible by the following formula. Moreover, Figure 8 shows the interommatidial angle from a different perspective.

$$vs = \frac{1}{2\,\Delta_{\phi}} \quad (3)$$

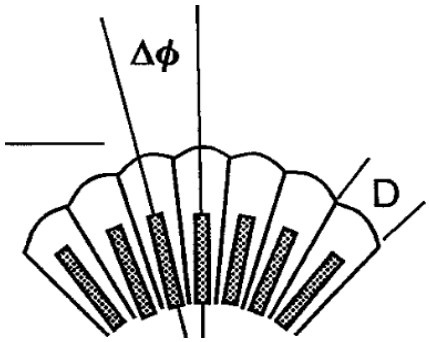


Figure 8. Angle between two receptors in compound eye where D is the facet diameter and Δ_{ϕ} the angle

Pseudopupil

Figure 9 illustrates that rhabdomeres are located in the inner focal plane of the facet lens, and virtual images of them appear in the depth of the ommatidium (Franceschini & Wehner, 1972). Therefore, sharp images are possible to observe when only at a certain distance from the cornea level (≅1mm). The "Pseudo" phenomenon happens when the focus of the eye is applied in an intermediary plane. Figure 9 represents the different planes for each ommatidium seen as the whole eye.

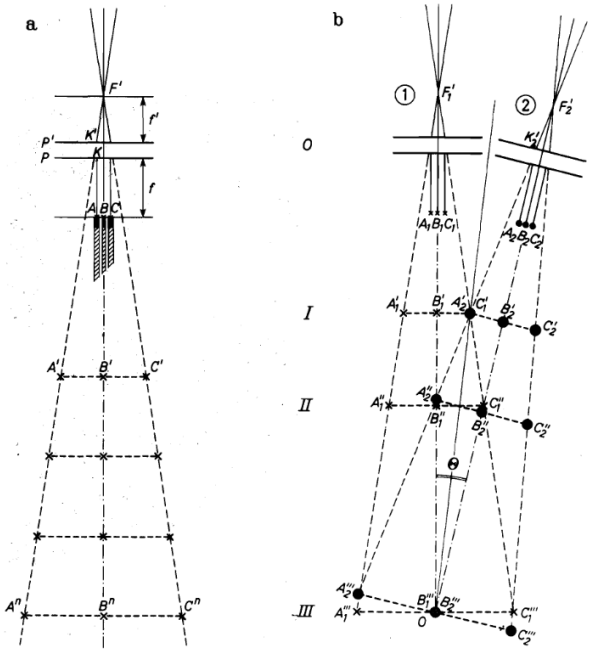


Figure 9. a) represents the ommatidium as an individual, A B C represent the rhabdomeres with A' B' C' and A^n B^n C^n illustrating the center of the intermediary planes. B) Represents the superposition of virtual images (intermediary planes) (Franceschini & Wehner, 1972)

Figure 9.b shows that the first superposition happens on the bisect of each interommatidial angle, the second one at the optical axis of each ommatidium and the third one at the

center of curvature of the eye (Franceschini & Wehner, 1972). When the eye is focused at a distance when the planes are superpositioned, at the center of the eye (third level), the phenomena called deep pseudopupil occurs, and when the eye is focused at the cornea level the corneal pseudopupil occurs.

GRACE

The Goniometric Research Apparatus for Compound Eyes (GRACE) is an apparatus developed to acquire images from the arthropod compound eyes. The GRACE system is built by the combination of mechanical and optical systems. The mechanical system consists of an Arduino controller model MEGA, three XMIC servo controllers and five STANDA motors. The Arduino MEGA is responsible for controlling the LED intensity, which travels through some diaphragms and is responsible for the light source impacting the eye to be studied, and for controlling the microscope height motor. The XMIC servo controllers are responsible for moving the system in the 5 possible coordinates (X,Y,Z,E{Θ},A{φ}) and the STANDA motors are the equipment actually performing the movement. The optical system consists of a camera pointing in the Z axis. On the other hand, the light from the LED is being directed by a half mirror. The half-mirror is both reflective and translucent in order that the reflected image can reach the microscopic camera, that drives the light signal directly to the eye. The camera and lenses capture the light effect in the compound eye perpendicular to the camera plane. Figure 11 gives a representation of the system.

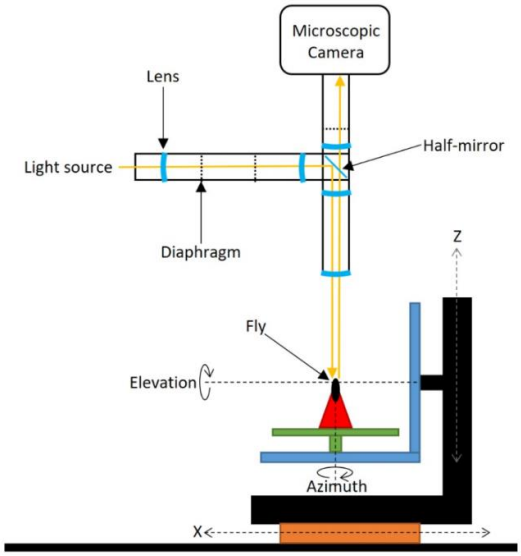


Figure 11 GRACE system representation. The plateaus are labeled with colors. The fly can rotate in a 5 degree of freedom movement; the green label can be move in the Azimuth direction (A{φ}), the black label can be move in the Z direction, X is moved by the orange label, Elevation (E{Θ}) is moved by the blue label while Y is perpendicular to this plane.(Niemeijer, 2018)

2.2. Problem Context

The compound eye of insects provides an unlimited source of inspiration for designing new apparatus that would positively impact technology development (Viollet et al., 2014). In fact, cameras, high resolution lenses, blind monitoring sensors, radian heaters, anti-reflecting coating, among several others groundbreaking inventions are inspired by the compound eye advantages and architecture. However, how to perform the visual mapping of an insect eye is still far from completely understood.

The Biology department along with the Engineering department of the University of Groningen have been trying to understand the visual attributes of the compound eye for years. The group developed the GRACE system with the goal of obtaining high quality pictures of the compound eye and further process them in order to obtain useful data. The fly can be carefully positioned on a plateau, which can move in any direction under a microscopic camera and aligned to the camera visual axis. The system is linked to a PC with an Intel Core i7-6700 CPU 3.40GHz and 8GB DDR4, that allows the control of the motors through MATLAB scripts; however, the possibility for manual control is also available. MATLAB scripts are coded instructions sent from the PC to the ARDUINO and servomotors, allowing a high precision movement of the motors, which consequently, move the plateaus where the fly is located. A light source sheds light onto the fly eye creating the pseudopupil phenomenon. Then, the pseudopupil will be used as the image centering guide, in other words, the center of the pseudopupil has to be the center of the image. Next, when the fly is centered, snapshots can be taken in several orientations of the fly's eye.

In order to obtain information of the visual axis of the house fly, interommatidial angles have to be identified. Therefore, a single image does not provide enough information about the angles. Thus, image mosaicking is necessary in order to derive the interommatidial angles. A stitching algorithm was developed, which can stitch images of the housefly in a specific elevation line, along the azimuths. However, several challenges have yet to be overcome in order to obtain the scan of a complete eye.

The current interaction with the GRACE system is based on a graphical user interface (GUI), which is limiting the user control over the scanning. Moreover, the autocentering script has considerable difficulties identifying the pseudopupil of the eyes, resulting in an off-centered photo. Therefore, the scanning procedure has to be performed semi-automatically where the user has to center manually in every new direction of the fly's eye. Thus, an elevation line would take approximately 40-50 min to be performed. Scanning different elevations to obtain a full eye scan would imply an unrealistic amount of time and several human errors given that the user has to be fully focused on each step. In addition to the scanning difficulties, the current stitching program is developed having in mind only one elevation line; therefore, modifications have to be made in order to be able to handle the amount of data required for a whole scan. Moreover, the different areas (depending on the eye direction) where the facet patterns are too similar, are prompt for errors when correlating the image, resulting in a wrong output. Therefore, multiple enhancements in the scanning procedure as well as stitching are needed in order to obtain the relevant data (pictures) representing a fly's eye.

2.3. System Description

In this chapter the system (GRACE) of the research will be explained. The aim is to describe the current MATLAB scripts controlling GRACE. Moreover, the coordinate system that GRACE employs will be elaborated on. Figure 12 represents the system.

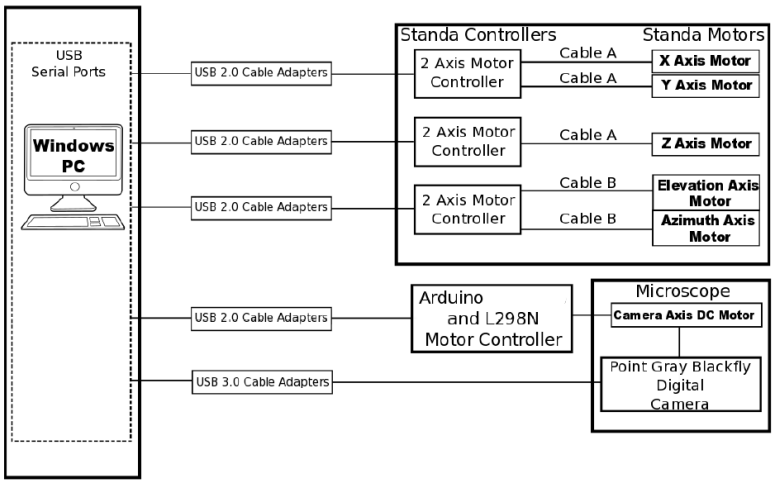


Figure 12. System GRACE. "All axes are controlled with Standa stepper motors except for the sixth camera axis, originally powered by an Arduino Mega, which remains unused. The Standa motors are controlled by the Standa 2-axis motor Controllers. The cable A (cable code- 8CA9F-15MR) is used to connect the controller (Model No.-8SMC4-USB-B9-2), and the X, Y, and Z motors (Model No.- 4247), cable B (cable code- 8CA15MR) is used to connect the elevation and azimuth motors (Model No.- VSS41). These motors are all connected to the Windows PC and are actuated with Matlab. The Point Gray Blacky Digital camera is used to capture the images. The camera has a resolution of 1920 *1200 pixels which translates to 2.3 MP" (Sahu, 2017).

2.3.1. Coordinate system

The apparatus uses the combination of two coordinate systems, Cartesian and Polar. The Cartesian coordinate system refers to the movement along the conventional X, Y, Z system, while the polar system involves Elevation (Longitude {Θ}) and Azimuth (Latitude {φ}). As the camera is aligned with the Z-axis, it is important to define the top of the hemisphere as the origin of the polar coordinate system. Figure 13 and 14 illustrates the coordinate systems.

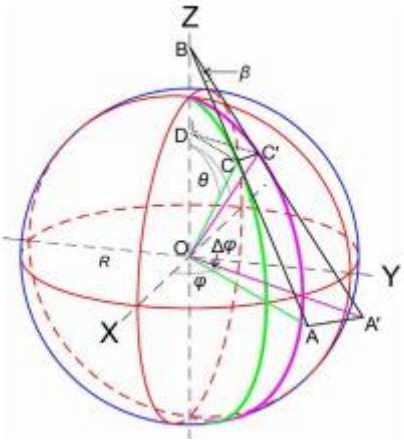


Figure 13. Diagram representing both coordinate system. It can be derived that the apparatus has five degrees of freedom.

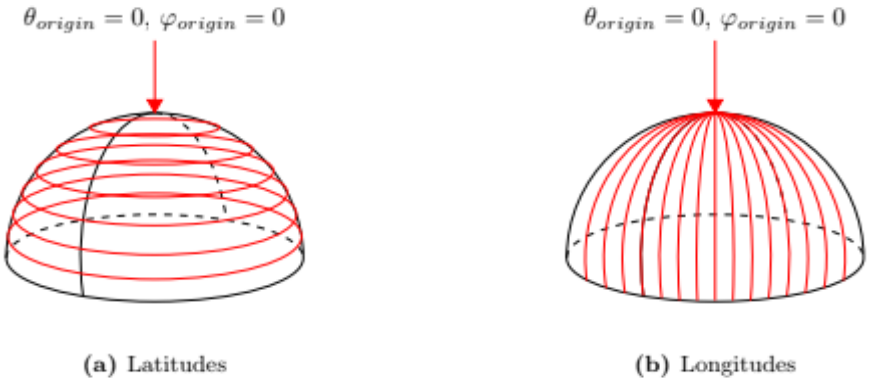


Figure 14. a) latitudes is formed by azimuth rotation b) longitudes in formed by elevation rotation (Niemeijer, 2018).

2.3.2. MatLab Scripts

Several scripts have been developed in order to control the scanning procedure. The most critical scripts are going to be explained as follow:

1. MovingMotorsTest

The function developed inside this script requires as an input the device ID of the STANDA motors, which is an integer from 1-5 depending on the motor to be rotated, and the number of steps to be moved. It is known that 100 steps represent 1 degree for the EL(Θ) and AZ(φ) coordinates. On the other hand, for X, Y, Z the distance per steps is 2.5 µm.

2. Auto Focus

The aim of this function is to obtain the sharpest image before the snapshot. The script defines a number of samples to be taken along the Z-axis in order to yield the optimal position on the plateau. After each sample, the script computes the moving average and plots the values; the peak of the plot represents the best focus. When the peak is at one of the plot borders, the process has to be repeated as the sharpest possible image has as yet to be detected.

3. Graphical User Interface

The GUI is the script that integrates all functions of GRACE actuated by MATLAB. However, this GUI is constraining the freedom on the user over the control of the scan and speed. Important errors are currently occurring in the Auto-Centering and LED control when using the GUI. Developing a script without the necessity of a GUI that can automatically scan several elevations lines with high centering accuracy is desired.

4. Auto Centering

This script is vital for the scanning procedure. The function computes the ratio *R* between the Red and Blue channel of the image. Then, proceed to subtract the multiplication of the B channel times *R* to the R channel. With this procedure the pseudopupil would theoretically be the only data left. Next, a Gaussian filtering is applied in order to identify the centroid of the pseudopupil. With the coordinates of the centroid of the pseudopupil, the function determines the amount of step*pixel to be actioned to yield a centered image.

5. Stitching algorithm

This is a complex algorithm that uses several functions in order to achieve the desired combination of the images. First, two overlapping images have to be the input of the program. Next, the algorithm proceeds to crop and rotate the images depending on the direction they were taken. The images need to be rotated in order to approximate the spherical coordinates to a 2D line. Afterwards, the cropped image is filtered to the blue channel where the facets' pixel data is located. Then, a threshold is applied to the image, and by using "regionprops" the centroids of the facets are located, see Figure 15. Moreover, the algorithm then uses "normxcorr2" to apply the normalized cross correlation. The results of this correlation are the offset of the pictures that then are applied to the facets for plotting motives. See figure 16. A Voronoi correlation approach will be tested in this research for a higher accuracy.

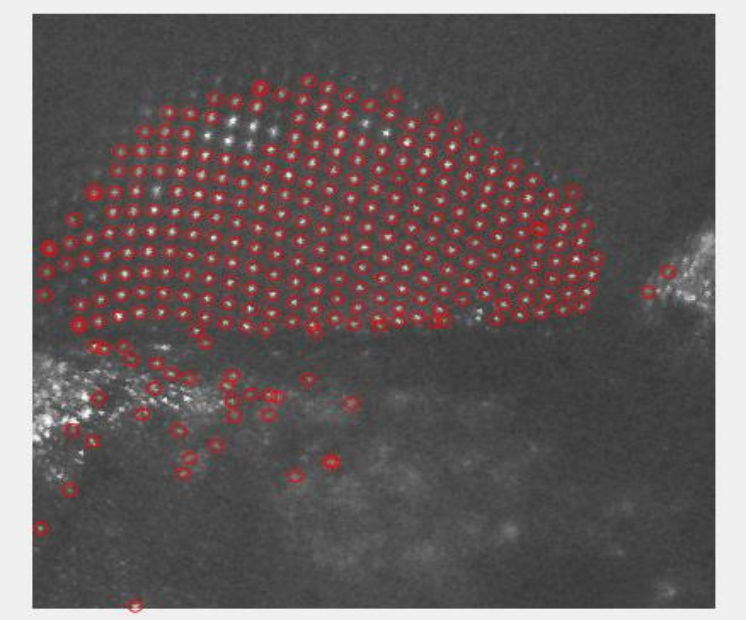


Figure 15. facets plot over the eye image

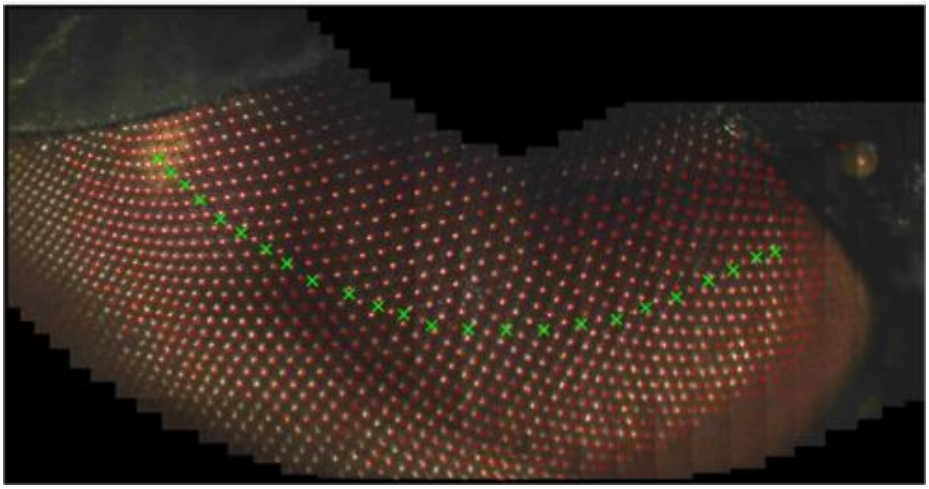


Figure 16. Current stitching algorithm

2.4. Problem Statement

The problem context analysis consequently yields the following problem statement:

"The procedure to run a scan through the GRACE system requires a considerable amount of human intervention, and the scripts for obtaining the required data for a complete fly eye image mosaicking are experiencing several bugs that are heavily impacting the process. Thus, the required data for a complete eye mosaicking have not been able to be collected. Moreover, the correlation algorithm is having difficulties to identify the proper offset in areas where facets patterns are highly similar"

3. Methodology

Theory oriented research aims to solve a problem encountered in the theory development of a specific scientific area, regarding a specific issue. Given the probabilities of an individual to change completely a theory or develop a new one, specific parts of the theoretical problem must be selected. On the

other hands, a practice-oriented research is meant to provide knowledge and information that can contribute to a successful intervention in order to change an existing situation. Following the previously mentioned definitions provided by (Verschuren, Doorewaard, & Mellion, 2010) the presented research will be based as a theory oriented research.

There are two types of theory-oriented research. First, theory developing, which aims to the filling of gaps in the construction of a theory. Second, theory testing research, which aims to test existing points of views and adjust them if necessary. This research will have a combination of both. On one side this research will improve the scanning procedure of the house fly, on the other, a new approach by Voronoi diagrams will be tested for the correlation procedure to verify if the approach would indeed perform as desired and yield more accurate results.

(Wieringa, 2014) has also identified a contrast within generalizability of a project and its realization. Generalizability refers to the possibility of applying a design to different environments. Therefore, following the diagram presented in Figure 17. The case for this research would lay on "Existential Generalization" as the knowledge provided by it is of high interest along the biology field.

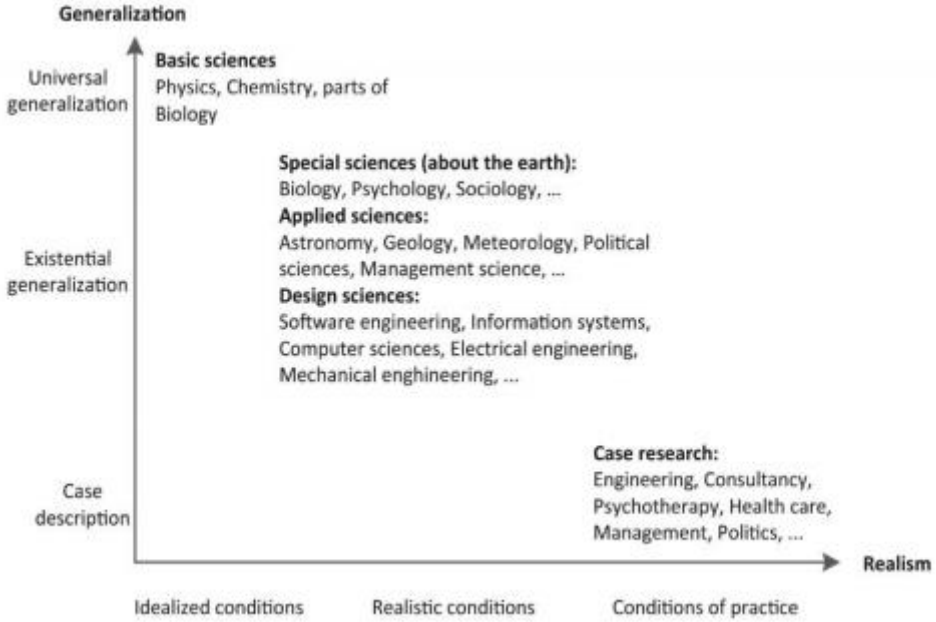


Figure 17. Wieringa's map for different knowledge [19]

4. Engineering cycle

The engineering cycle is a rational problem solving problem (Wieringa, 2014). Such a cycle consists of five stages:

- Problem investigation
- Treatment Design
- Treatment Validation
- Treatment Implementation
- Implementation Evaluation

For an engineering cycle, the problem needs to be properly identified and defined, and it must be diagnosed beforehand (Verschuren et al., 2010). Aligning to the presented research, as it is well known the difficulties present on the scanning and stitching process, the problem has been already diagnosed and it requires the initialization by the stages of the cycle.

5. Design goal

"Develop an automatic scanning procedure which would be able to perform the complete scan of the housefly eye within six hours. While, testing a new approach of correlation of the output images based on Voronoi in order to investigate possibilities of a correlation output of 0.9+ while keeping a high accuracy regarding the facets region. Within the 21 weeks given by the University of Groningen"

5.1. Validation

The goal above would be validated under the following set of criteria: Table 1 explains the criteria for the scanning procedure while table 2 for the Voronoi correlation.

Table 1. Evaluation criteria for the scanning process

	Criterion	*Description*
1	*Scanning time*	*90 degrees of the house fly should be completed in less than 2 ½ hours. While a full 180 degrees should be completed under 6 hours.*
2	*Quality of the Images*	*Output pictures have to be sharp and centered. Centered refers to: the eye's pseudopupil have to mark the center of the picture*
3	*Automation*	*The complete procedure should only involve the user for visual inspection and intervention in case noise are present due to movement of the fly. The code should be able to move through lines of Az and El as well as identifying borders. A border is an area where the pseudopupil is lost*

Table 2. Evaluation criteria for Voronoi correlation process

	Criterion	*Description*
1	*Regions*	*Obtaining a matrix with pixel values for each region*
2	*Correlation*	*The correlation output between regions to be of 0.9+*
3	*Distance validation*	*Distance taken from reference points should match from the offset between regions.*

6. Research Question

1. ***How to develop an algorithm able to move around Az and El lines that the GRACE system would follow without the necessity for user intervention***

The above central question has the following sub-questions:

1.1. How can the boundary (cuticle of the eye) be defined in order for the system to understand it needs to stop and move to the opposite direction?
1.2. Which parameters defines the identification of the pseudopupil?
1.3. How to accurately identify the pseudopupil for centering?
1.4. What would be more time effective, maintaining the GUI or developing an algorithm consisting only of functions

2. ***How can the Voronoi theory enhance the correlation program?***

The above central question has the following sub-questions:

2.1. Which parameters defines the seed of the Voronoi cells?
2.2. How can the region matrix be developed?
2.3. Which of the current state of art correlation models would perform better for region by region correlation?

7. Experiment

In this section the report will explain the procedures applied for the obtaining of the results. Elaborating on the different aspects of both the scan process and the Voronoi correlation approach.

7.1. Scan

The developed algorithm aims to successfully automatize the obtainment of the common house fly's eye photos. However, before elaborating in depth about the procedure implemented, the report will explain why the house fly was of high interest for the research at hand.

The house fly is a widely researched animal. Biologists along the world are highly interested in obtaining useful data of this animal in order to comprehend their behavior. Currently, ground breaking works about virtual imaging the fly brain have been released. An example is the work of the Janelia Research Campus with Google to obtain a high resolution map of the house fly brain, recently published this year. Even though, several researches are being carried out about the house fly, a complete comprehension of the fly compound eye is still not acquired. Moreover, the analysis proposed for the fly could be applied for the comprehension of other insect compound eyes, e.g. butterflies. Therefore, given the biologists' interest in house flies it would be beneficial studying this animal. Nonetheless, house flies have interesting characteristics that are advantageous for the procedure.

As opposite to butterflies, house flies' pseudopupil intensity increases as time under the light passes. The fly's pseudo pupil will reach a stable intensity after a few seconds (approximately 60 seconds) as seen in Figure 18.

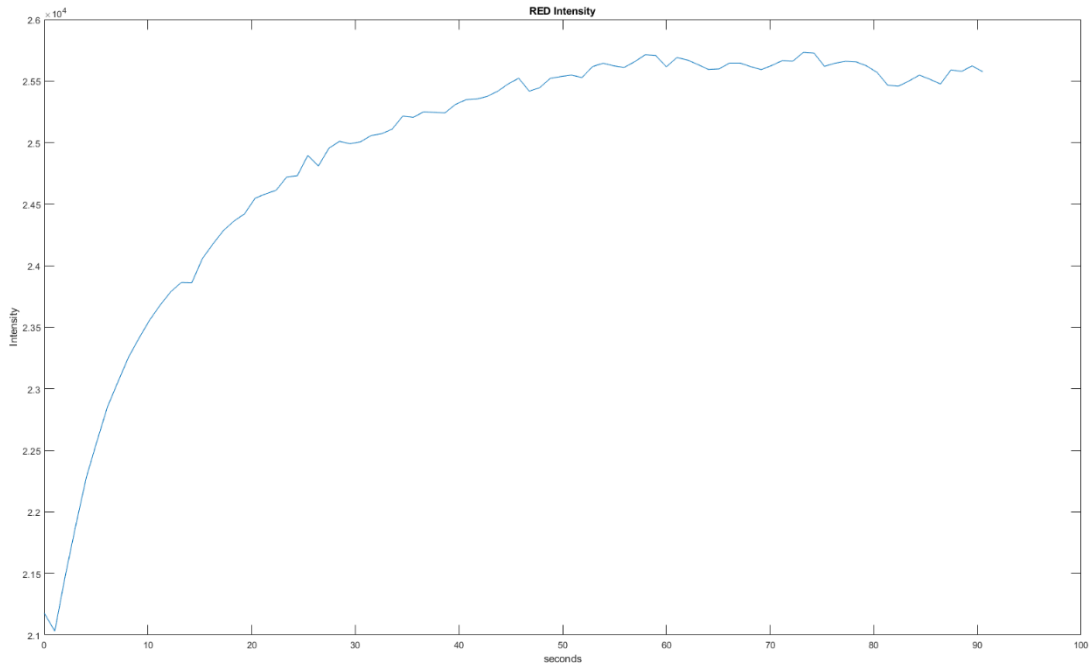


Figure 18. Pseudo pupil analysis on the Red Channel [20]

This behavior is highly interesting for the research purpose. Because, the pseudopupil will remain visible regardless of the amount of time in the presence of light. And second, given the season the research took place, which was winter, the house fly was one of the few specimens available for analysis.

Therefore, the procedure implemented was as follows:

1. The first step of the procedure is the preparation of the fly. The fly is introduced carefully in a plastic tube, in which it is pushed until its eyes are clearly visible. Then, for stability purposes, it is waxed to the tube with a small spot. Figure 19 and 20 shows the sample after preparation.

Figure 19. Sample prepared after tube insertion and waxing

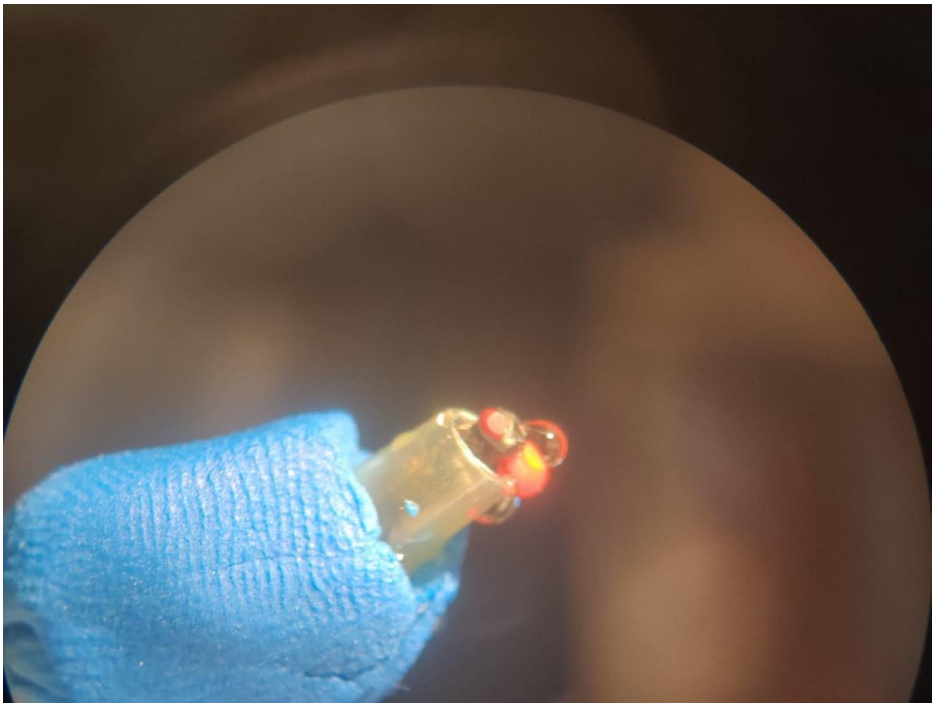


Figure 20. Sample zoomed in.

2. Second, the centering of the plateaus has to be performed with a special guiding pin. The pin has a sharp dent, of which the tip has to be sharp at El0 and AZ0 in order to center and focus the Z axis. Next, AZ and EL rotation have to be centered. In order to center them, the tip of the pin must be at the intersection of the axis. By placing the pin in such intersection, the tip of the pin will remain focused along any EL and AZ rotation.

3. The third step is the placement of the specimen on the GRACE apparatus. The plastic tube should be introduced in the magnetized pin. The pin allows the user a rotation of 180 degrees along the X-axis. Then, the tube should be carefully stitched to the pin with clay, and place in the Azimuth plateau of the apparatus. At the moment of placing the sample in the plateau the user should carefully position the top of the eye aligned with the Z-axis. Moreover, it is critical that the focusing of the fly is done by moving the pin through the 180 degrees of freedom instead of operating the STANDA Z-axis motor to prevent losing the configuration performed in the previous step. Image 21. Shows the position of the specimen.

a) b)

Figure 21. a) Show the fly position such that the top is aligned with the Z axis illustrated by the red arrow. B) shows the plastic tube clayed to the pin, the arrow illustrates the 180-degree movement of the pin

4. The initialization of the system is performed by starting the MATLAB script "Inizialization2020", please refer to section 7.1.1 subsection 1 and for the detailed code refer to Appendix A. The script is in charge of uploading the libraries needed for controlling the STANDA motors. After the motors' initialization, a test is performed to corroborate that the device IDs are given properly, the order of movement should match the number and axis name. Next, the script proceeds to initialize the camera, which in this case is a Point Grey model BFLY-U3-23S6C-C. Afterward, the ARDUINO MEGA is initialized. And finally, the ROI is manually selected and the video is started.

5. Once the equipment is centered and initialized, the developed script for scanning can be applied, which is called "Scanning2020", please refer to section 7.1.1 subsection 2 and for the detailed code please refer to appendix B. The presented script does not longer require a GUI. Meaning that the user has total freedom to configure the scan at his best convenience as well as stopping the procedure in case of an unexpected issue raises. For the development of the script the following path plan is applied: At EL0, which represent the top of the eye, only one picture should be taken. Refer to subsection 2.3. Then, from EL10 to EL30, azimuth steps of 20 degrees should be taken. Refer to subsection 2.4. Once the EL reaches 40 degree, the azimuth steps should be reduced to 10 degrees. Refer to subsection 2.5. This path should produce the following set of pictures illustrated by Figure 22.

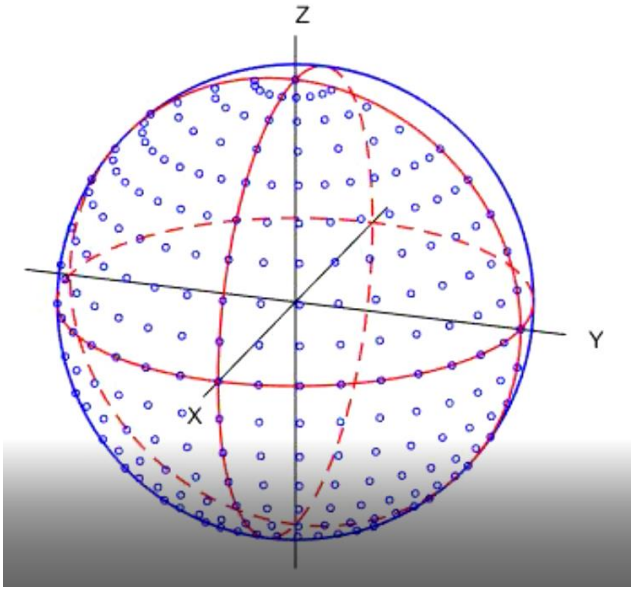
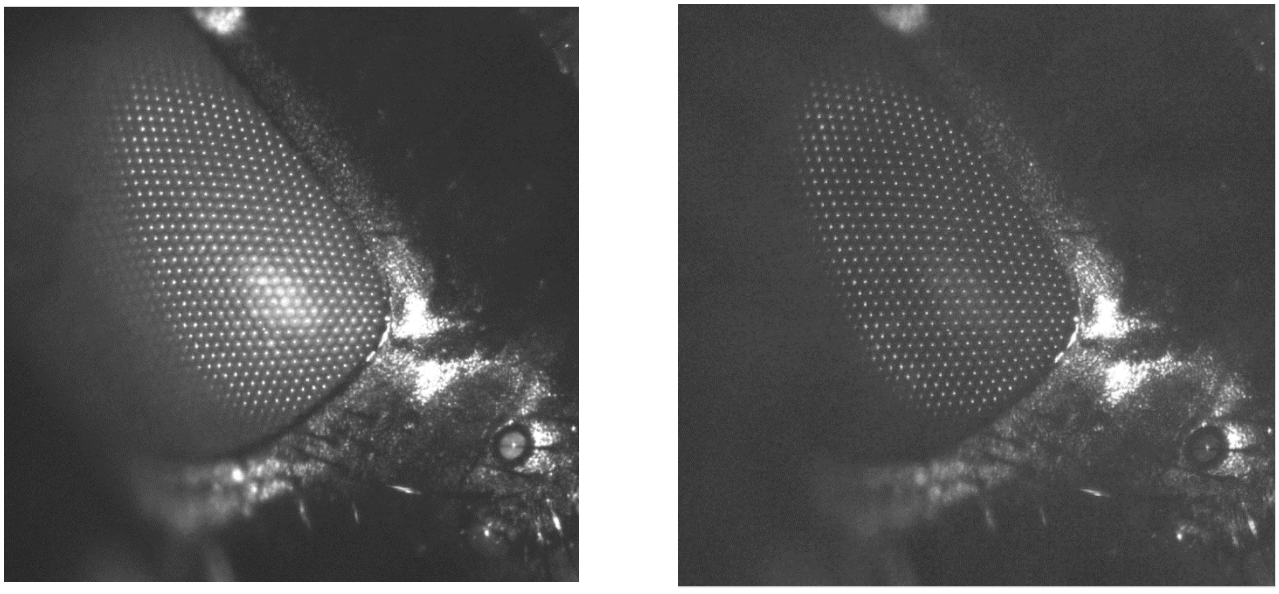


Figure 22. Desire set of pictures following the path explained above. Where each blue circle represents the center of a picture taken.

However, "Scanning2020" is built with critical functions that were further developed as they were malfunctioning with the GUI. It was identified that auto centering *("AC" please refers to appendix C,* was not identifying the pseudopupil perfectly; therefore, impacting the overall time as well as quality of the picture. To briefly explain, the previous *"AC"* script identified the pseudopupil by implementing a black box that covered the region of the image where such was located. Then, a summation of all the channels is performed to estimate a ratio *R* between the red a blue channel. In the case of the pseudopupil the red channel is of high interest as it has the data for the pseudopupil as seen in Figure 23a. On the other hand, blue channel, see Figure 23b, has the information for the facets and cuticle. Once the value of ratio *R* has been estimated the script subtracts the information of the blue channel from the red channel, applies a Gaussian to the remaining of the picture and ideally the pseudopupil should have been identified.



a) b)

Figure 23. a) Red channel where the pseudopupil is clearly visible. B) blue channel where facets and background is visible

This script was modified with the aim to acquire more accurate data, while making the process faster. The developed script works with light- and dark-adapted pictures. Light-adapted refers to a picture taken 10 seconds after the LED has been turned on, which aims to obtain a picture with the pseudopupil effect stable. Dark-adapted refers to a picture taken immediately taken after turning on the LED having passed 10 seconds wait with the light off, which aims to obtain a picture without the effect of the pseudopupil. The picture taken in both cases is an average of three images taken in an interval of 0.1 second, which yields a sharper image. Once the photos have been taken, the red channel from the dark-adapted picture is subtracted from the red channel of the light-adapted red channel. Then a Gaussian filter is applied and with "regionprops" the centroid of the pseudopupil is identified. Afterwards the offset from the center of the ROI can be estimated, which then proceeds to center the fly. For movement is X and Y is estimated that the steps per pixels is =0.523.

It is expected to find in the planned path cuticle borders. As the path starts at AZ=0 and needs to move through positive degrees and when hit the border return and moves towards a negative direction in azimuth the identification of the loss of the pseudopupil had to be done. In each "AC" call the script keeps track of a so-called variable "Hmax". Hmax represents the maximum intensity present after applying a softening by Gaussian. Depending on the value of Hmax, which was estimated to be above 2000 for a picture with pseudopupil presence and below 2000 for an image without pseudopupil presence, the algorithm knows if it should stop and goes back. Refer to subsection 2.4.6 and 2.5.5. Depending on which direction the scanning is being performed it will either go back to a negative direction or it will return to AZ=0 and then move the elevation plateau by 10 degrees. "AC" also allows to keep track of the displacement of x and y and it should be taken into account when returning to any position to make sure the pseudopupil returns inside the ROI. The proposed script allows an automatize scan from 0 to 90 degrees. However, cuticle reflection should be supervised as it can be identified as a pseudopupil when the pseudopupil is lost.

7.1.1. MATLAB code overview

1. **Initialization**
   - 1.1. Initialize STANDA motors
   - 1.2. Upload libraries
   - 1.3. Motors test
   - 1.4. Camera initialization
   - 1.5. Initialize variable "mega"
   - 1.6. The user defines the ROI
   - 1.7. Starts the video
2. **Scan**
   - 2.1. Defines AZ vectors for the set of EL<40º and EL≥40º
   - 2.2. Check EL value
   - **2.3. If EL=0**
     - 2.3.1. Auto-focus
     - 2.3.2.Auto-center
     - 2.3.3.Auto-focus
     - 2.3.4.Save file
     - 2.3.5.Move EL by 1000 steps
   - **2.4. If 0<EL<40**

2.4.1. Auto-focus
2.4.2.Auto-center
2.4.2.1. Save X and Y displacement
2.4.3.Auto-focus
2.4.4.Take average snapshot
2.4.5.Save file
2.4.6.Check Hmax
2.4.6.1. If Hmax<2000
2.4.6.2. Return the amount of AZ steps plus -2000
2.4.6.3. Apply X and Y displacement
2.4.6.4. Auto-Focus
2.4.6.5. Auto-Center
2.4.6.6. Auto-Focus
2.4.6.7. Take average snapshot
2.4.6.8. Save File
2.4.6.9. Check Hmax
2.4.6.10. If Hmax <2000
2.4.6.10.1. Return to AZ0
2.4.6.10.2. Move EL by 1000 steps
2.4.6.10.3. Check EL
2.4.6.10.3.1. If <30 º repeat 2.4
2.4.6.10.3.2. If EL=40 º move to 2.5
2.4.6.11. Else move AZ -2000steps
2.4.6.12. Repeat 2.4.6.1
2.4.7. Else move AZ 2000 steps
2.4.8. If AZ=0 º
2.4.8.1. Move El by 1000 steps
2.4.9.Else repeat 2.4

2.5. <u>If 40≤EL<100</u>

2.5.1. Auto-Focus
2.5.2. Auto-Center
2.5.2.1. Save X and Y displacement
2.5.3.Auto-Focus
2.5.4.Take average snapshot
2.5.5.Check Hmax
2.5.5.1. If Hmax<2000
2.5.5.2. Return the number of AZ steps plus -1000
2.5.5.3. Apply X and Y displacement
2.5.5.4. Auto-Focus
2.5.5.5. Auto-Center
2.5.5.6. Auto-Focus
2.5.5.7. Take average snapshot
2.5.5.8. Save file
2.5.5.9. Check Hmax
2.5.5.10. If Hmax<2000
2.5.5.10.1. Return to AZ0 º
2.5.5.10.2. Move EL by 1000 steps
2.5.5.10.3. Check EL<1000
2.5.5.10.3.1. If EL=90 º stop scan
2.5.5.10.3.2. If EL<100 º repeat 2.5
2.5.5.11. Else move AZ -1000steps

2.5.5.12. Repeat 2.5.5.1
2.5.6. Else Save file
2.5.7. Move Az 1000 steps
2.5.8. If AZ=0 º
2.5.8.1. Move EL 1000 steps
2.5.9.Check EL<100 º
2.5.10. Stop can if EL=90 º
2.5.11. Else repeat 2.5

7.2. Voronoi Correlation

After the picture has been obtained, the next step in the research was to test a novel approach to image correlation based on the Voronoi diagram. In the image processing field [21] proposed an approach to correlate images using Voronoi regions. However, this model is highly constrained and not flexible as labeled regions are correlate with same labeled region from the objective image. Therefore, the novel approach proposed in this research is a flexible approach where one region can be compared with all the regions of the objective image in order to find the best output. The procedure tested consists of three scripts, and they will be explained as follows:

1. The first script in the procedure is called "PolygonExtraction". The aim is to build the Voronoi diagram by inputting the facets centroids as seeds. The code works as a for loop where the set of images saved from the scanning are read. The script operates as follows. First, the pictures need to be rotated, to align the different AZ pictures with the reference, which is AZ0. For this, a correction factor as a consequence of the rotation is applied and it is defined as "beta (β)". The correction factor takes into account the amount of rotation in azimuth and translate it through multiplying it to the cos(EL*), please refer to appendix D.* Then, after rotation, the pictured is cropped and filtered to the blue channel. Within the function "FacetsCentroidscat" is called, *please refer to the appendix.* The later function applies a threshold to the image where the facets are identified and by applying "regionprops" the vector X*2 of facets is identified. Such vector X*2 is the input of the function built within MATLAB called "Voronoin" which builds a Voronoi diagram, see Figure 24.

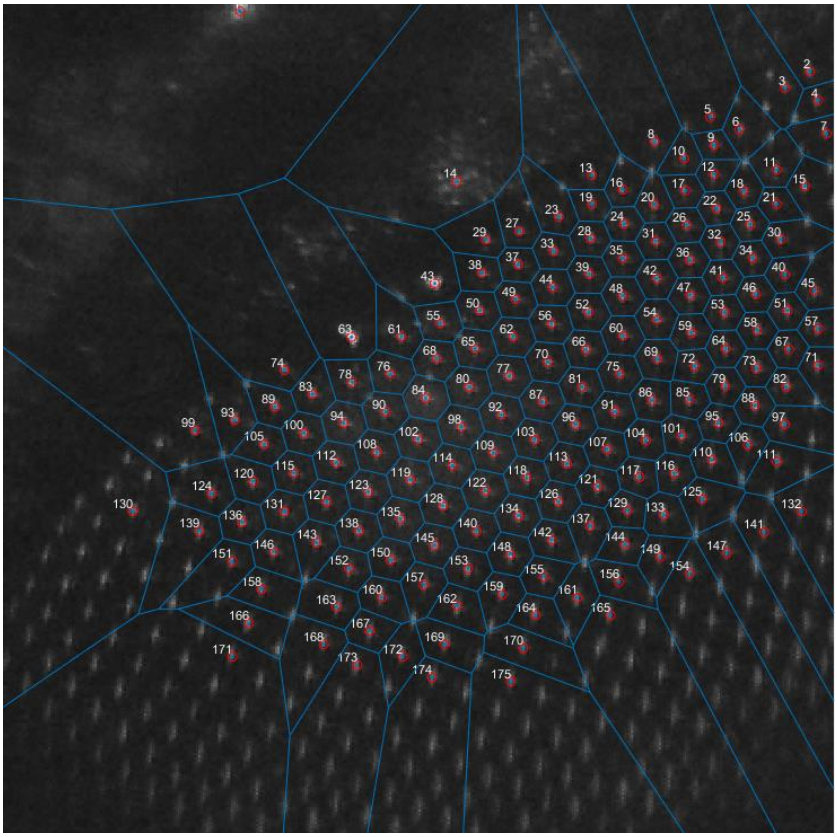
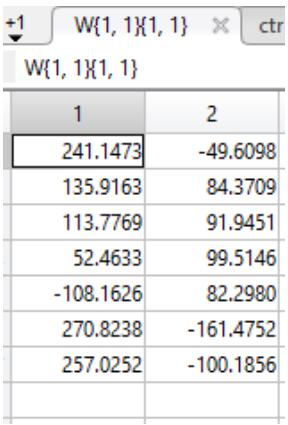


Figure 24. Voronoi diagram built from the facets centroid. Picture taken at EL40AZ-20

The regions identified are labeled for future reference for results validation. Once the diagram is built, important data to save are the vertices of each polygon. The MATLAB built-in function “voronoin” is capable of obtaining the vertices; however, in order to arrange them so it can proportionate easy data to handle, they are arrange following the connection points. The vertices coordinates are yield in a vector Y*2 and the connection points are given in a cell of X*2, which match the length of the facets matrix. See figure 25.



W{1, 1}{1, 1}

1	2
241.1473	-49.6098
135.9163	84.3709
113.7769	91.9451
52.4633	99.5146
-108.1626	82.2980
270.8238	-161.4752
257.0252	-100.1856

Figure 25. Represent the vertices of a Voronoi region. Labeling is an advantage as it can be identified that this region is for the first processed pictured linking the labeled facet 1.

2. The next step is to identify the intensity matrix for each matrix. This matrix would have the polygon shape. The script implemented for such task is called “GettingRegions” please refer to the appendix. The script built a polygon on top of the image and binarize it. Then,

with the output matrix of 1 and 0 an overlapping procedure is done with the original matrix and the outcome is a perfect polygon-shaped matrix with intensities values inside. See figure 26.

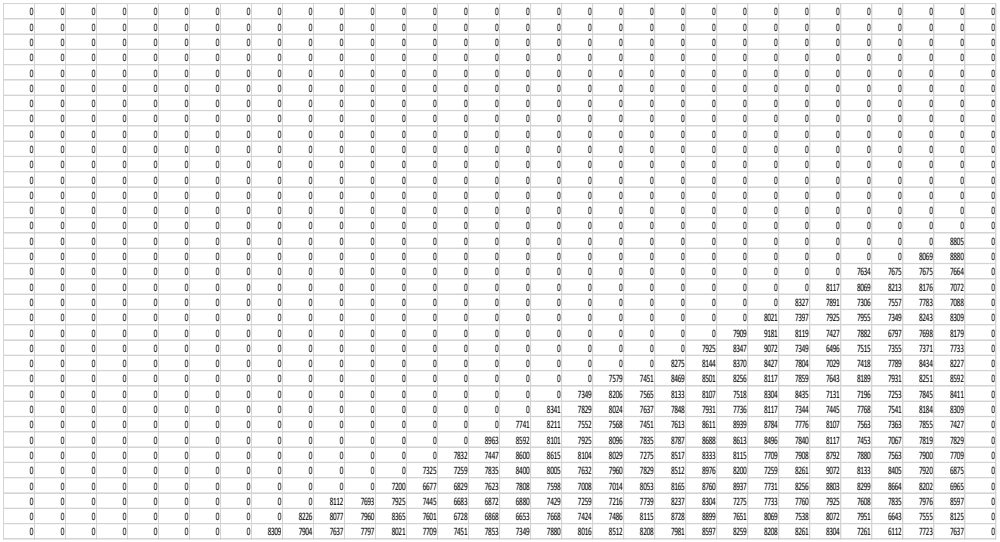


Figure 26. Region matrix with intensities.

3. The third and last step implemented in the research is the correlation of regions. The correlation is a new approach proposed, which aims to correlate the X amount of region in image 1 (R1.X) with the Y amount of regions in image 2 (R2.Y). The script implement is called *"ZNCCF" please refer to the appendix.* This script is based in the following mathematical model for ZNCC:

   A region is defined by $R_i^k(N_i^k, M_i^k)$ where *k* indicates the number identification of the target image (left(k=g) or right(k=d)), and *i*, indicates the n number of seeds of the Voronoi region. R characteristics are defined by the number of pixels M_i^k and its intensity shape M_i^k.

   To define the matrix M_i^k the Voronoi theorem will be introduced as it will set the boundaries for the matrix shape. Let S=$\{q_1^k q_1^k q_1^k, \dots, q_n^k\}$ be a set of points of the image *k*. These points are "called" seeds and are the detected facets centroid. The Voronoi polygons associated with the point q_n^k, is the set of points closest to the image plane as other points of S [22]. This polygon is defined in Eq. (3).

$$P(q_i^k) = \left\{ q \in I^k \; \exists \; d\left((q, q_i^k) < d(q, q_j^k) \right) \right\} \quad (3)$$
$$for\ j \ \in \{1,2, \dots, n\} \backslash \{i\}$$

   Then the M_i^k can be defined as shown in Eq. (4).

$$M_i^k = \left(a_{pq}^{ki}\right)_{\substack{1 \le p \le h_i^k \\ 1 \le h \le h_i^k}} \quad (4)$$

   Where a represent the pixel inside the polygon vertices along the width and height of Image matrix. Then the correlation formulas are applied, which are the following ones:

$$Z1 = \sum_{p=1}^{h} \sum_{q=1}^{l} \left(a_{pq}^{gi} - \bar{A}^{gi}\right)\left(a_{pq}^{dj} - \bar{A}^{dj}\right) \quad (5)$$

$$where\ j = \{1,2,3 \dots, n\}\ and\ i\ stays\ constant$$

$$Z2 = \sum_{p=1}^{h}\sum_{q=1}^{l}\left(a_{pq}^{gi} - \bar{A}^{gi}\right)^2 \qquad (6)$$

$$Z3 = \sum_{p=1}^{h}\sum_{q=1}^{l}\left(a_{pq}^{dj} - \bar{A}^{dj}\right)^2 \qquad (7)$$

$$\bar{A}^{gi} = \frac{\sum_{p=1}^{h}\sum_{q=1}^{l} a_{pq}^{gi}}{Npi} \qquad \bar{A}^{dj} = \frac{\sum_{p=1}^{h}\sum_{q=1}^{l} a_{pq}^{dj}}{Npj} \qquad (8)$$

$$ZNCC\left(R_i^g, R_j^d\right) = \frac{Z1}{\sqrt{Z2 * Z3}} \qquad (9)$$

8. Results

8.1. Scanning time, Quality, Automation

Two sets of scans from EL0 to EL180 were performed successfully. Please refer to appendix section scan results. It is concluded that the lower area of the eye (EL90-EL180) requires user supervision. This area contains the mouth of the animal, which the obvious movement of such, can cause noise when the pseudopupil is lost, which impacts in highest intensity available. Therefore, the algorithm identifies the mouth as a pseudopupil instead of applying step in section 7-2.4.6.1 or 2.5.5.1. When this phenomenon occurs, the user stops the scan and initializes it again at the following step of the stopping point.

Even though the lower part is a challenge for the algorithm. Scanning time has been improved considerably. On both occasions, the scan time was below six hours. Please refer to Table 3 for more detailed information. Moreover, the code, once the mouth is out of the ROI, performs as expected. Therefore, a fully automatized path that does not require the intervention of the user, acquired approximately the 323 images files of the eyes. Refer to appendix B for the code implemented. In addition, it is noticeable that the number of images per scan varies depending on the natural morphology of the eye, quality of the pseudopupil and position of the fly on the pin.

Table 3. Results regarding time and number of files of the proposed algorithm. The decimals values in seconds are depreciable. The total scanning time takes into account the automatic scan from hitting start until it automatically stops

Test	*1*	*2*	*Average*
Number of images	*334*	*313*	*323*
Average number of files for EL line below 40	*16*	*13*	*14,5*
Average number of files for EL line above 40	*20*	*23*	*21,5*
Average time per image (s)	*40 +/- 10*	*40 +/- 10*	*40 +/-10*
Average time per EL line below 40 (s)	*640 +/- 160*	*520 +/- 130*	*580+/- 145*
Average time per EL line above 40 (s)	*800 +/- 200*	*920 +/- 230*	*860 +/- 215*
Below EL 30 rotational interval (◦)	*20*	*20*	
Above 30 rotational interval (◦)	*10*	*10*	
Total scanning time (hours)	*3.74 hours*	*4 hours*	*3,87*

Image quality was another aspect to be validated. It was verified through the scans that the algorithm was able to maintain the pseudopupil at the center of the ROI. Figure 27,28 and 29 represents the EL30 lines of the first test, where it can easily be verified the quality of the images.

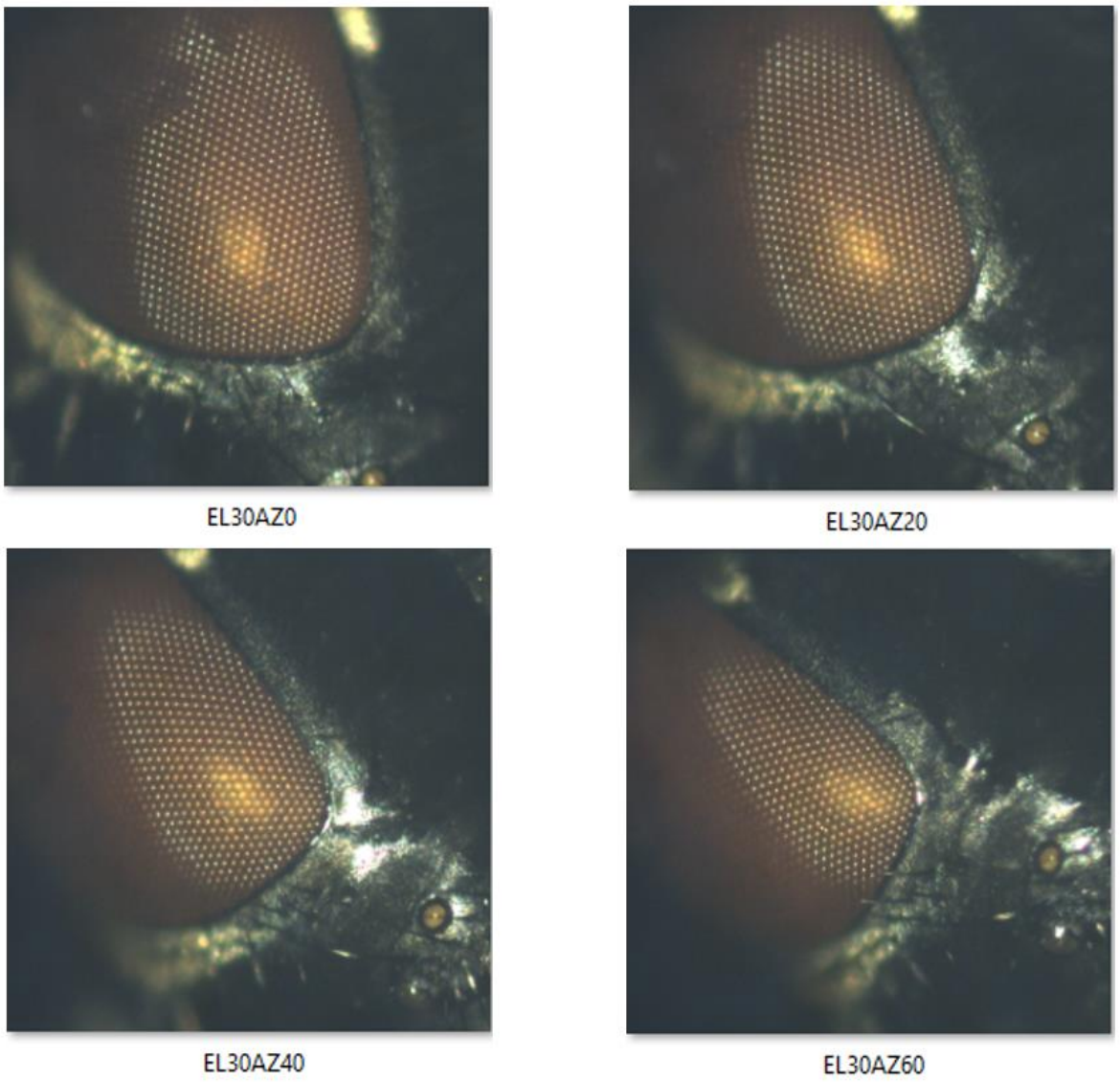
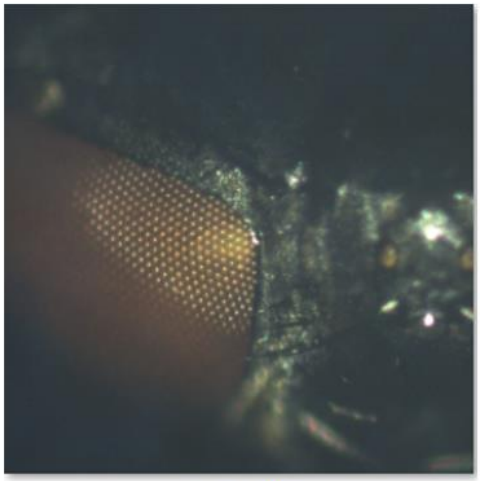
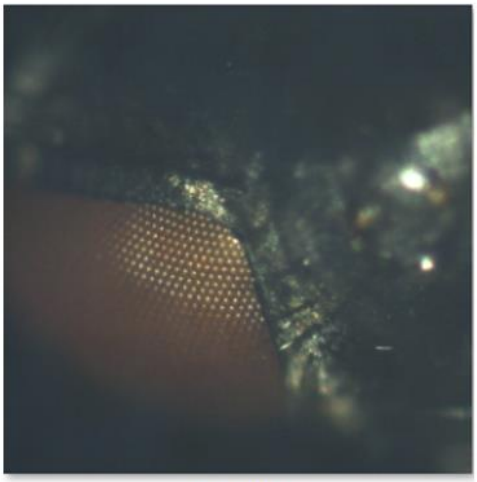


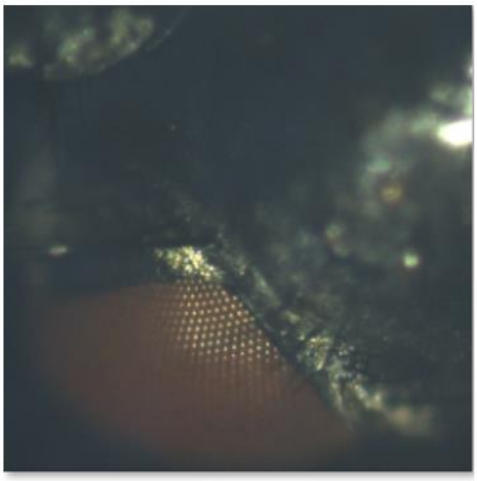
Figure 27. Set of images for the line of EL30



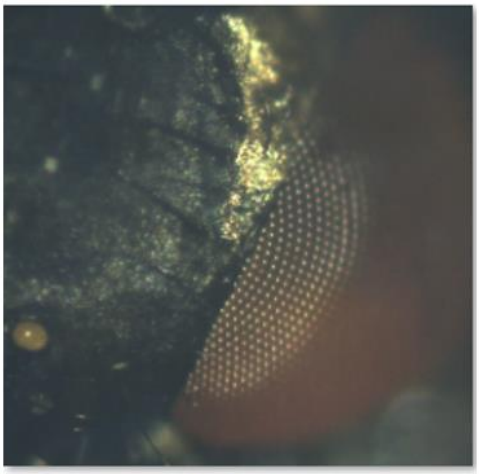
EL30AZ80



EL30AZ100



EL30AZ120



EL30AZ240

Figure 28. Set of images for the line of EL30

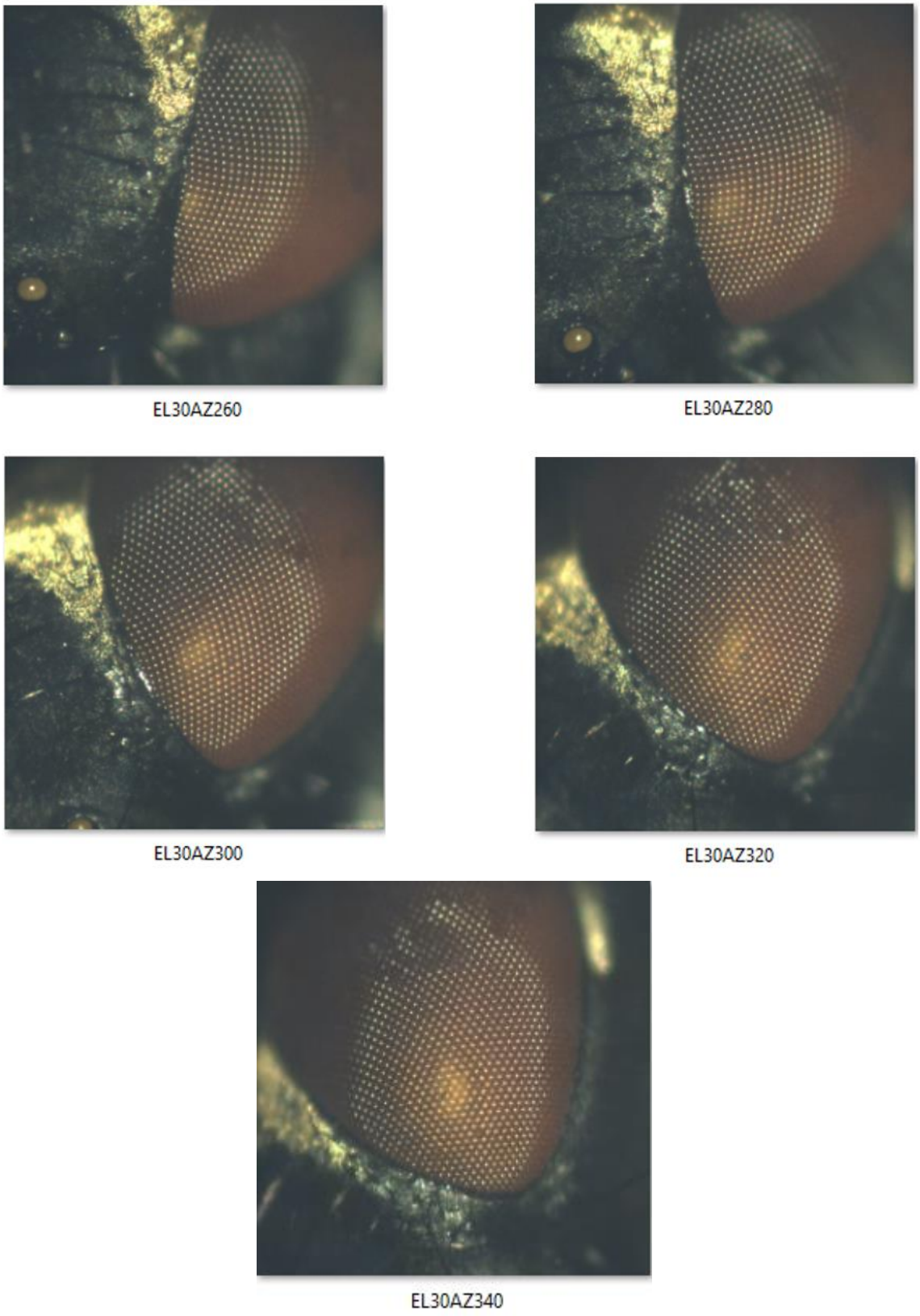


Figure 29. Set of images for the line of EL30

Figure 27 illustrates how the algorithm is able to keep the house fly within the ROI while performing high quality snapshots. Moreover, the reader can visualize the cuticle border “effect” at EL30AZ120 and EL30AZ240.

8.2. Regions, Correlation, Distance Validation

By implementation of the procedure explained in section 7.2 a successful Voronoi diagram was built as seen in Figure 30. Moreover, regions R were possible to identified with their respective N and M matrix.

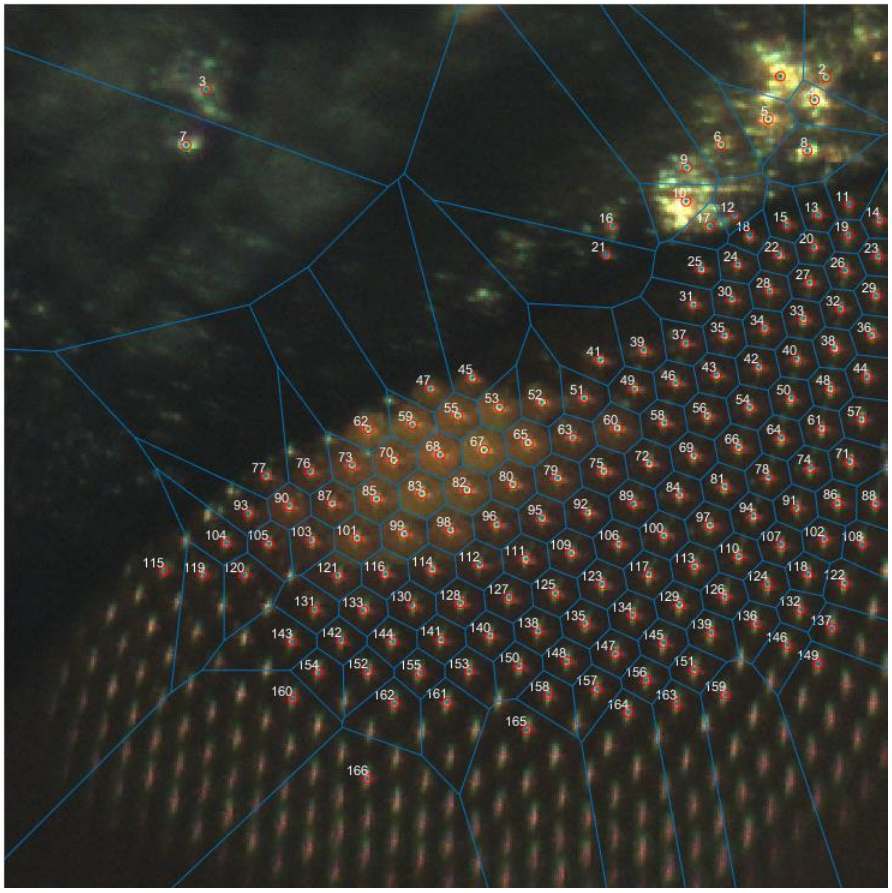


Figure 30. Voronoi diagram using facets centroid as seeds and labeled.

The algorithm responsible to find the M matrix, which represents the shape and amount of pixels inside a polygon worked as planned. Please refer to appendix F. Figure 31 and 32 illustrates the M matrix for region labeled as 100. It can clearly be seen that the shape of the matrix matches the shape of the polygon.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	8283	8299	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	8976	8728	8496	8467	8213	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	8904	9533	9334	9312	8337	7973	8308	8368	7717	0	0	0	0	0	0	0	0	0
0	0	0	0	0	9451	9107	9360	8776	9552	9008	8117	8651	9360	8573	8016	8165	0	0	0	0	0	0	0
0	0	0	0	10458	9285	9083	9475	9458	9573	9176	8672	8228	8157	8468	7965	6829	8296	8187	7640	0	0	0	0
0	0	0	9552	9275	8693	8987	9931	10136	9840	9536	9552	8208	7163	7325	7163	6949	7259	7168	7259	8024	0	0	0
0	0	8456	8835	9612	9979	10073	9888	10354	9941	10522	10637	8959	7752	7984	8101	8517	7221	7361	7763	7198	0	0	0
0	8688	8507	8400	10256	11461	10355	9456	9333	9835	11536	11461	10301	8496	8976	9264	8669	7829	7907	8016	6872	0	0	0
9462	8925	8082	8464	9863	10275	9618	9019	9335	8787	9744	11189	9195	8269	8978	9328	8811	8496	8584	8675	8191	0	0	0
9696	9835	10037	9648	8771	9840	10080	9840	10168	8789	7387	8213	8387	8400	9147	9360	8816	8304	9099	9648	8435	8501	0	0
9200	9797	9274	9640	8962	9427	10310	10936	9508	10392	9243	8283	8547	8091	7935	8560	8358	7979	8571	9051	8106	8096	0	0
0	8880	9149	9264	8597	7925	7693	8880	9552	9552	10171	9739	8963	8405	7893	7835	8184	7829	7760	8021	6973	6971	0	0
0	8624	9358	8851	8682	8800	8219	7973	8508	8229	8584	8797	8918	8621	8016	7613	8958	7883	7375	8493	7876	7445	0	0
0	9264	9237	8597	8453	9264	8461	8117	8368	8016	7789	7925	7560	7728	7992	8597	9019	8405	8440	8112	7485	7829	0	0
0	11728	9300	8120	8397	8331	8399	8459	8120	8691	8456	7904	7866	7261	7247	8307	7894	8123	9382	8491	7933	7885	7700	0
0	11653	10139	8693	8400	8117	7949	8309	8725	8688	8752	8117	7661	7733	7459	7451	8368	9072	9237	8789	8259	8208	0	0
0	0	9479	8891	8015	8307	8759	8389	8627	7499	7343	7933	8187	8077	7634	7035	9038	9264	8059	8235	8638	0	0	0
0	0	7829	8304	7899	7451	7579	8400	8051	7824	8125	7829	7813	7541	7296	7067	8984	9264	8256	0	0	0	0	0
0	0	8835	8456	7770	7467	8180	8421	8068	8747	7814	7261	7227	7381	7990	7152	7259	8101	0	0	0	0	0	0
0	0	0	0	0	8208	7987	7920	7728	8304	7613	7541	7699	7067	7403	8213	8221	0	0	0	0	0	0	0
0	0	0	0	0	0	0	7533	8290	8000	7945	8005	6940	6547	7119	8085	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	7275	7637	6981	6400	7267	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	6685	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 31. M matrix for R100.

Figure 32. Shape of R100

However, the implementation of ZNCC was unsuccessful, and various explanation can be derived for the behavior obtained by the algorithm. Correlations of 0.8-0.91 were possible to obtain. Nevertheless, the main issue comes by performing a flexible ZNCC. Table 4 represents the results for a set of two images where dead facets where present. When a dead facet is present it helps the user to validate the offset output of the correlation as a complete certainty of the movement in X and Y can be derived.

Table 4. Correlation offset performing ZNCC

	EL30AZ60	EL30AZ80
Number of regions	390	360
Max correlation value	0.8577	
Region correlated	239	206
Dead facet coordinates	(114, 190)	(124, 202)
Dead facet offset	(-10, -12)	
Region centroid coordinate	(267, 165)	(260,31)
Correlation offset	(7, 164)	
Time (sec)	378.0862	

On the contrary of (Laraqui, Saaidi, Mouhib, & Abarkan, 2015) work this research employed a flexible ZNCC. (Laraqui et al., 2015) correlated only regions with the same label while this novel work fixed one region in this case from EL30AZ60 and correlates them with the 360 regions of EL30AZ80. The process takes a considerable amount of time as it is based on a set of only two pictures (532.53 sec for two images). Moreover, the result to highlight is the difference between the analyzed offset of the dead facet with the ZNCC result. This discrepancy is due to the highly regular pattern of the eye facets. Therefore, multiple similar polygon exists in the pattern heavily impacting the reliability of this approach.

Unfortunately, this approach does not work under highly similar pictures. When the algorithm faces similar polygons pattern such as the case in a house fly, it is very likely that the correlation result is not adequate. Besides, the values of intensities are often similar as the facet shines are highly similar, these values heavily impact the average of the region as no discrepancy can be noted. Therefore, contributing to the miscorrelation.

9. Discussion

This section will evaluate the assessment in the validation criteria given in section 5. In Table 5 it is shown a summary of the milestones.

Table 5. Assessment criteria for the scanning process

	Criterion	*Description*	*Assessment*
1	*Scanning time*	*90 degrees of the house fly should be completed in less than 2 ½ hours. While a full 180 degrees should be completed under 6 hours.*	*very possible*
2	*Quality of the Images*	*Output pictures have to be sharp and centered. Centered refers to: the eye's pseudopupil have to mark the center of the picture*	*Yes*
3	*Automation*	*The complete procedure should only involve the user for visual inspection and intervention in case noise are present due to movement of the fly. The code should be able to move through lines of Az and El as well as identifying borders.*	*Very sufficient; however, lower part requires user supervision and in cases intervention*

Table 6 Assessment criteria for Voronoi correlation process

	Criterion	*Description*	*Assessment*
1	*Regions*	*The Voronoi Diagram have to be able to build regions generated from the facets centroid and label them. Moreover, pixel matrix per region have to be yield.*	*Yes*
2	*Correlation*	*The correlation output between regions have to be of 0.9+*	*Possible*
3	*Distance validation*	*Distance taken from reference points should match from the offset between regions.*	*Negative*

As analyzed in the previous section, the average scanning time of 18 elevation points in a set of points of EL= {0,10, 20, ….,90} is 3,87 hours. Therefore, the first validation criteria ***"Scanning time***" can easily be achieved. Moreover, the images are of high quality and properly centered, validating the second criteria ***"Quality of images"***. The third and last criteria for the scanning was the automation, during the experiment it was experienced in both occasions that the user needed to intervene at the area

where the mouth entered the ROI; therefore, the criteria is highly likely to be achieved; however, still some intervention is needed to complete the 180 degrees. (Niemeijer, 2018) obtained results for an optimal path in order to be able to capture the 180 degrees of the eye. However, the results show that user skill and time was a heavy constraint for the number of pictures to be taken per hour. Moreover, in reality the time was higher than suggested in the research. This research results provide to the University of Groningen for the first time the capability of capturing a high number of images without facing time constrains.

Regarding the correlation algorithm validation criteria, works of (Laraqui et al., 2015) & (Pokojski & Pokojska, 2018) yield interesting results using Voronoi for image correlation and stitching. However, their work is highly constrained. One important aspect to point out is the input images, as they are images with clear discrepancies in contrast to the ones used on this research, which are highly similar. Clear discrepancies imply that it is easier to obtain a good correlation. Moreover, (Laraqui et al., 2015) proposed a set of constraints for the region characteristic that can be achieved only under such environment. The present research contradicts such works. First, when such algorithm is applied to highly similar photos as the ones obtained, none of those proposed algorithms work. Therefore, a flexible algorithm had to be developed. Nevertheless, the correlation output is still not reliable as the constant facet pattern creates several highly similar polygons. In addition, intensities often are similar at the same X and Y coordinates. Thus, resulting in a correlation close to the labeled polygon and far from the real one. This approach proves that any approach with Voronoi towards highly similar images should be avoided, as the algorithm needs clear discrepancies to build different polygons, which then would easily be correlated to work

10. Conclusion and recommendations

The work presented in this research answers both core questions of the research. Which were, "***How to develop an algorithm able to move around Az and El lines that the GRACE system would follow without the necessity for user intervention" and "How can the Voronoi theory enhance the correlation program?".*** First, a significant improvement was experienced by disposing the GUI from the MATLAB algorithm. In fact, by moving to a function controlling based of the GRACE system the user has more flexibility to program the "path of scan" at his best convenience while keeping time at an optimal level. Furthermore, the switch of the "Auto-center" approach to a dark and light adapted images has greatly improved the accuracy of such algorithm, which gives promising opportunities for future research in obtaining further information about the compound eye. Second, it was concluded from this research that a Voronoi approach to image correlating should be avoided. A Voronoi approach represents a high constraint algorithm which this research proves is not suitable for highly similar images. Through implementing a flexible ZNCC it was verified that indeed the algorithm cannot correlate properly polygon regions; consequently, yielding a wrong offset.

There are several recommendations that can be made for future research:

- The fly's placement on the pin should always be the same. It was noticed that the current method involves a high probability of human error when rotating the fly to scan the second part of the eye. Moreover, no clear identification of the position of the fly means that once the team needs to place another one most likely it will not be identical; therefore, the results can't be replicated. This can be seen in the discrepancy in number of pictures taken on both experiments.
- The proposed pin base for higher tracking of the fly's position can be seen in Figure 33.

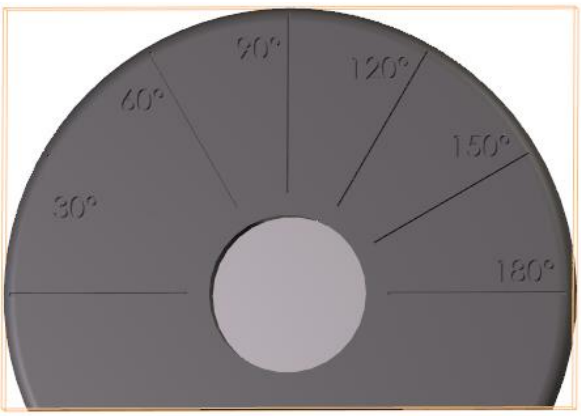


Figure 33. Proposed base pin for the experiment. Dimensions are 1.60mm*2.20mm*4mm

- Use a threaded pin to avoid the use of clay as it can cause human errors.

- By introducing the tracking of Z values in the scan the shape of the eye could be elaborated in a future research.
- It could be of interest for future image correlation research to apply the proposed algorithm on pictures with high discrepancy to evaluate the correlation accuracy

11. References

Bitsakos, K., & Fermüller, C. (2006). Depth estimation using the compound eye of dipteran flies. Biological Cybernetics, 95(5), 487-501.

Buschbeck, E. K., & Friedrich, M. (2008). Evolution of insect eyes: Tales of ancient heritage, deconstruction, reconstruction, remodeling, and recycling. Evolution: Education and Outreach, 1(4), 448-462.

Cheng, Y., Cao, J., Zhang, Y., & Hao, Q. (2019). Review of state-of-the-art artificial compound eye imaging systems. Bioinspiration & Biomimetics, 14(3), 031002.

Di Stefano, L., Mattoccia, S., & Tombari, F. (2005). ZNCC-based template matching using bounded partial correlation. Pattern Recognition Letters, 26(14), 2129-2134.

Dobrin, A. (2005). A review of properties and variations of voronoi diagrams. Whitman College, , 1949-3053.

Franceschini, N., & Wehner, R. (1972). Information processing in the visual systems of arthropods.

Kota, K., & Jonnalagadda, S. (2016). Image mosaicing using feature based technique. International Journal of Scientific & Engineering Research, 7(6), 1117--1125.

Land, M. F. (1997). Visual acuity in insects. Annual Review of Entomology, 42(1), 147-177.

Laraqui, M., Saaidi, A., Mouhib, A., & Abarkan, M. (2015). Images matching using voronoï regions propagation. 3D Research, 6(3), 27.

Niemeijer, R. (2018). Scanning path for rapid mapping of a compound eye with GRACE.

Nilsson, D., & Kelber, A. (2007). A functional analysis of compound eye evolution. Arthropod Structure & Development, 36(4), 373-385.

Pokojski, W., & Pokojska, P. (2018). Voronoi diagrams–inventor, method, applications. Polish Cartographical Review, 50(3), 141-150.

Rubin, D., Miserez, A., & Waite, J. (2010). Advances in insect physiology: Insect integument and color, eds C jérôme and JS stephen.

Sahu, K. K. (2017). Automated Image Feature Identification for Motorised Mapping of Insect Eyes,

Stavenga, D. G. (1979). Pseudopupils of compound eyes. Comparative physiology and evolution of vision in invertebrates (pp. 357-439) Springer.

Tomlinson, A. (2012). The origin of the drosophila subretinal pigment layer. Journal of Comparative Neurology, 520(12), 2676-2682.

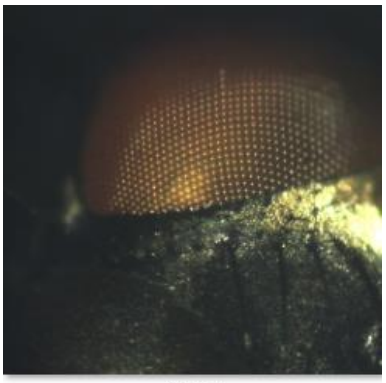
Verschuren, P., Doorewaard, H., & Mellion, M. (2010). Designing a research project Eleven International Publishing The Hague.

Viollet, S., Godiot, S., Leitel, R., Buss, W., Breugnon, P., Menouni, M., . . . l'Eplattenier, G. (2014). Hardware architecture and cutting-edge assembly process of a tiny curved compound eye. Sensors, 14(11), 21702-21721.

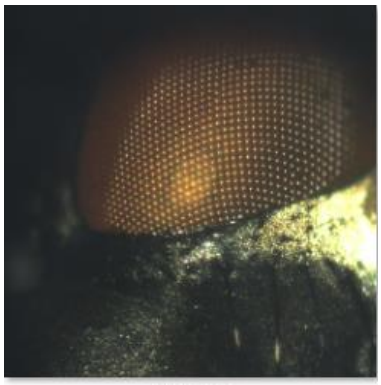
Wieringa, R. J. (2014). Design science methodology for information systems and software engineering Springer.

Appendix

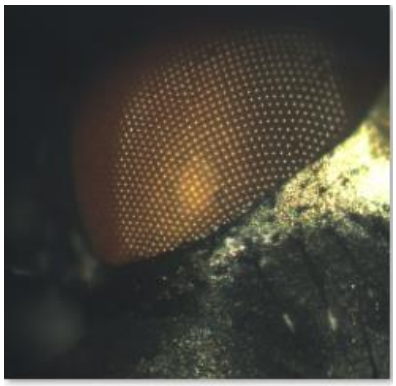
Scan results: Examples of two elevation lines obtained



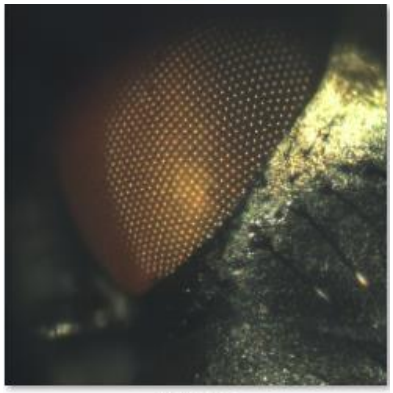
EL0AZ0



EL10AZ0

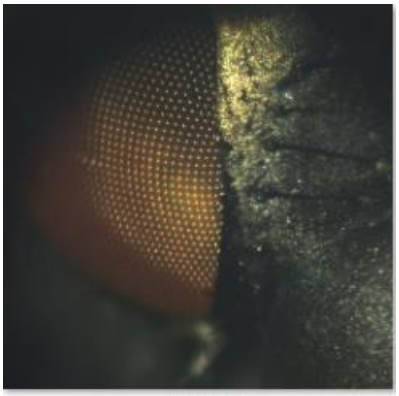


EL10AZ20

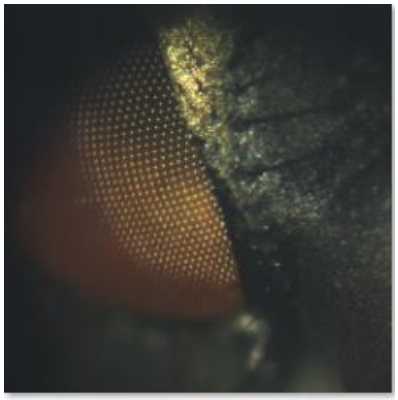


EL10AZ40

EL10AZ60



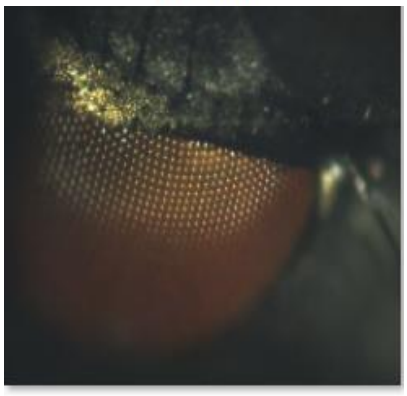
EL10AZ80



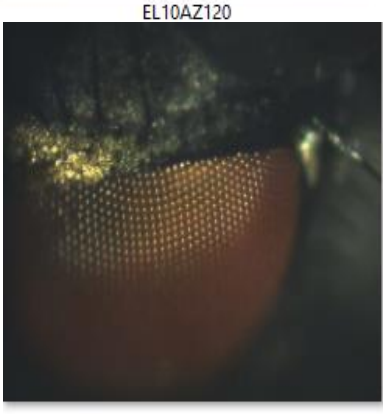
EL10AZ100

EL10AZ120

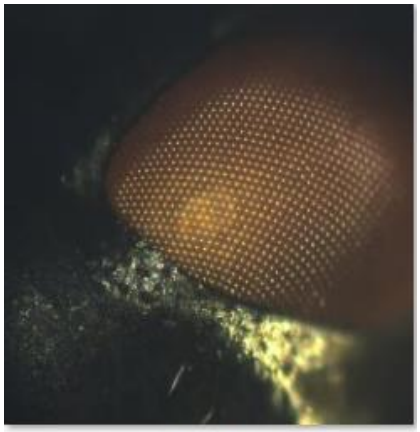
EL10AZ140



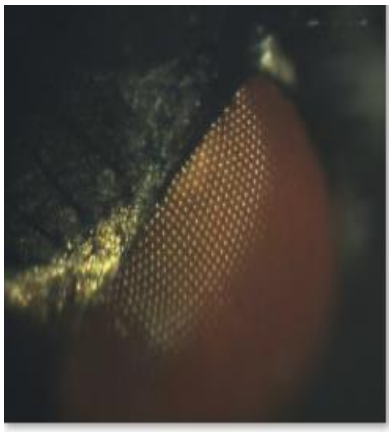
EL10AZ160



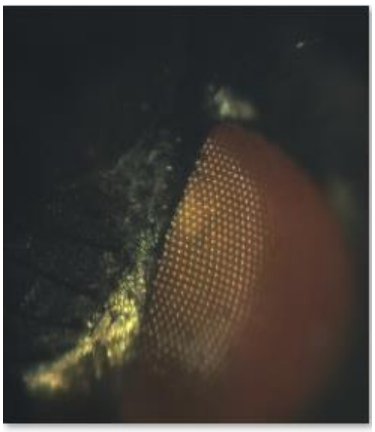
EL10AZ180



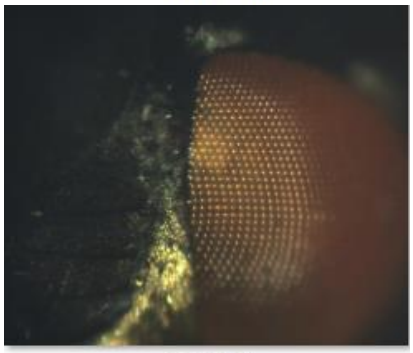
EL20AZ-40



EL10AZ220

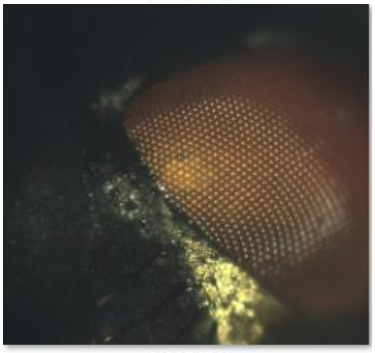


EL10AZ240

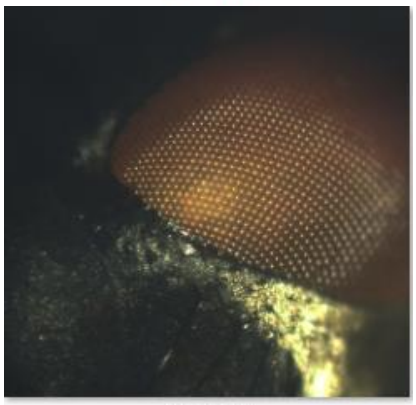


EL10AZ260

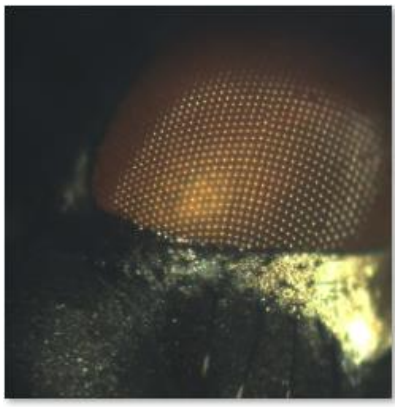
EL10AZ280



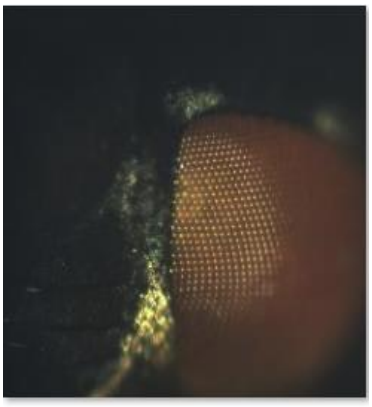
EL10AZ300



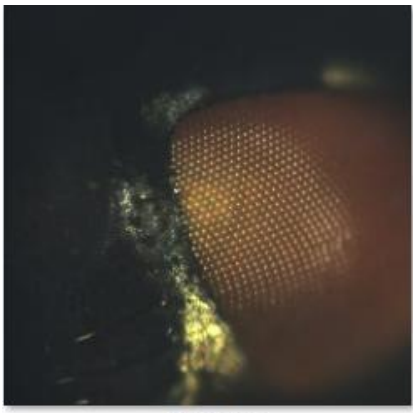
EL10AZ320



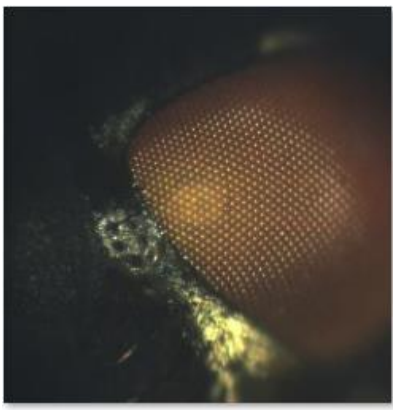
EL10AZ340



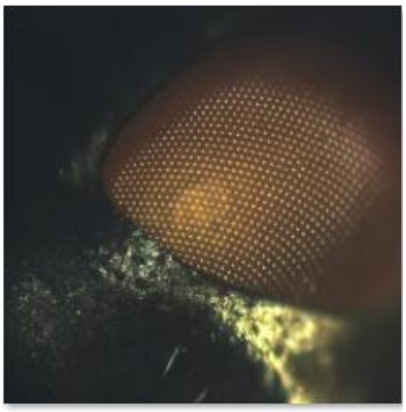
EL20AZ-100



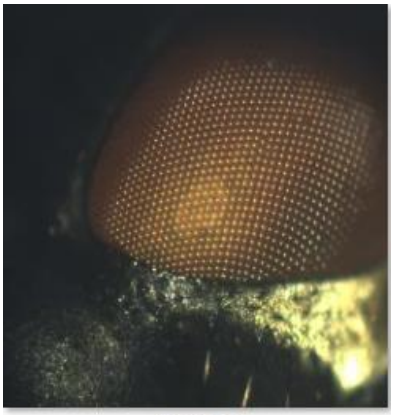
EL20AZ-80



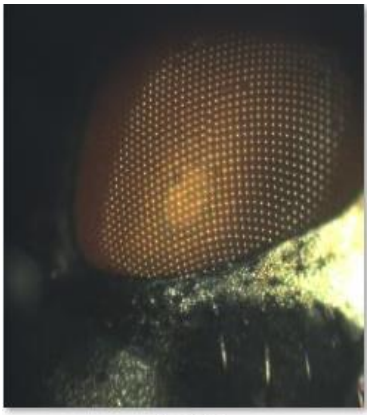
EL20AZ-60



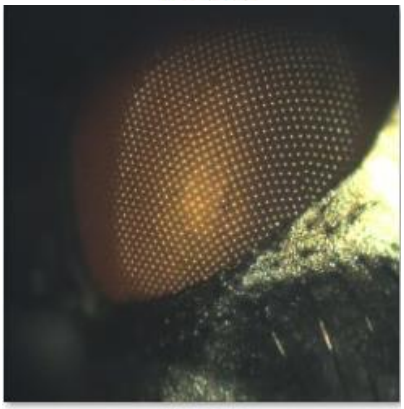
EL20AZ-40



EL20AZ-20



EL20AZ0



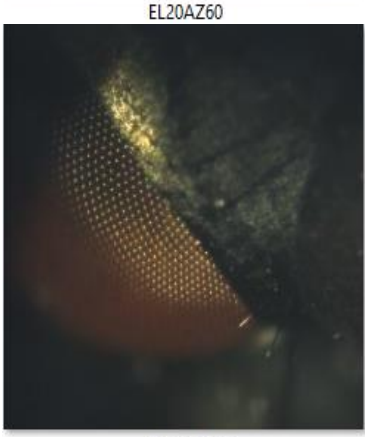
EL20AZ20

EL20AZ40

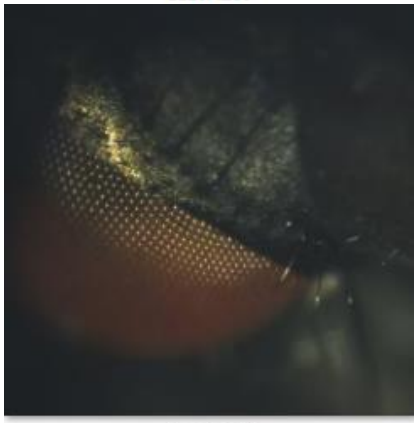
EL20AZ60

EL20AZ80

EL20AZ100



EL20AZ120



EL20AZ140

MATLAB codes

A. Initialization

```
 %% Setting the Path
clc
close all
clear

addpath(genpath('C:\Users\P155002\Documents\Scanning2020'));
cd('C:\Users\P155002\Documents\Scanning2020');


disp('End of section Setting the Path')
% Initializing STANDA Motor Controllers

StandaLibraryOpenJd
%
D=100; % Dsteps=1degree
MovingMotorsTest(1,D) %Y
MovingMotorsTest(1,-D) %Y

MovingMotorsTest(2,D) %Z
MovingMotorsTest(2,-D) %Z

MovingMotorsTest(3,D) %E
MovingMotorsTest(3,-D) %E

MovingMotorsTest(4,D) %None
MovingMotorsTest(4,-D) %None

MovingMotorsTest(5,D) %X
MovingMotorsTest(5,-D) %X

MovingMotorsTest(6,D) %A
MovingMotorsTest(6,-D) %A

%To take the current position of a motor
%position_status1 = struct();
%state_position1 = ximc_get_position(1,position_status1)

%clc
disp('End of section Motors')


% INITIALIZE CAMERA
%%
%Initialization of camera
%close the video if it exists

if exist('vid','var')
   fprintf('closing previous video object');
```

```
    %stop(vid);
    %close(vid);
    delete(vid);
    clear ('vid');
end
close all
%clc
disp('End of section Close Camera')

global vid;

VideoFormat = 'F7_BayerRG16_1920x1200_Mode0';
vid = videoinput('pointgrey',1, VideoFormat);
VidBits = 16; % Caution: could be 8 if a different VideoFormat is used.
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
src.WhiteBalanceRBMode  = 'Manual';
src.WhiteBalanceRB = [550 810];
src.Gain = 29.9964;  OFF.
src.Shutter = 12;
src.Exposure = -1.778;
src.Brightness = 2.8076;
CurrGain = src.Gain;
CurrShutter = src.Shutter;
src.FrameRate = 30;

%clc
disp('End of section Initialize Camera')
%%

% Initializing ARDUINO
if exist('mega','var')
    fclose(mega);
    delete (mega);
    clear ('mega');
end


global mega
mega = serial('COM9'); %Create serial port object Windows, check your port
set(mega,'Tag','SerialObj');

%leo = serial('/dev/ttyACM18'); %Create serial port object on Linux - CR Test
set(mega,'DataBits',8);
set(mega,'StopBits',1);
set(mega,'BaudRate',9600); %Has to match the value specified in Arduino.
set(mega,'Parity','none');
set(mega,'Timeout',60);%time waits
mega.BytesAvailableFcnMode = 'terminator';%interrupt callback
mega.BytesAvailableFcn = @instrcallback;%display a message to the commandline when
fopen(mega); %Open the connection with the Arduino.
```

```
disp('End of section Setting ARDUINO')

%% Controlling the LED Intensity


frame = getsnapshot(vid);


mframe = max(max(frame(:,:,1)))

%%
Value = 255;

Value=round(Value);
Value=int2str(Value);
Command=['<','2',',',Value,'>'];
fprintf(mega,Command);



%%
% Procedure to set a ROI based on Ginput
%New ROI
clear X Y
frame = getsnapshot(vid);
imshow(frame)
[X, Y] = ginput(1);
% vid.ROIPosition = [0 0 1920 1200];
dist=350;
XOffset =  X-dist; %For 10X then it is +-250 and and W and H of 500.
              %For  5X then it is +-150 and and W and H of 300.
YOffset =  Y-dist;

WidthX  = dist*2;
HeightY = dist*2;
vid.ROIPosition = [XOffset YOffset WidthX HeightY];
close all
clear XOffset YOffset
disp('End of section Setting ROI')

%Starting the video

%%
start(vid); % puts vid object in "running" state.
preview(vid); % puts the full vid image into the figure if it fits.

%%
Value = 255;

Value=round(Value);
Value=int2str(Value);
```

```
Command=['<','2',',',Value,'>'];
fprintf(mega,Command);

% Scannning20200209

close all
```

B. Scannin2020

```
dirname='C:\Users\P155002\Documents\Scanning2020\ScanData\';
% if exist(dirname,'dir') == 0
% mkdir(dirname);
% else
% end

%%
% create folders for measured images
% kmax = 10;
% for k=1:kmax
%    EL=10*(k-1)
%    dirNew=['EL',num2str(EL)];
%    if exist([dirname,'\',dirNew],'dir') == 0
%       S=sprintf('elevation is %2d ',EL);
%       disp(S)
%       disp([dirname,'\',dirNew])
%    else
%       direx=[dirname,'\',dirNew] ;
%    [SUCCESS]= mkdir(direx );
%    end
%    pause
%  end
%%
% motor numbering
kmax = 9
Y_Axis_ID = 1;
Z_Axis_ID = 2;
E_Axis_ID = 3;
X_Axis_ID = 5;
A_Axis_ID = 6;
D = 100; % D = 100 steps is 1 degree for elevation and azimuth
ELstep = 10*D; % elevation is 10 degrees
%% for 0
for K=1:1
   el=0;
AF(Z_Axis_ID,vid);
        [dsx1,dsy1, Hmax] = AC(vid,X_Axis_ID, Y_Axis_ID);
        AF(Z_Axis_ID,vid);
        frame = AverageSnapGRACEOriginal(vid,3);
        Scanned(:,:,:,K) = frame;
        %state_position2 = ximc_get_position(2,position_status);
        %G=state_position2.Position;
```

```
        dirname1=[dirname,'EL',num2str(el),'\'];
    fname=['EL',num2str(el),'AZ',num2str(0),'.tif'];
    fullFileName = fullfile(dirname1,fname); %filename
    imwrite(frame, fullFileName);
        MovingMotorsTest(E_Axis_ID,StepEL);
end
%% 10-40

%%
for k=8:kmax     % elevation parameter
  EL=(k)*10;  % elevation EL = 0, 10, 20, ...
%    if EL == 0
%       Az=0
%       jmax = 1; % maximum value azimuth parameter
  if EL<40
    jmax = 18;
    Az=20;
    AZstep = 20*D; % rotation step is 20 degree
    PV=zeros(18,3);
  else
    jmax = 36;
    Az=10;
    AZstep = 10*D; % rotation step is 10 degree
    PV=zeros(36,3);
  end
  for j=1:jmax  % azimuth parameter
    AF(Z_Axis_ID,vid); % autofocus
    [dsx1,dsy1, Hmax] = AC(vid,X_Axis_ID, Y_Axis_ID); % autocentet
    AF(Z_Axis_ID,vid); % another autofocus
    frame = AverageSnapGRACEOriginal(vid,3); % average of three images
    AZ = (j-1)*Az; % angular position of azimuth
    PV(j,1)=dsx1;
    PV(j,2)=dsy1;
    PV(j,3)=AZ;
    dirname1=[dirname,'EL',num2str(EL),'\'];
    fname=['EL',num2str(EL),'AZ',num2str(AZ),'.tif'];
    fullFileName = fullfile(dirname1,fname); %filename
    imwrite(frame, fullFileName);
    MovingMotorsTest(A_Axis_ID,AZstep);
    disp(Hmax) % maximum value of difference of channel 3: LA-DA
    if Hmax<1200
      % j is number of steps made so far
      MovingMotorsTest(A_Axis_ID,-(j+1)*AZstep);
      for o=1:j
      dsx=sum(PV(o,1))
      dsy=sum(PV(o,2))
      MovingMotorsTest(Y_Axis_ID,-dsy);
      MovingMotorsTest(X_Axis_ID,-dsx);
      end
      % azimuth returns to (zero minus one) position
      for h=1:jmax
```

```
            AF(Z_Axis_ID,vid); % autofocus
            [dsx1,dsy1, Hmax] = AC(vid,X_Axis_ID, Y_Axis_ID);
            if Hmax<2000
               break
            end
            AF(Z_Axis_ID,vid); % another autofocus
            AZ = -h*Az; % angular position of azimuth
            frame = AverageSnapGRACEOriginal(vid,3);
            dirname1=[dirname,'EL',num2str(EL),'\'];
            fname=['EL',num2str(EL),'AZ',num2str(AZ),'.tif'];
            fullFileName = fullfile(dirname1,fname); %filename
            imwrite(frame, fullFileName);
            MovingMotorsTest(A_Axis_ID,-AZstep);
         end
         MovingMotorsTest(A_Axis_ID,h*AZstep);
         break
      end
   end
MovingMotorsTest(E_Axis_ID,ELstep);
end
```

C. Auto-Centering

```
function [dsx1,dsy1,Hmax] = AC(vid,X_Axis_ID, Y_Axis_ID)

% Focuses "down" into eye, finds the PP and centers it in the image by adjusting x,y motors.
% This function focuses down, centers the PP.  Does not save a photo or focus back up again.

% Center_DPP: Centering the image in the DPP.
%-------------------------------------------------------------------------
% - Input: vid: video.
% - Output: - dsx: Proportionality constant.
%         - dsy: Proportionality constant.
%-------------------------------------------------------------------------

StepsPerPix = 0.523;

FindPP; % nested function to ID the PP and find its coordinates.

% center the PP

disp(['Now centering the PP.  dx, dy =' num2str(dsx1) ', ' num2str(dsy1)]);

MovingMotorsTest(Y_Axis_ID,dsy1);%  (Y direction) is moved dsy steps.
pause(0.2);
MovingMotorsTest(X_Axis_ID,dsx1);%  (X direction) is moved dsx steps.

% now we find the PP again & display to see if it's well centered:


% ============================================================
```

```
% ==================== Nested function(s) ======================
% ==============================================================

   function [] = FindPP
      % For finding the PP and its displacement from image center.
      % turning off the light

      mega1 = instrfind('Tag','SerialObj');

      Value = 0;
      Value=round(Value);
      Value=int2str(Value);
       Command=['<','2',',',Value,'>'];
      fprintf ( mega1 ,Command);
      %turning on the light
      pause(10)
      Value = 255;
      Value=round(Value);
      Value=int2str(Value);
      Command=['<','2',',',Value,'>'];
      fprintf(mega1,Command);
      pause(0.1)
      IM = AverageSnapGRACEOriginal(vid,3);% Immediately return single image f from vid
      pause(10)
      %IMA=getsnapshot(vid);
      IMA=AverageSnapGRACEOriginal(vid,3);
       f1=figure;
      set(f1,'Position',[10 100 900 400]);

subplot(131);imshow(IM(:,:,1));subplot(132);imshow(IMA(:,:,1));subplot(133);imshow(10*(IMA(:,:,1)-
IM(:,:,1)))

      %difference dark adapted and light addapted
      H=IMA(:,:,1)-IM(:,:,1);
      f2=figure;
      set(f2,'Position',[10 600 900 400]);
      subplot(1,2,1);   imshow(5*H)
      %isolate pseudopupupil by gaussian filtering
      H=LowpassImGauss(H);

      Hmax=max(max(H))

      %H=imgaussfilt(H,5);
      J=(H/Hmax > 0.7);
      subplot(1,2,2);imshow(J)
      hold on

      if Hmax >= 2000
         %center of pseudopupupil PP
         stat=regionprops(J,'centroid');
         x=[stat.Centroid];
```

```
        plot(x(1),x(2),'*b')
        %y=stat.Centroid(2)
        f3=figure;
        set(f3,'Position',[1500 100 200 200]);
        imshow(IMA)
        hold on
        plot(x(1),x(2),'*b')

        PP=[x(1) x(2)];
        %Image center coordinates


        cx = size(IMA,1)/2;
        cy = size(IMA,2)/2;
        ImCtr = [cx, cy];

        sfx=size(IMA,2); % x is 2nd
        sfy=size(IMA,1);



    dpx = sfx/2 - x(1); % CenterX  - PPx
    dpy = x(2) -(sfy/2); % PPy - center y


    dsx1 = -StepsPerPix*dpx; % dsx1 = number of steps to move dpx pixels.
    dsy1 = -StepsPerPix*dpy;

    disp (['dsx1, dsy1 = ' num2str(dsx1) ',  ' num2str(dsy1)]);
    %pause(0.3);

    else
       dsx1 = 0;
       dsy1 = 0;
       display('works!')

  end

end
end
```

D. <u>Polygon Extraction</u>

```
clear
El=30; % value elevation
Az=60; % first values of azimuth
Az2=80; %last value of azimuth
[Az]=60:20:80;
Azstep=20; %Calculating Azimuth step

dirname='E:\MATLAB\codigomanual\Steof5\EL30';
```

```
dirname2='E:\MATLAB\codigomanual\croppedfiles';
filename1= dir(fullfile(dirname,'*.tif'));

Tfilename1=numel(filename1); % read how many photos are contained in the filed
imedge=400;
for S=1:Tfilename1
 filename3=['EL',num2str(El),'AZ',num2str(Az(S)),'.tif'];
 F=fullfile(dirname,filename3);
 I=imread(F);
 beta=cosd(El)*Az(S);
 IR=imrotate(I,-beta);
 [rows, columns, ~] = size(IR);
 mid=[rows/2,columns/2];
 x0=mid(1)-(imedge/2)+0.5; % x of origin of cropped image
 y0=mid(2)-(imedge/2)+0.5; % y of origin of cropped image
 bigbox=[x0 y0 imedge-1 imedge-1];
 IMcrop= imcrop(IR,bigbox);
 fname = ['EL', num2str(El),'AZ',num2str(Az(S)) '.tif'];
 fullFileName = fullfile(dirname2,fname); %filename
 imwrite(IMcrop, fullFileName);
 Ifilt=IMcrop(:,:,3);
 Isave{S}=Ifilt;
 figure (S);
 imshow(I)
 hold on
 FP=FacetCentroidscat(Ifilt);
 F1= FP(:,1);
 F2= FP(:,2);
 Fac{S}=[F1 F2];
 plot(FP(:,1),FP(:,2),'or');
 [sortedY, sortIndeY] = sort(F2);
 sortedX = F1(sortIndeY);
 U{S}=[sortedX, sortedY];
 [g]=find(U{S}(:,1));
 labelpoints(U{S}(:,1),U{S}(:,2),g,'FontSize', 8, 'Color','w' )
 hold on
 voronoi(U{S}(:,1), U{S}(:,2));
 [vx, c]=voronoin(U{S});
 vx(any(isinf(vx),2),:) = [];
 V{S}=vx;
 C{S}=c;
 H1=numel(C{S});
 ct=C{S};
 for T=1:H1
    ct1=ct{T};
    ct2=ct1-1;
    ct2(ct2==0)=[];
     ctr{T}=[ct2];
 end
 ctr1{S}=ctr;
 clear ct1 ct2 ctr
```

```
end
  %% Calculating corners
 for i=1:Tfilename1
FacX=Fac{i};
numF = numel(FacX(:,1));
%A=zeros(numF,1);
Vr=V{i};
%Vr(Vr<0)=0;
   Cr=ctr1{i};
 for h=1:numF

     numC=numel(Cr{h});
     X=zeros(numC,1);
     Y=zeros(numC,1);
     for j=1:numC

         p=Cr{h}(j);
         X(j)=Vr(p,1);
         Y(j)=Vr(p,2);
     end
   Corner{h}=[X Y];
   A{h}=polyarea(X,Y);
 end

 W{i} = Corner;
 A1{i}=A;
 clear X Y p numC numF Corner A
 end
```

E. Facets Centroids

```
function [FP,I_Thresholded]=FacetCentroidscat(Ifilt)


% function to determine centroids of facets
I_Thresholded=(Ifilt/max(max(Ifilt)) > 0.6);


SE = strel('diamond',2);

IT_Dilated = imdilate(I_Thresholded,SE);

stat=regionprops(IT_Dilated,'centroid');
nums=numel(stat);
FP=zeros(nums,2);
for i=1:nums
    FP(i,:)=[stat(i).Centroid(1) stat(i).Centroid(2)];
end

% Show image and facet centroids
plot(FP(:,1),FP(:,2),'or');
```

```
% hold on
%FP = round(FP);

pause(0.01) %in seconds
end
```

F. Getting Regions

```
W=importdata('Corners.mat') ; %import previous calculated
Ifilt1 =importdata('filteredphotos.mat') ;
Q=importdata('Facetscentroid.mat') ;
numW=numel(W);
im=600; %check is the same values as first algorithm
m=(1+im)/2; %middle of cropped image
ims=im/2; % size central area
hims=ims/2;
x1=m-hims+0.5;
y1=m-hims+0.5;
%%
for S=1:numW
   cor=W{S};
   corners=cor{S};
   numC=numel(cor);
   Ifilt=Ifilt1{S};
   for T=1:numC
     corners=cor{T};
     rect=[corners(:,1) corners(:,2); corners(1,1) corners(1,2)];
     W1=roipoly(size(Ifilt(:,1)), size(Ifilt(:,2)),Ifilt,rect(:,1),rect(:,2));
     A=find(W1==1);
     [y,x]=ind2sub(size(Ifilt),A);
     U=[y,x];
     LP=Ifilt(W1);
     numU=numel(U(:,1));
     M=zeros(size(W1));
     for P=1:numU
       x=U(P,1);
      y=U(P,2);
       klm=LP(P);
       M(x,y)=klm;
     end
     W2{T}=M;
   end
   M1{S}=W2;
   clear W2 M
  clear P

end
fname='Regions';
save(fname,'M1');
```

G. ZNCCF

```
tic
M1=importdata('Regions.mat');
numM=numel(M1)

for S=1:numM-1
   x0=M1{S};
   y0=M1{S+1};
   numx=numel(x0);
   numy=numel(y0);
   for T=1:numx
      w1=x0{T};
      for H=1:numy
         w2=y0{H};
         u=znccf(w1,w2);
         u1{H}=u;
      end
      U2{T}=u1;
      clear u1 u
   end

 U{S}=U2;
 clear U2

end

numv=numel(U)
 for K=1:numv
    c=U{K};
    numc=numel(c)
    for S=1:numc
    c1=c{S}
    c2=cell2mat(c1);
    [maxCorrValue, maxIndex] = max(abs(c2(:)));
     [Indy, Indx] = ind2sub(size(c2),maxIndex(1))
    Z{S}=[maxCorrValue, maxIndex];
    z1{S}=[Indy, Indx];
    end
    numz=numel(Z)
    V=zeros(numz,2)
    for H=1:numz-1
       U1=Z{H};
       u1=U1(1,1);
       u2=U1(1,2);
       V(H,1)=u1;
       V(H,2)=u2;
    end
    [max, maxI] = max(abs(V(:,1)));
    CorrV{K}=[max, maxI];
    final{K}=V;
  clear u1 u2
 end
```

```
time=toc
```

H. ZNCC function

```
function m = znccf(w1, w2)

   w1 = w1 - mean(w1(:));
   w2 = w2 - mean(w2(:));

   denom = sqrt( sum(sum(w1.^2))*sum(sum(w2.^2)) );

   if denom < 1e-10,
      m = 0;
   else
      m = sum(sum((w1.*w2))) / denom;
```