Interference Analyzer

Technical documentation

Software requirements

Experimental physics: “Analysis of interference images”

by “only-three-left” team of developers

Faculty of mathematics, physics and informatics,

Comenius University of Bratislava,

13.10.2021

Table of contents

Introduction

1.1 Purpose of requirements document

This is a formal document created with the standard IEEE/ANSI 830-1998 (Recommended Practice for Software Requirements Specifications). The main purpose of this document is to specify all of the requirements for the system, which was created as the compulsory project to the subject “Development of the information systems” at the Faculty of Mathematics Physics and Informatics (UK BA). The requirements document is dedicated to all of the stakeholders. Moreover it can also work as an agreement between the developers and the client.

1.2 Scope of the product

The main goal is to create software for the Department of Experimental Physics at the Faculty of Mathematics Physics and Informatics (UK BA). Software will determine a bending/curving of samples from the images of the interference pictures, so we can reconstruct the shape of the sample.

1.3 Definitions, acronyms and abbreviations

I***nterference pattern*** - consists of light reflected from surface A and light reflected from the bent sample B, which can result in a wave of greater, lower, or the same amplitude

Stakeholder - is anyone who influences or interferes with product development

GUI (Graphical User Interface) - it is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

1.4 References

- Github project repository
- Subject webpage
- Documentation archive
- Photo collection (including photo of samples and GUI mock-up)

1.5 Overview of the remainder of the document

In the remaining chapters of this document the reader will be informed about the system. Second chapter describes the perspective of the application and all functionalities.

General description

2.1 Product perspective

Product will be a graphical desktop application. After uploading or acquiring an input image, the application will be able to detect the radius of curvature of the sample once the user specifies the pattern center and line direction of measurement. A corresponding graph of intensity along the selected line will be shown as well.

2.2 Product functions

There are several product functions defined:

- Load images from filesystem or capture image from USB camera
- Allow user to upload a camera calibration config file from the filesystem
- Enable or disable the camera calibration
- Show the image
- Zoom the image
- Allow the user to define a line on the image (interference center and a point) and specify how many maximums user wants to see
- Generate clear interference graph
- Allow user to extend the width of the line, making it a circle sector, so that the generated graph gets cleaner by integrating the values acquired from this extension
- Calculate the radius of curvature of the sample
- Determine the maximum and minimum radius of curvature for all possible lines starting from the interference pattern center
- Allow user to generate point cloud document representing image, create .XYZ file

2.3 User characteristics

User of this software is a student or a university professor, who is doing a physical experiment in the laboratory, studying the interference. It is assumed that the software user has some basic physics knowledge in the field of interference. Nevertheless, the app interface and functions can be understood without it.

2.4 General constraints

It is assumed that the image uploaded to the program has a clear interference picture. This requires prior camera calibration and additional tools for camera fixing. Ideally the camera should be placed so that the plane of the camera

sensor and the plane of the screen with the pattern are parallel, which leads to the more accurate calculations.

2.5 Assumptions and dependencies

This software requires a PC running it to have a JRE of version 8 installed and to have a standard I/O system. It is assumed that the input contains the interference image produced in the result of the physical experiment. Otherwise it is assumed that the user has a camera connected via USB port and that the captured image can be decoded by OpenCV library.

Specific requirements

3.1 Functional requirements

This section gives an overview of the specific functional requirements for the software. For better understanding of importance of any single requirement, we define three priority levels:

[P1] - High priority
[P2] - Medium priority
[P3] - Low priority

3.1.1 *[P1]* Option to read image from filesystem

User can select the option to load the image from the filesystem via the common FileChooser Dialogue window. This image is then to be displayed to the user, allowing him to do measurements on it.

3.1.2 *[P1]* Option to read image from camera

A user can select the option to capture the image via the connected camera using a camera preview on/off switch. When the switch is off, the preview is stopped and the last image retrieved from the camera can be analyzed. This image remains displayed to the user, allowing him to do measurements on it. After switching the preview switch on, the image is overdrawn by new frames from the camera.

3.1.3 *[P1]* Specify the center of an interference pattern

A user can select the option to select a pixel in the interference pattern shown in the window using a mouse pointer. This pixel will be considered to be the center of the interference pattern in the forthcoming calculations.

3.1.4 *[P1]* Specify the target point of the line segment for calculations

A user can select the option to choose the second point of the line segment where measurement is to be done. The first one [3.1.3] must be located in the center of the pattern, the second one is an arbitrary pixel.

3.1.5 *[P1]* Specify the reference line segment in the interference pattern

A user can select the option to use the mouse pointer to select the starting and final points of a reference line segment in the interference pattern image. The measured length in pixels is shown in an edit box and can be entered by the user also manually. The reference line length in pixels is then used to recalculate the pixel/mm ratio together with the input in 3.1.6.

3.1.6 *[P1]* Assign the length to the reference line segment

A user can specify the length (in the input box) of the reference line segment in millimeters.
These two values (3.1.5 and 3.1.6) are then used for radius calculations.

3.1.7 *[P1]* Limit the number of maximus to find

The user can specify an upper limit on the number of maximums to detect on the selected line segment. After the specified limit is reached, further maximums will not be searched for.

3.1.8 *[P1]* Produce the graph

After the user selects the line segment, the graph showing pixel intensity along the selected line segment is automatically shown in the graph area in the main user interface window. User has an option to smoothen the graph by adjusting the width of the line if the image is not clear enough. The graph visualizes the detected maximum extreme points.

3.1.9 *[P1]* Analyze maximums and minimums of the graph to get radius

With the use of the information received from the image and programmed formulas the app is able to calculate the radius of the interference for any two consecutive maxima detected on the selected. It produces a list of such radiuses up to the limit specified (3.1.7) or the number of detected maxima on the line segment. All of these values are then presented to the user in the text area (3.1.16).

3.1.10 *[P3]* Calculate maximum and minimum radius

After the interference image has been received by any input source the user can select the option to calculate the radius of curvature of the

sample for every possible line defined from the center of the interference, iterating by angular step specified by the user in a separate edit box. The user is then presented with the result consisting of the following values: maximum interference radius and minimum interference radius and the angles under which they were found. The lines representing the two directions of measurement are then drawn on the image and labeled with these values.

3.1.11 *[P3]* Export PNG image

The user can at any time select an option to save the currently shown image of the interference pattern including the marked line segment, or minimal and maximal sample radius directions in PNG format.

3.1.12 *[P3]* Simple user manual

Either in-app manual or external documentation describing the app's workflow will be accessible from the software.

3.1.13 *[P1]* Camera calibration configuration file

Software provides an option to load the camera calibration file from the filesystem. User then can enable or disable it.

3.1.14 *[P2]* Perform sweep measurement to obtain a point cloud

The user has an option to start an automatic sweep of the whole interference pattern along lines going from the specified pattern center in all possible directions (using an angular step specified in an edit box). For each direction, all the consecutive pairs of maxima up to the specified limit will produce one point [X,Y,Z], which is determined from the radius of the curvature of the sample for that particular pair of maxima. The resulting point cloud - the x, y and z coordinates of such 3D points are saved to a ".xyz" data specified by the user.

3.1.15 *[P1]* Additional experimental parameters

Let user insert parameters based on which we will calculate: user might need to input parameters of laser setup (distances between laser, sample, camera etc). In particular these are:
D [mm] - the distance between the sample and the screen with the pattern
lambda [nm] - the wavelength of the laser

3.1.16 *[P1]* Text report from the experiment

The main user interface window will contain a text area where the program will output and append all important values during the experiment, i.e. the coordinates of the center of the interference pattern, the selected target point of the analyzed line segment,

reference line length in pixels and millimeters, the coordinates of the detected maxima, the calculated radiuses for each pair of the maxima, angles for the minimum and maximum radius, and all other important values. This text output is automatically appended to a text file including timestamps.

3.2 Non-functional requirements

In this chapter we are talking about specifications that describe the system's operation capabilities and constraints that enhance its functionality. Simply said, the requirements that describe operational qualities rather than the behavior of the product.

3.2.1 Performance

The software does not include any unnecessary artificial delays. Performance delays are minimised as much as possible.

3.2.2 Portability and compatibility

The software is compatible with most popular operating systems. The software does not have any specific hardware requirements, so it can be deployed on every modern PC. Portability is available through Docker image system.

3.2.3 Reliability and availability

Software processes any possible exception occurred, making its run flawless for the user. No app crashes are expected and tolerated, which means the app is available to the user for its full operational time.

3.2.4 Maintainability

Modularity and clean code structure of the software makes it easily maintainable. The app supports small and effective patch flow.

3.2.4 Security

There is no additional security layer defined as the operating data is not personal, so the app doesn't have to be protected against malware attacks or unauthorized access.

3.2.5 Localization

Software does not support any localization features. English was chosen as a primary language for any outputs provided by the software as it is the common international tongue.

3.2.6 Usability

All app features are easily accessed through interface elements. Overall GUI solution is user-friendly and requires minimum additional knowledge. Interface is rather intuitive and uses a calm and pleasant color scheme.

3.3 Interface requirements

The placing and the purpose of every GUI element is described in detail in GUI design documentation in close collaboration with the client.

Design of the information system

Experimental physics: “Analysis of interference images”

by “only-three-left” team of developers

Faculty of mathematics, physics and informatics,

Comenius University of Bratislava,

30.10.2021

Table of contents

Introduction

1.1 Purpose of the document

The main purpose of this document is to have completed and detailed design of the information system. It contains all information needed to explain and understand the functionality of the system, as well as, how to implement the system. This document is primarily dedicated for developers. The content in this document covers all requirements from the software requirements document.

1.2 Scope of the document

This document is closely related to the software requirements catalog, so it is required to be familiar with it.
It contains completed and detailed design of implementation of all requirements from the software requirements document. It further specifies external interfaces, overall design of the user environment, including visualization. The implementation proposal is described by diagrams. In addition, there are described which technologies have been used and the final deployment into operation.

1.3 Overview of the following chapters

The following chapters deal with the external interface, data model, user interface and its visualizations. The last chapter describes detailed system implementation design.

Specification of external interfaces

The application runs locally and communicates with the user using a graphical user interface. The core of the system communicates with one external device, which is a camera (connected to the device via a USB port). The system detects this port, and communicates for further needs such as taking the picture.

Data model

The software has three file outputs: two camera calibration files, the image with marked directions of minimum and maximum curvature of the sample and the “.csv” containing the characterization of the interference image - the 3d vector value for each pixel.

3.1 The camera calibration files

Camera calibration process results in two binary files, which can be then read and decoded by the software to correct the image. First one contains an intrinsic matrix. The second one contains a matrix of distortion coefficients. They will be called “intrinsic” and “distortion” respectively. There will be no extension specified as opening them with standard applications gives no sense.

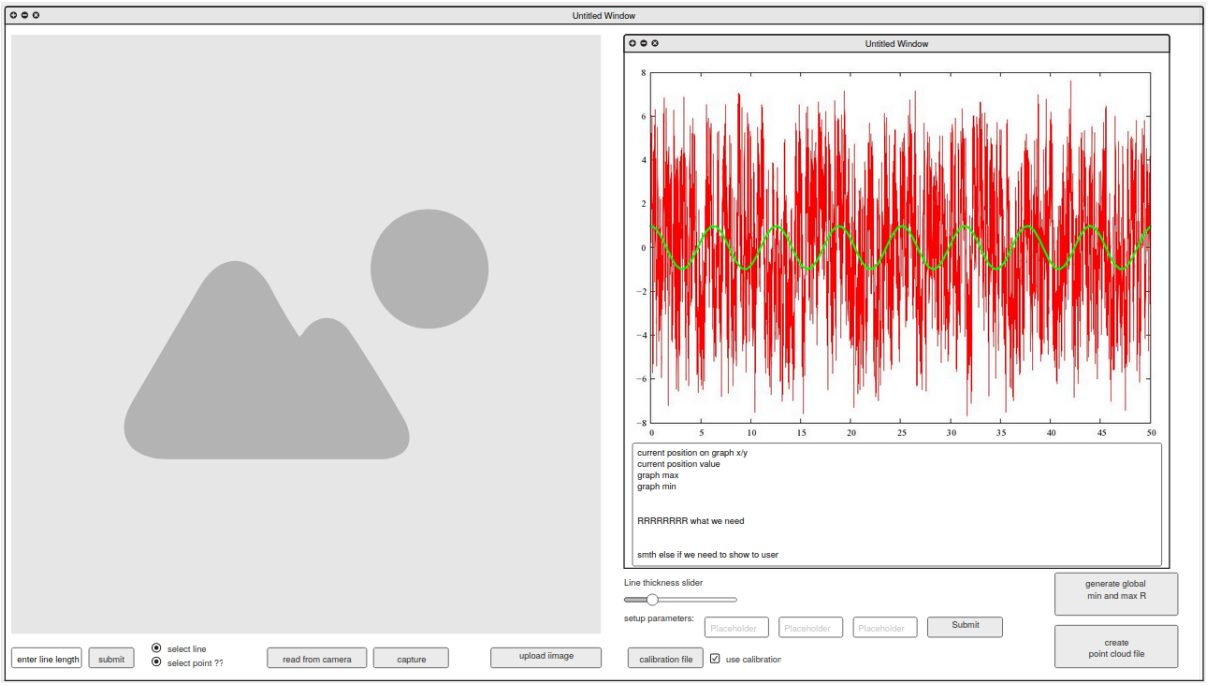
3.2 The output image

The software generates an image with minimal and maximum curvature directions marked on it as two lines. This file can then be saved in selected by the user folder with the specified by user name. The extension of the file is “PNG”.

3.3 The output .xyz file

The user can select an option to generate an output text file containing a point cloud representing the 3d points of the sample.
The file has four columns named alpha, x, y, and z respectively and as many rows as the number of points obtained. Alpha represents the angle of the vertical plane in which the point x, y, z was found.

User interface design



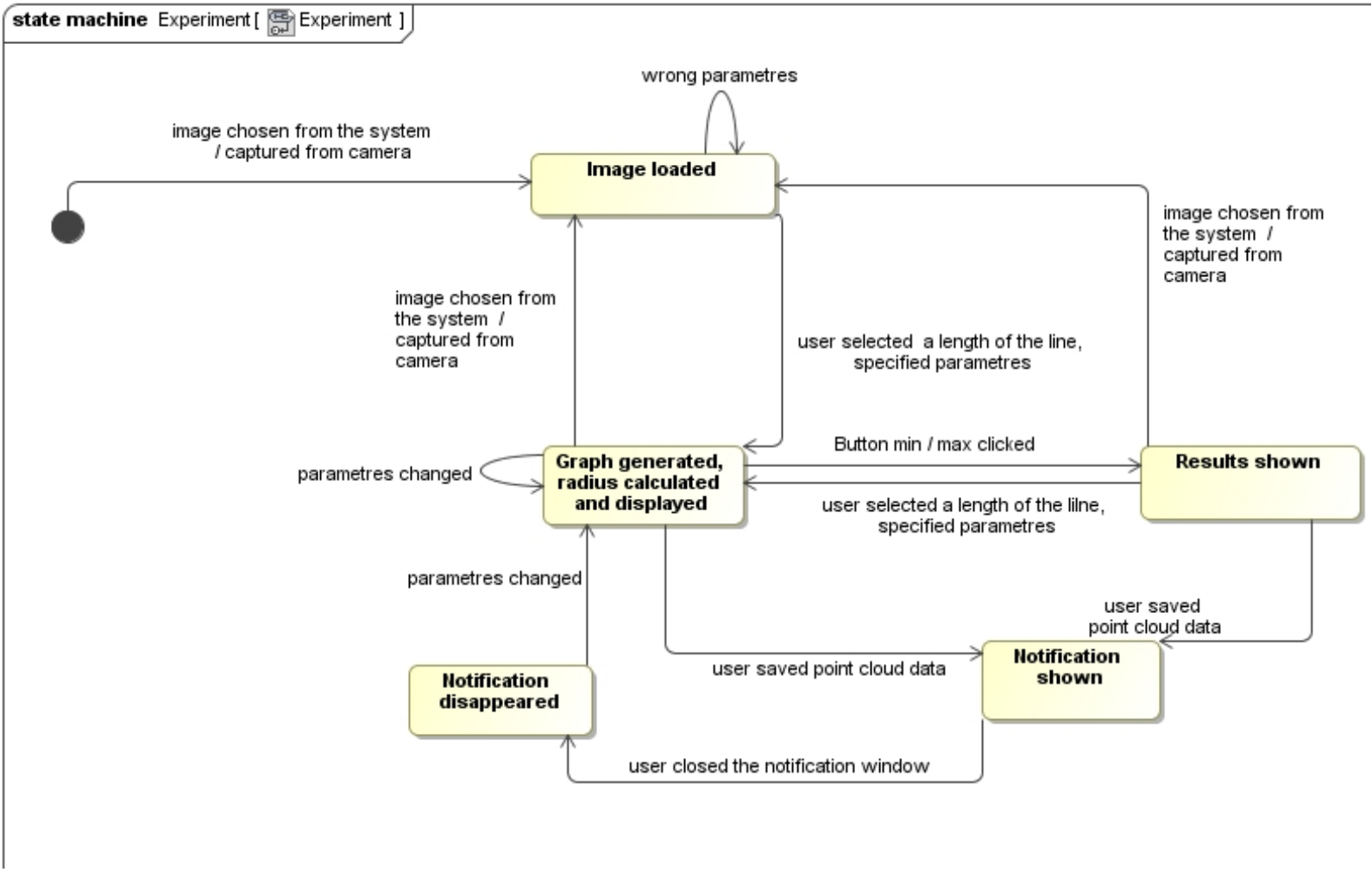
4.1 Graphical user interface

4.1.1 LINE LENGTH INPUT BOX - write length of line
4.1.2 SUBMIT BUTTON - push button to specify line length
4.1.3 SELECT LINE/SELECT POINT RADIO BUTTON - option for user to select one point (and second one in the middle) or two different points
4.1.4 READ FROM CAMERA BUTTON & CAPTURE BUTTON - opens new window with real-time data from camera with option to capture image and use it
4.1.5 UPLOAD IMAGE BUTTON - upload image from local file tree
4.1.6 CALIBRATION FILE BUTTON - select calibration file from local tree
4.1.7 CALIBRATION CHECKBOX - use calibration file to calibrate camera
4.1.8 SETUP PARAMETER FIELDS - enter setup distances (important for formulas)
4.1.9 LINE THICKNESS SLIDER - move left and right to make line of measurement thinner or wider respectively
4.1.10 MIN MAX BUTTON - generate min and max R on the image
4.1.11 SAVE BUTTON - generate point of cloud and save as csv file
4.1.12 IMAGE PLACEHOLDER - shows captured/uploaded image
- selected place will be highlighted as one/two points

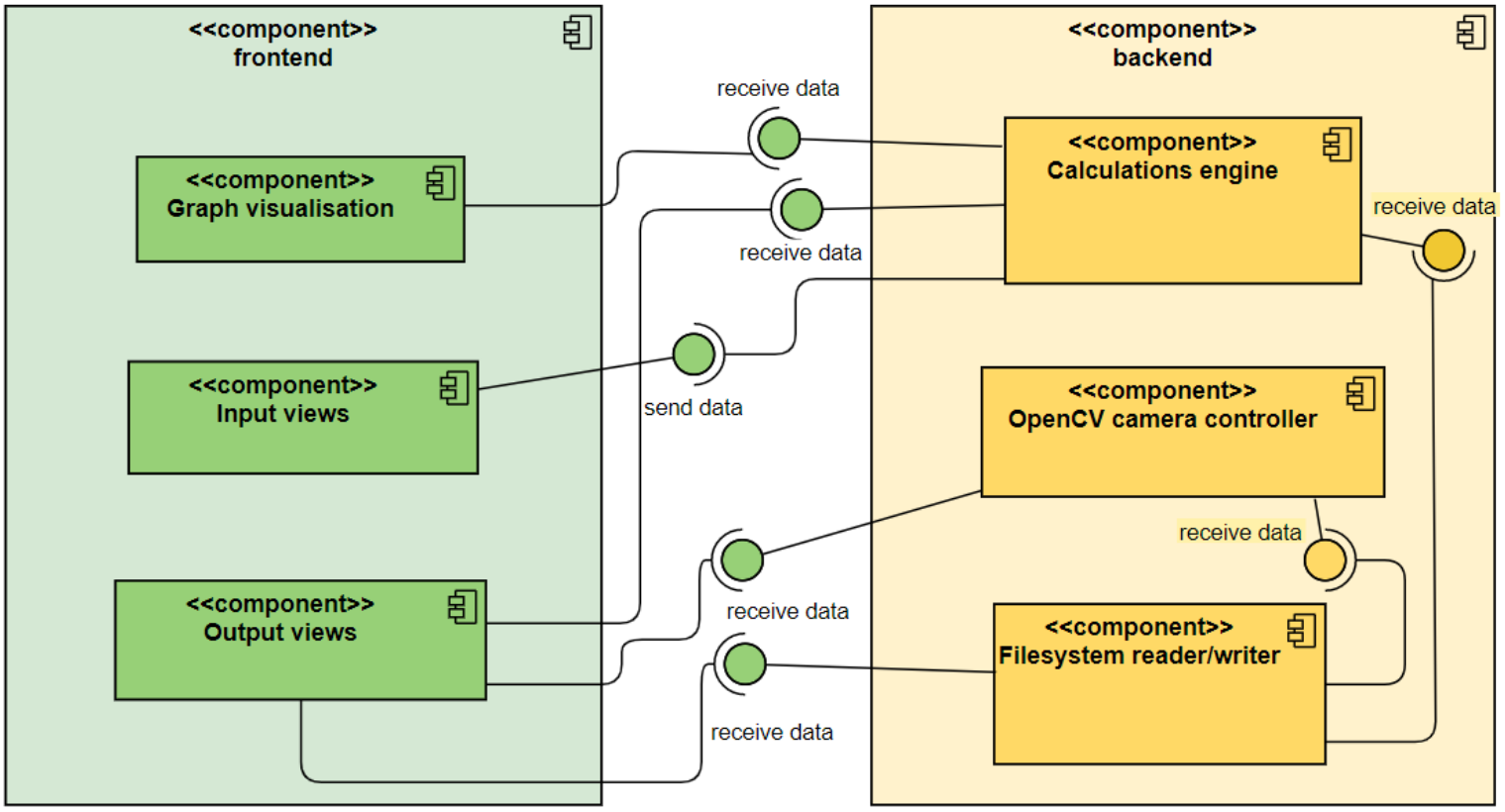
4.1.13 GRAPH WINDOW - only shows generated graph
4.1.14 INFORMATION BOX - user can see all the information here
- current x/y position on graph
- current position value
- graph total max
- graph total min
- R
- whatever else we need to show to us

Implementation design

5.1 State Diagram of user interface

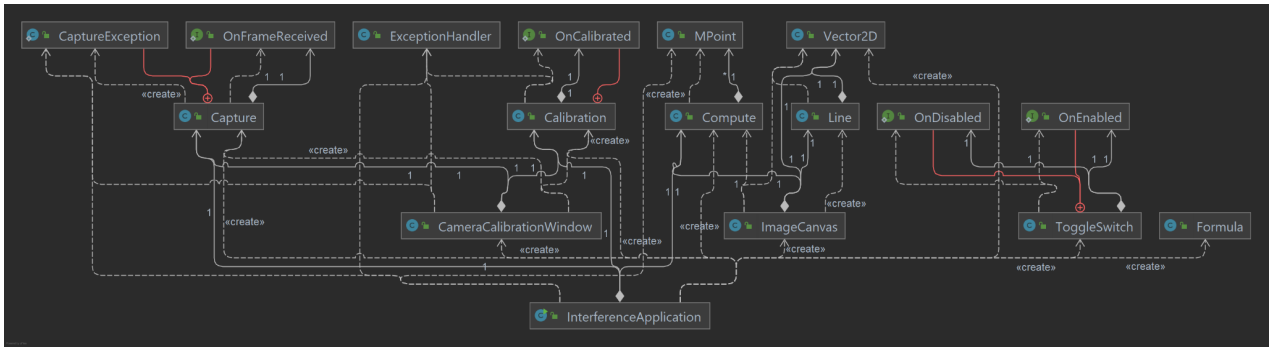


5.2 Component Diagram

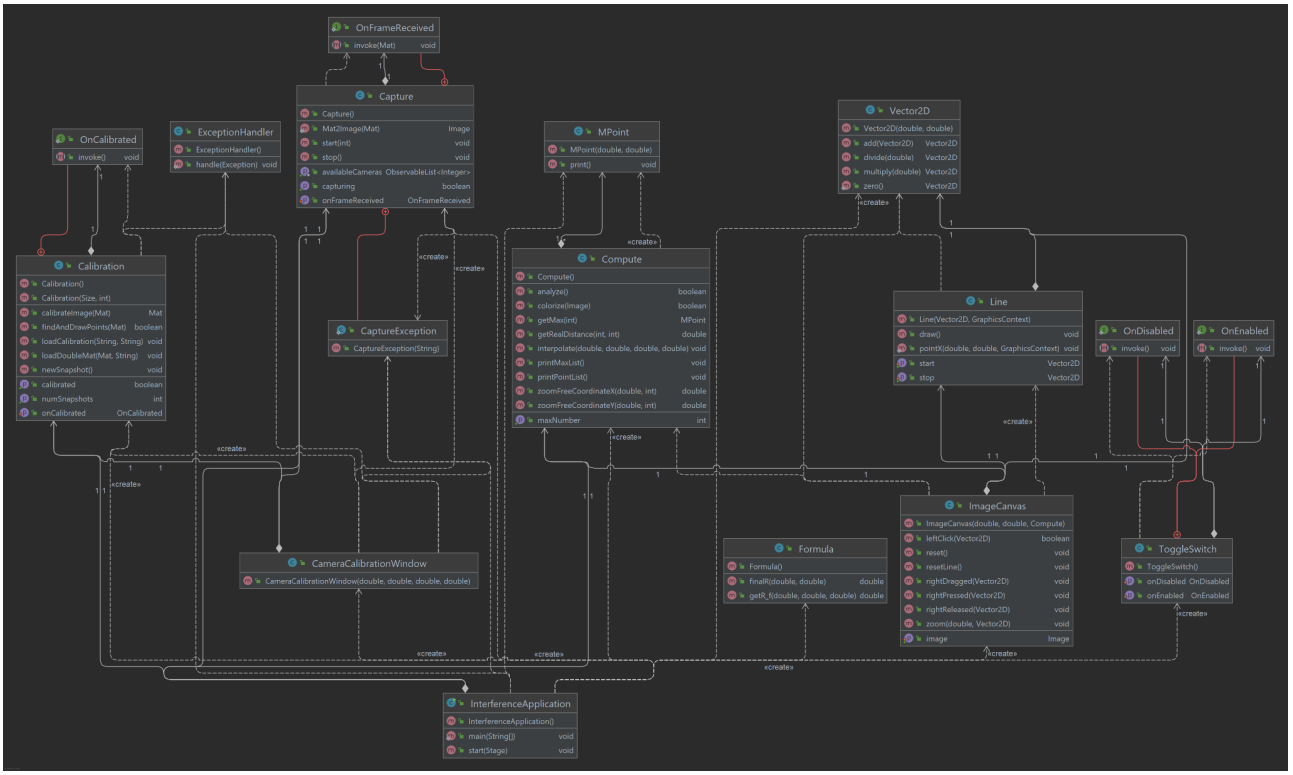


5.3 Class Diagram

5.3.1 Simplified version



5.3.2 Full version



5.4 Applied technologies

5.4.1 Java

Java is the main language of development used in this software's development process.

5.4.2 JavaFX

JavaFX is the main GUI tool used in the development of this software. Besides using common application, scene and stage modules, a LineChart class is also used for chart visualisation.

5.4.3 OpenCV

Core OpenCV libraries are one of the most important parts of the app. VideoCapture, Calib3d, Imgproc and Imgcodecs classes are used for the camera calibration process.

5.4.3.1 org.opencv.calib3d.Calib3d

- *findChessboardCorners*
  This method is used to write the positions of the chessboard corners into the Matrix for further calibration purposes.
- *drawChessboardCorners*
  This method is used to draw the chessboard corners on the bitmap, which is then presented to the user.
- *calibrateCamera*
  This method produces intrinsic and distortion coefficient matrices from the points found.

5.4.3.2 org.opencv.videoio.VideoCapture

- *open*
  This method is used to start the camera capture.
- *release*
  This method is used to release the capture.
- *read*
  This method is used to read the next frame from the camera.

5.4.3.3 org.opencv.imgcodecs.Imgcodecs

- *imencode*
  This method is used to encode the matrix into the PNG image.

5.4.3.4 import org.opencv.imgproc.Imgproc

- *cvtColor*
  This method is used to convert the image into the grayscale.

5.5 Plan of the implementation

- User interface
    - Main app window + graph views
    - Calibration window
    - User Manual
- Camera calibration
    - Save to binary file
    - Read from binary file
    - Apply calibration data to the input image
- Capturing/uploading image
    - FileChooserDialogue for uploading
    - Stream camera image to the ImageView of the main window, capture the frame with dedicated button
- Generate graph based on user-specified line
    - Specify line on the image
    - Slider for extending the line into the circle segment
    - Generate graph with some algorithm
- Find total min/max R through all directions
    - find the correct solution
- Save image as file (point clouds)
    - find the correct solution

Testing scenarios

1. scenario: User clicks on the „upload image from system“ button
Desired output: the file explorer window will appear

2. scenario: User clicks on the „read from camera“ switch
Desired output: after he prompted for the camera selection the image from camera appears on the canvas

3. scenario: User chooses the image from the system
Desired output: Image is shown on the left side of the application window

4. scenario: When is the image uploaded, the user marks two spots,
Desired output: There will be shown two green points (as a beginning and end of the line) and the „Input the length of the line“ window will appear

5. scenario: After inputting the length of the line, the user marks another two spots
Desired output: there will be shown two blue points (as a beginning and end of the line) and the „Input the number of maximums“ window will appear

6. scenario: User wants to change the thickness of the line
Desired output: user drags the slider to one or the other side

7. scenario: User will fill out the size of D and push Submit button
Desired output: D will be taken into account in future calculations

8. scenario: Whenever the user wants to change the line
Desired output: he clicks on the other spot inside the image, and the window with the „Input the number of maximums“ window will appear (scenario 5)

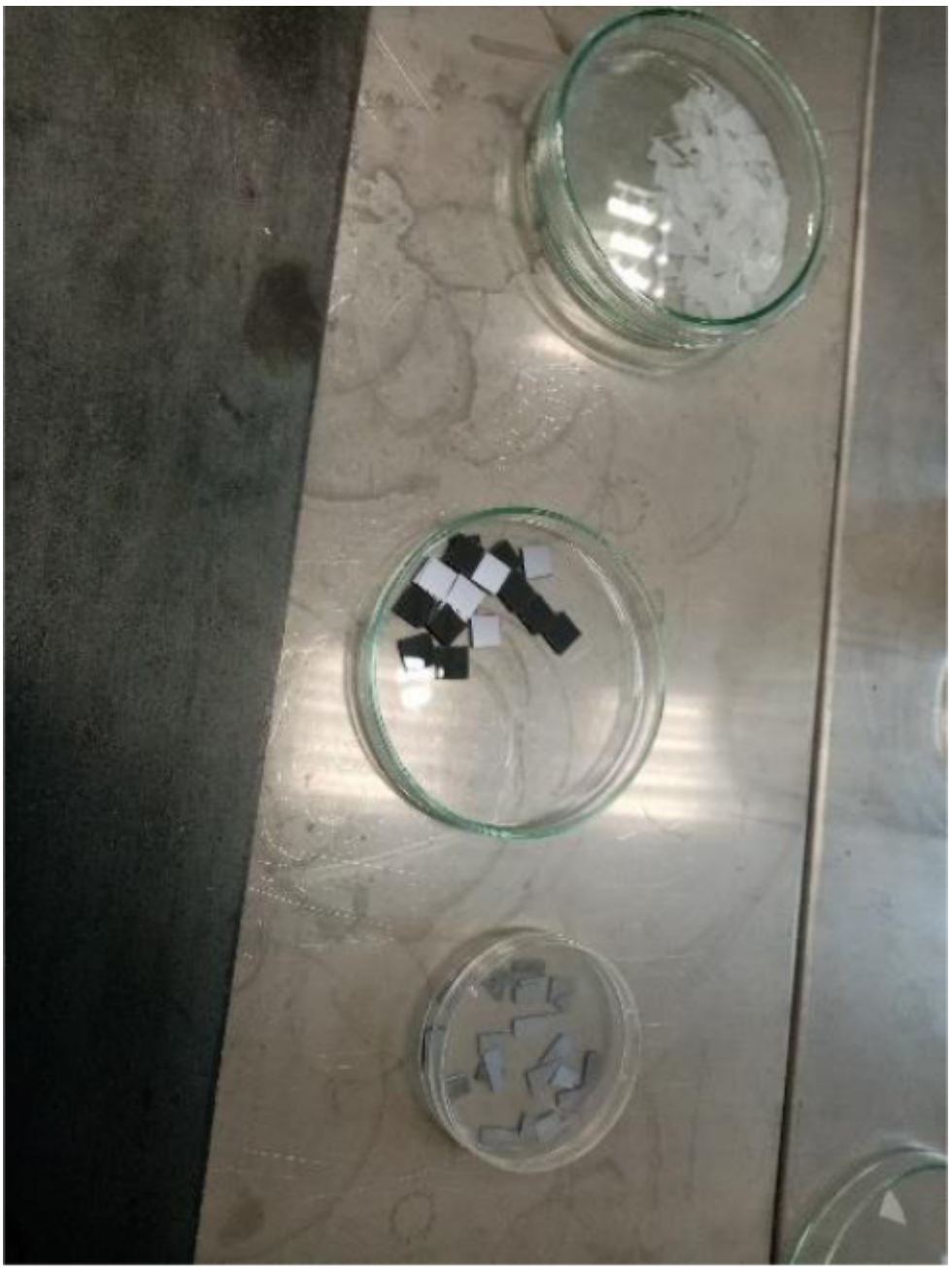
9. scenario: User defines the line
Desired output: graph will be shown and the result of desired maximums – 1 will be shown in the output box under the graph visualization

10. scenario: User wants to reset line
Desired output: user clicks on the label “Edit” in the upper left corner, and then clicks on the label “Reset line”, then he can mark 2 spots in the image (scenario 5)

11. scenario: User will click outside the image
Desired output: nothing happens

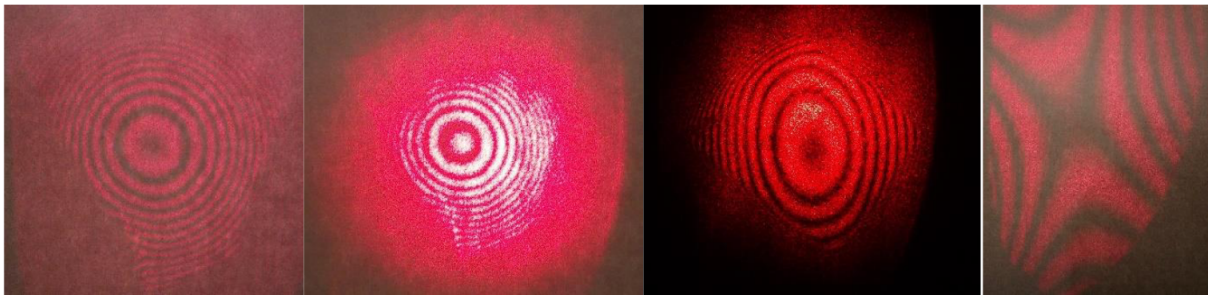
Appendices

4.1 Actual samples, subjects of the experiments

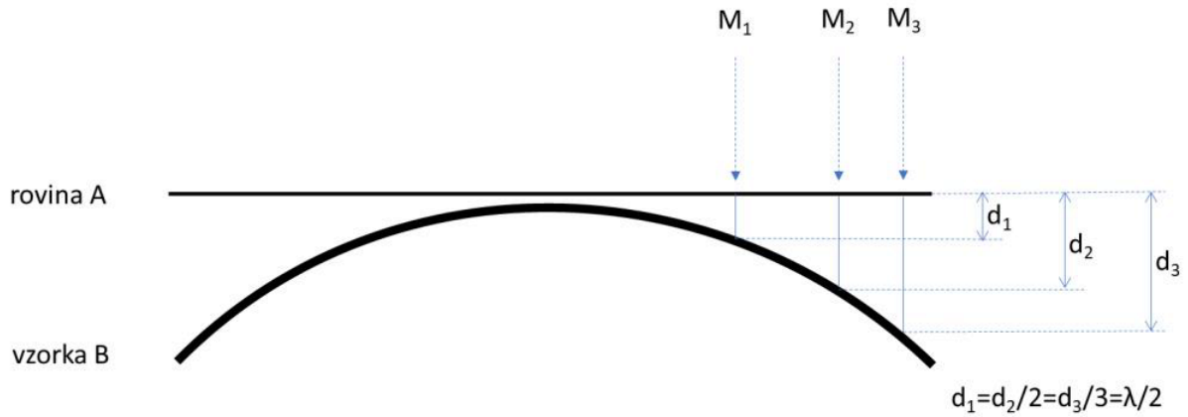


4.2 Laboratory, where the experiments are being performed

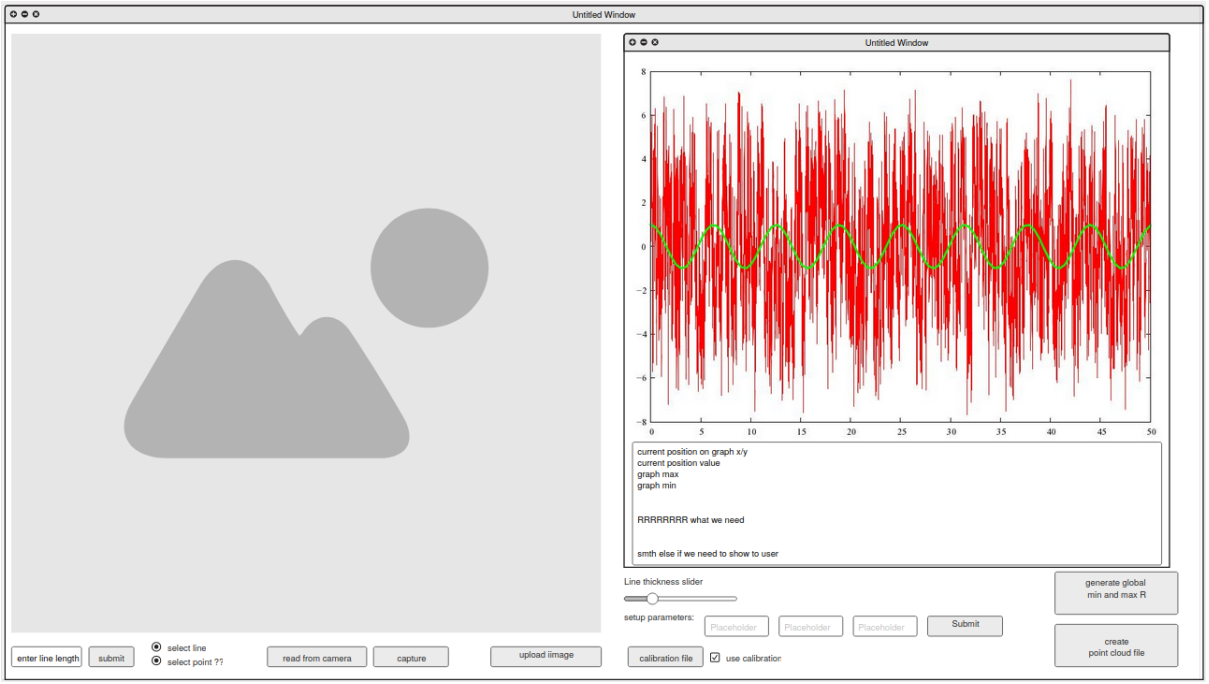
4.3 Examples of interference images



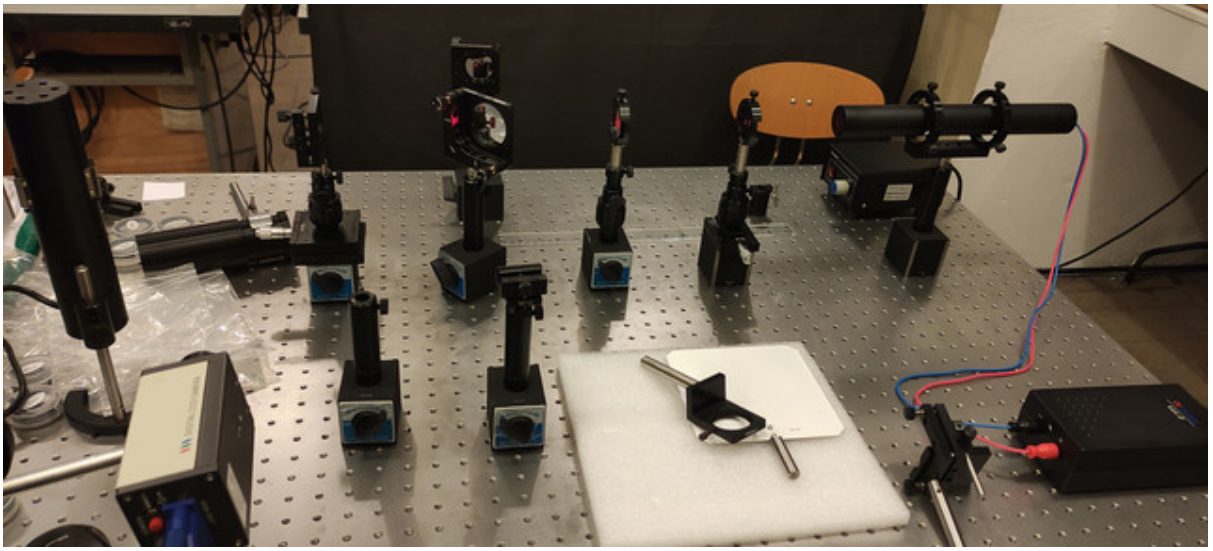
4.4 Geometry of the interference image experiment



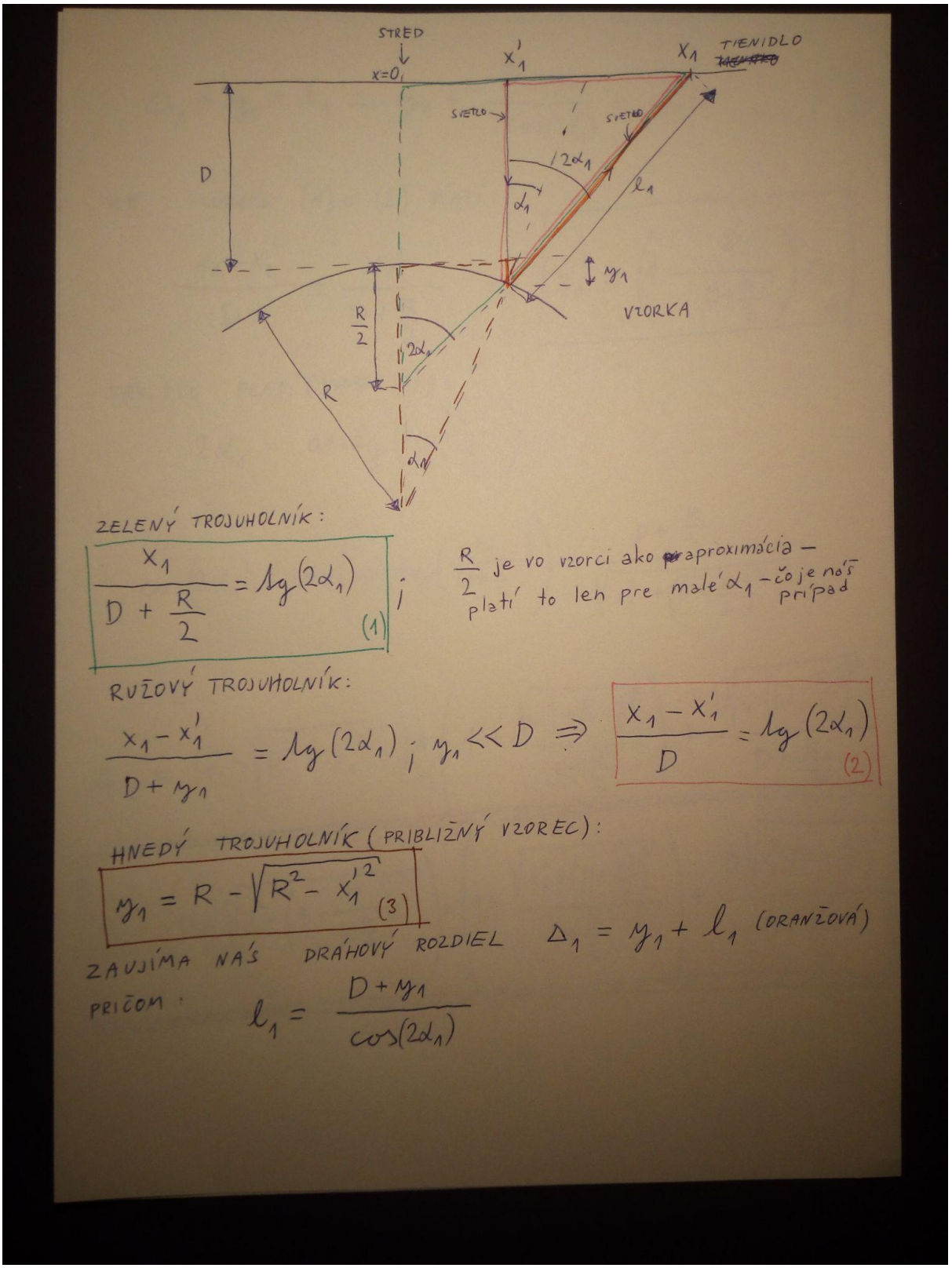
4.5 GUI mock-up



4.6 Actual measuring set-up



4.7 Formulas



ZELENÝ TROJUHOLNÍK:

$$\frac{x_1}{D + \frac{R}{2}} = tg(2\alpha_1) \quad (1)$$

; $\frac{R}{2}$ je vo vzorci ako aproximácia – platí to len pre malé α_1 – čo je náš prípad

RUŽOVÝ TROJUHOLNÍK:

$$\frac{x_1 - x_1'}{D + y_1} = tg(2\alpha_1) \; ; \; y_1 << D \Rightarrow \frac{x_1 - x_1'}{D} = tg(2\alpha_1) \quad (2)$$

HNEDÝ TROJUHOLNÍK (PRIBLIŽNÝ VZOREC):

$$y_1 = R - \sqrt{R^2 - x_1'^2} \quad (3)$$

ZAUJÍMA NÁS DRÁHOVÝ ROZDIEL $\Delta_1 = y_1 + l_1$ (ORANŽOVÁ)

PRIČOM: $l_1 = \frac{D + y_1}{\cos(2\alpha_1)}$

$$\Delta_1 = y_1 + l_1 = y_1 + \frac{D + y_1}{\cos(2\alpha_1)} = y_1\left(1 + \frac{1}{\cos(2\alpha_1)}\right) + \frac{D}{\cos(2\alpha_1)}$$

Z ● VZORCOV (1) a (2) PLATÍ:

$$\frac{x_1 - x_1'}{D} = \frac{x_1}{D + \frac{R}{2}} \Rightarrow \boxed{x_1' = \left(1 - \frac{D}{D + \frac{R}{2}}\right) x_1} \quad (4)$$

TAKISTO PLATÍ (VZOREC (1)):

$$2\alpha_1 = \operatorname{arctg}\left(\frac{x_1}{D + \frac{R}{2}}\right)$$

$$\cos(2\alpha_1) = \cos\left(\operatorname{arctg}\left(\frac{x_1}{D + \frac{R}{2}}\right)\right) = \frac{D + \frac{R}{2}}{\sqrt{x_1^2 + \left(D + \frac{R}{2}\right)^2}}$$

ČIŽE

$$\Delta_1 = \left(R - \sqrt{R^2 - x_1'^2}\right) * \left(1 + \frac{\sqrt{x_1^2 + \left(D + \frac{R}{2}\right)^2}}{D + \frac{R}{2}}\right) + \frac{D\sqrt{x_1^2 + \left(D + \frac{R}{2}\right)^2}}{D + \frac{R}{2}}$$

$$\boxed{\Delta_1 = \left(R - \sqrt{R^2 - \left(1 - \frac{D}{D + \frac{R}{2}}\right)^2 x_1^2}\right) * \left(1 + \frac{\sqrt{x_1^2 + \left(D + \frac{R}{2}\right)^2}}{D + \frac{R}{2}}\right) + \frac{D\sqrt{x_1^2 + \left(D + \frac{R}{2}\right)^2}}{D + \frac{R}{2}}}$$