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.

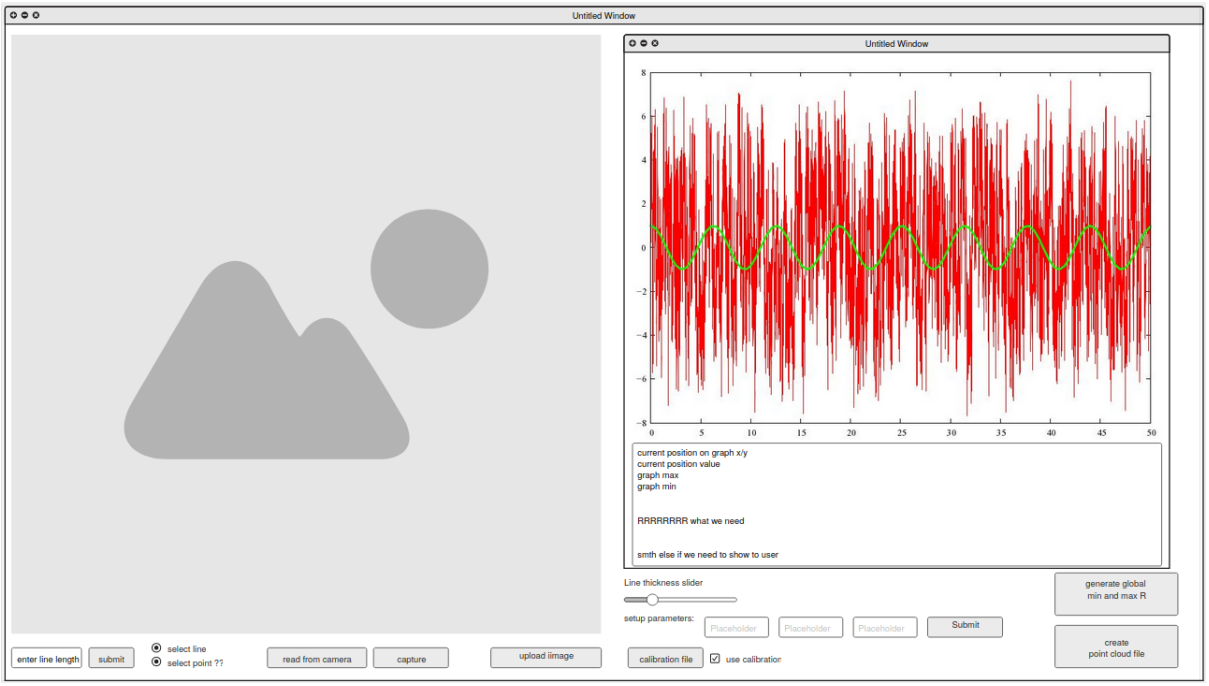
3.2 The output image

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



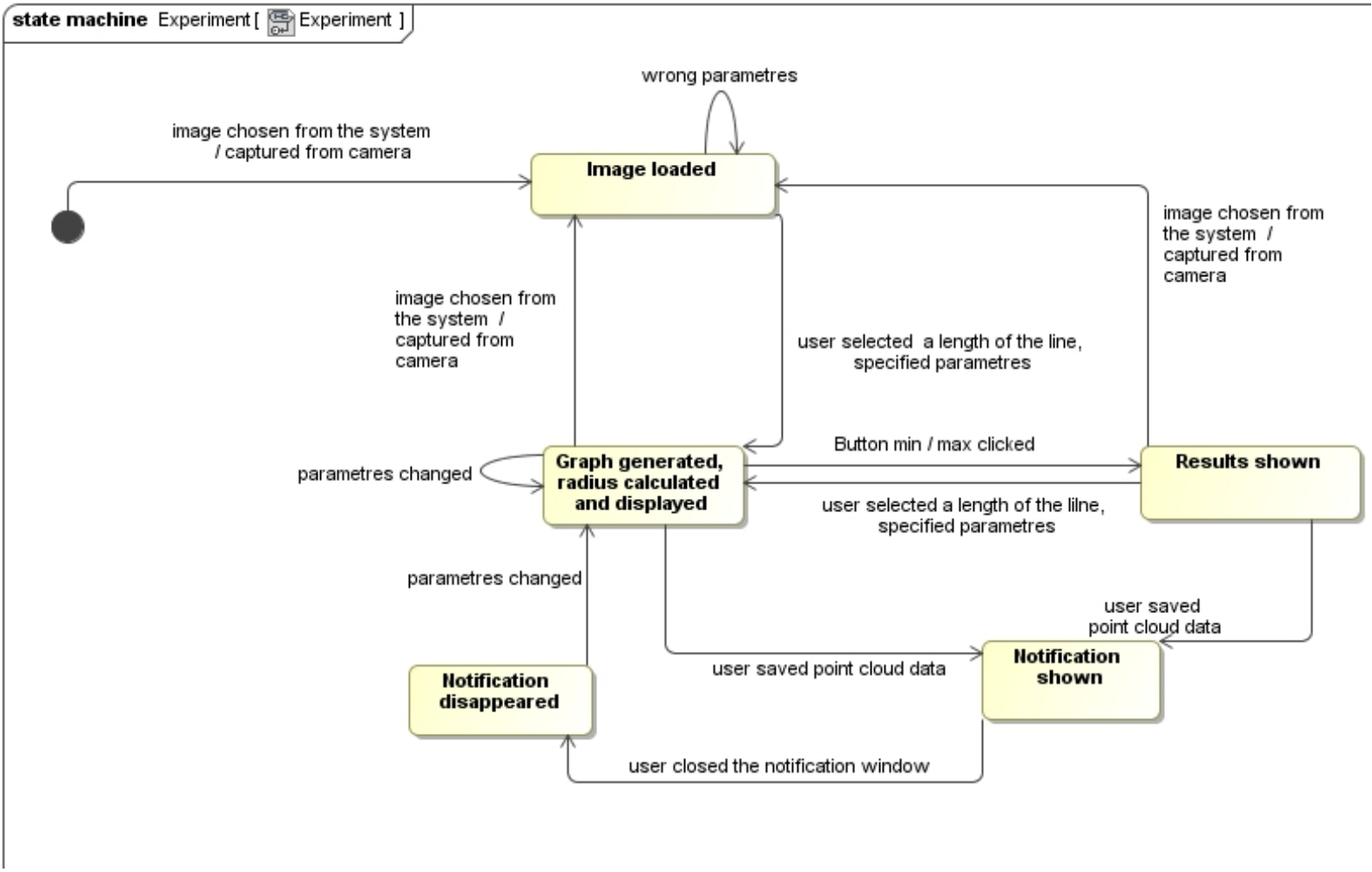
4.1 Graphical user interface

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

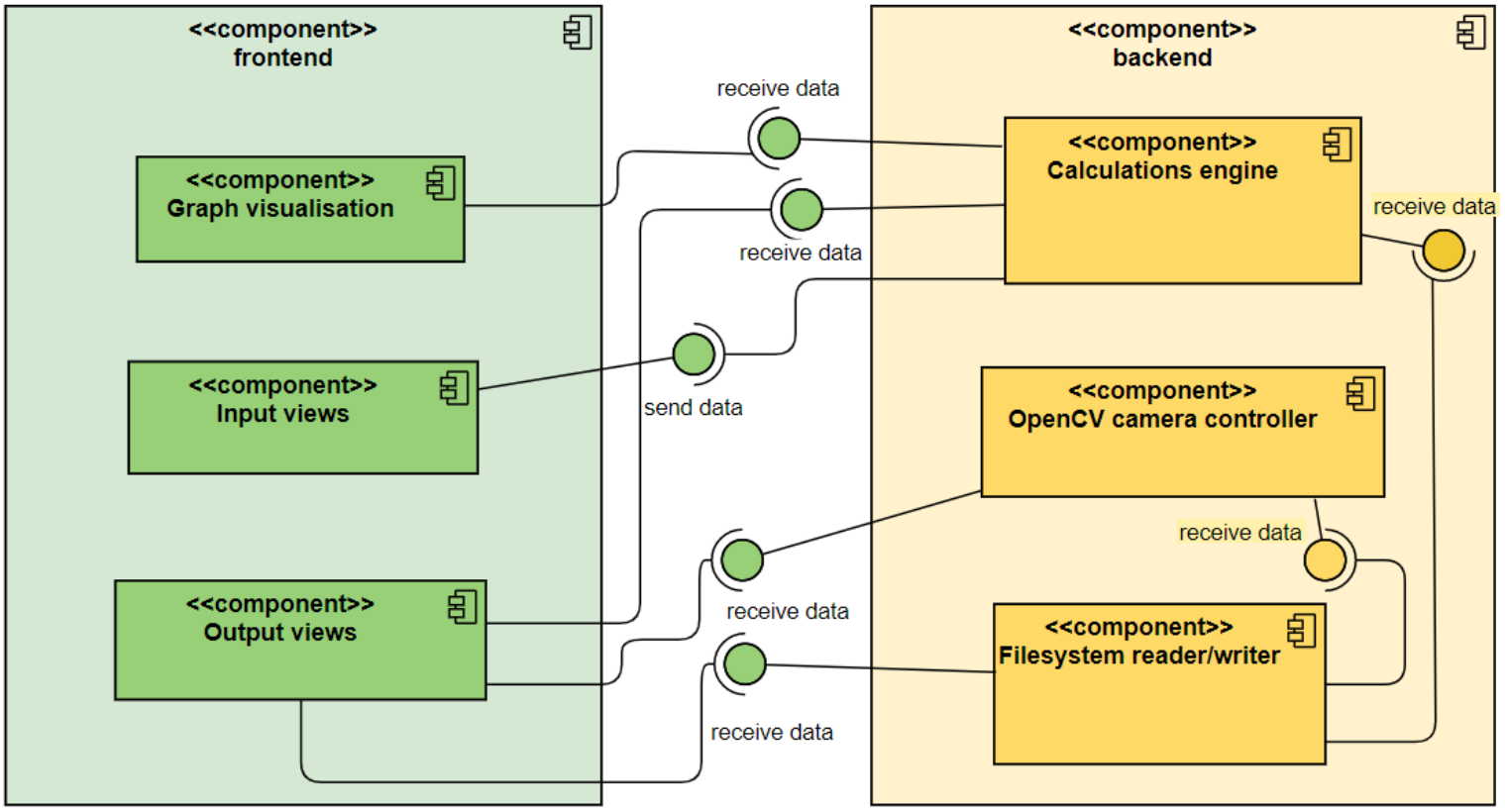
4.1.13 GRAPH WINDOW - only shows generated graph
4.1.14 INFORMATION BOX - user can see all the information here
- current x/y position on graph
- current position value
- graph total max
- graph total min
- R
- whatever else we need to show to us

Implementation design

5.1 State Diagram of user interface

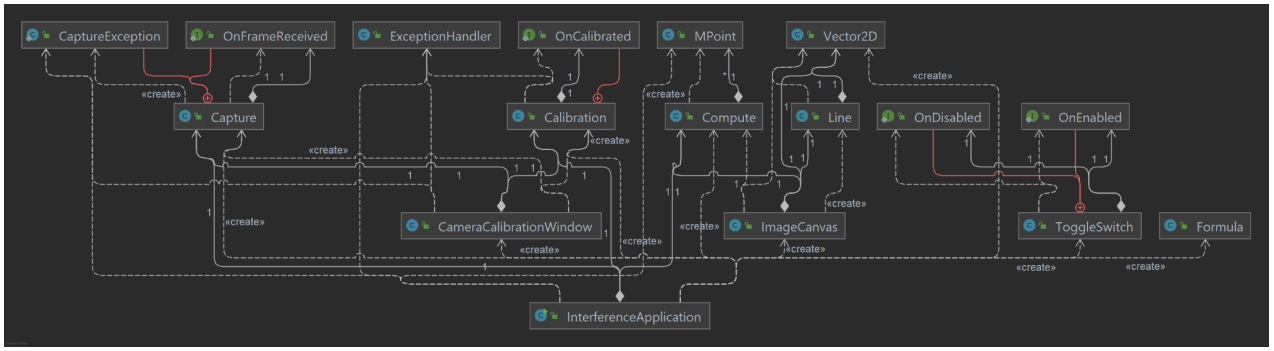


5.2 Component Diagram

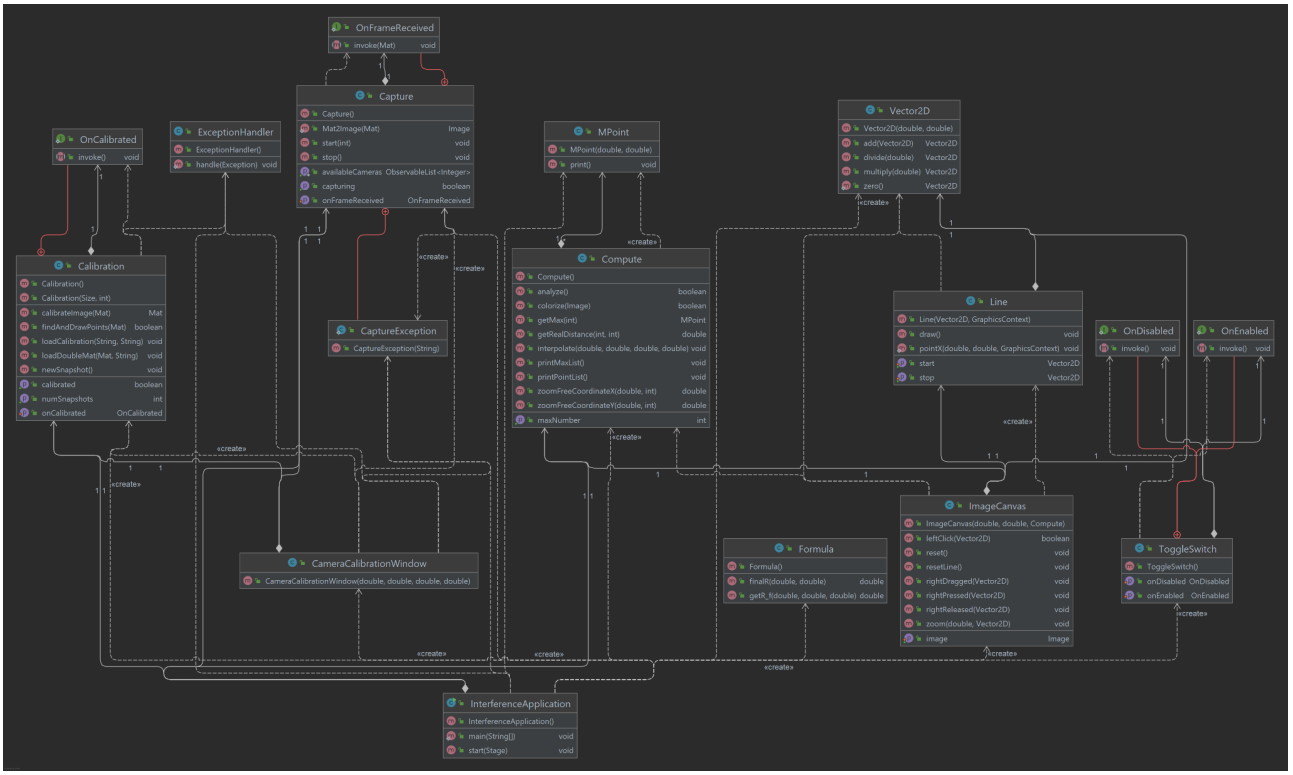


5.3 Class Diagram

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