SILESIANUNIVERSITY OF TECHNOLOGY
FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

Final Project

Algorithms of vision detection of manipulated object position

Author: Krzysztof Palka
Supervisor: Tadeusz Szkodny, PhD, DSc.
Consultant: Adrian Łęgowski, MSc, Eng.

Gliwice, February 2015

Table of Contents

1 Introduction

Nowadays, one of the most important parts of the robotized positions are vision systems which provides coordinates of position and orientation of manipulation. These parameters are calculated by appropriate software. In the process of designing such software, errors of coordinates read from camera matrix are taken into account. These errors are causing inaccuracies in calculation of coordinates of points in the reference system associated with technological site. In order to determine accuracy of coordinates of observed points, error analysis is necessary. Those errors are caused by: read errors of coordinates from camera matrix, optical distortions of camera, errors of parameters describing optical system of camera and calculation errors. In process of vision system design sought to minimize mentioned errors.

Calibration template presented in fig.1 was used during work. It consists of red markers in vertices of white and black squares. In the picture marked axes, where x,y means axes of reference system and x_c,y_c means axes of camera coordinate system. Also pointed the indicative position of 2 cameras.

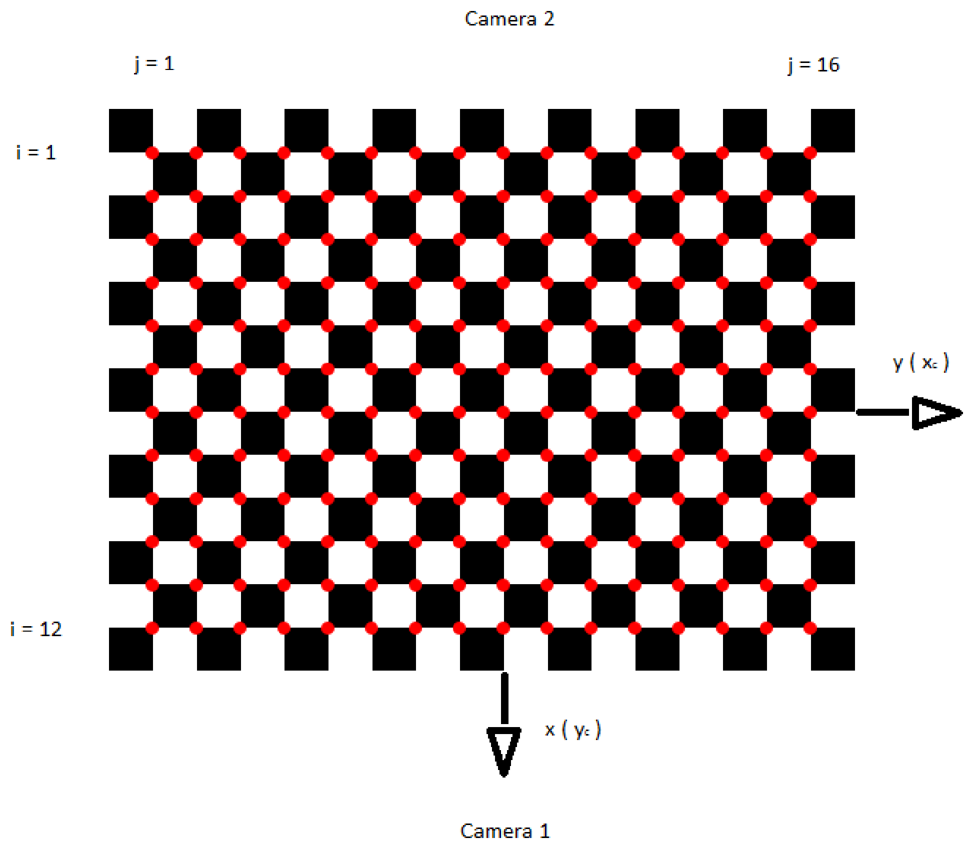


Fig. 1. Calibration template

To read points coordinates any graphical software can be used, but such a way poses a risk of making errors. In order to reduce chance of occurrence of such errors and to automatize gathering of these coordinate, program from engineer project of Artur Meller [1] was used. Afterwards these coordinates were corrected using mathematical model presented in works[2,3] in order to reduce error caused by distortions mentioned before

Camera [4] algorithm was used to calculate coordinates x_c y_c z_c of camera coordinate system and x y z of reference system. Focal length and pixel size were assumed from camera datasheet in order to minimize errors.

2 Aim and scope of work

In this chapter are presented basic assumptions which should be satisfied within this project.

2.1 Aim of the project

Aim of this project is to create software which calculates position of the object of manipulation, using coordinates of characteristic points gotten from external program. Algorithm was elaborated which calibrate the camera in order to (using calibrated data) obtain position of object in the form of matrix which can be easily transformed to the form specified by robot. During work following tasks were satisfied:

- Elaborating and making the object and calibration template,
- Elaborating algorithms of cameras geometrical calibration and their implementation in the external computer,
- Cameras geometrical calibration in the environment of external computer,
- Creating algorithms to obtain position of observed object using stereovision and their implementation in the external computer,
- Design of worksite and application tests.

2.2 Used elements

During work on project following elements were used

- Web camera EDIMAX IC-7100P (Fig.2),

Fig. 2. Camera EDIMAX IC-7100P

- PC with operational system Windows and Microsoft Visual Studio software,
- Router.

3 Mathematical model used in software

In this chapter are presented formulas and calculations used in the software made in this project.

3.1 Calibration

In this point are described all the operations executed before determining the position of the object position, such as calibration and coordinates correction. In some parts of this chapter are presented exemplary results of calibration.

3.1.1 Position and orientation of camera

In this point are presented calculations of position and orientation coordinates of cameras set in two different angles to the surface of calibration template from fig.1. Camera coordinate system x_c y_c z_c is associated with center of the center of camera matrix. Camera settings are presented in the fig.3.

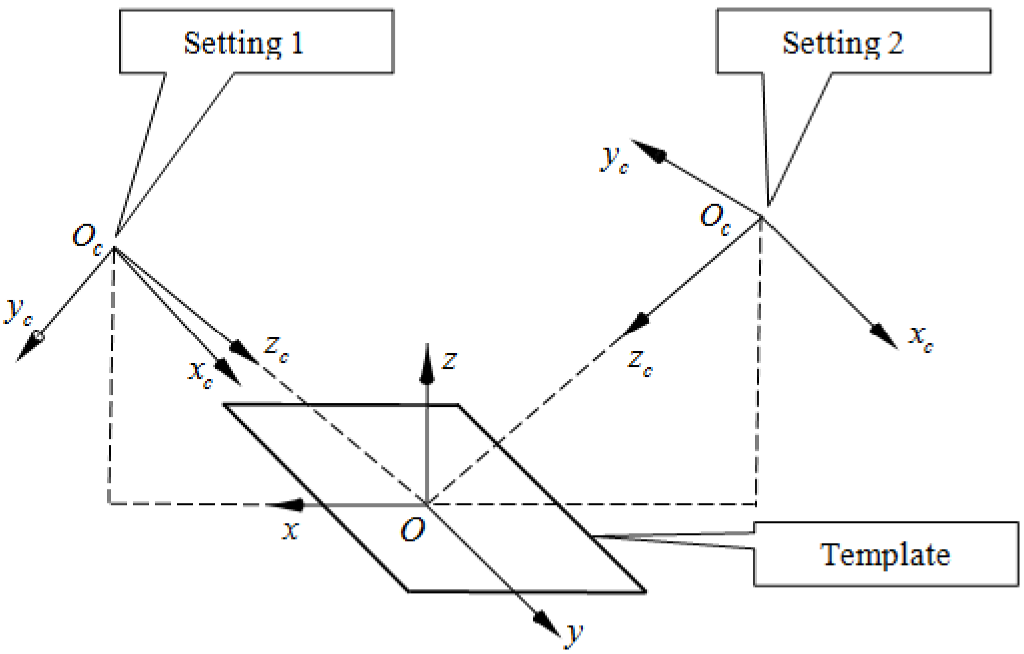


Fig.3. Camera settings

Images of template at setting 1 and 2 are presented in the fig.4a and fig.4b respectively.

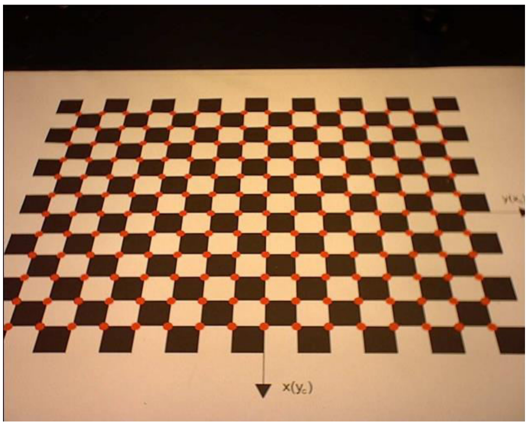


Fig.4a. Image - setting1

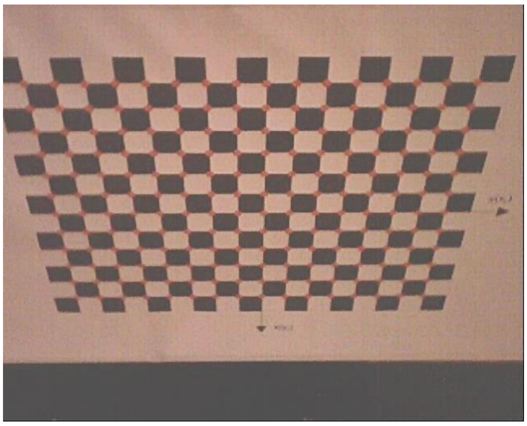


Fig.4b. Image - setting2

Coordinates of all characteristic points form template are obtained in pixels by software which is wider described in part 5 of this work. Using datasheet of used cameras Edimax IC-7100P results focal length f_c = 5.01 *mm* pixel size $2.8 \cdot 10^{-3}$ *mm* x $2.8 \cdot 10^{-3}$ *mm.* When we multiply coordinates in pixel by proper pixel size we obtain read coordinates from image in *mm* in camera system $x_c\ y_c$. Coordinates of these can be easily calculated in the reference system xy , because they are located in the vertices of squares with known side length.

Homogeneous matrix $\mathbf{T}_c$ of transformation described in equation (1) is used to represent position and orientation of camera system $x_c\ y_c\ z_c$ with respect to reference system xyz

$$\mathbf{T}_c = Trans(d_x, d_y, d_z) Rot(z,\gamma) Rot(y,\beta) Rot(x,\alpha) \quad \textbf{(1)}$$

This notation represents successive transformations with respect to reference system *xyz.* Mentioned before transformations are: rotation around axis x by angle α, rotation around axis y by angle β, rotation around axis z by angle γ, displacement d_z along axis z, displacement d_y along axis y and displacement d_x along axis x [5,6,7].

Algorithm *Camera* [4] is used to determine coordinates. Input parameters needed by this algorithm are, mentioned before, coordinates α , β , γ , d_x , d_y , d_z; coordinates of points A, B, C from template: ${}^cz_A$, ${}^cz_B$, ${}^cz_C$ in system $x_c\ y_c\ z_c$, also ${}^cx_{Ac}$, ${}^cy_{Ac}$, ${}^cx_{Bc}$, ${}^cy_{Bc}$, ${}^cx_{Cc}$, ${}^cy_{Cc}$ in system $x_c\ y_c\ z_c$ (read from camera) and x_A , y_A , z_A , x_B , y_B , z_B , x_C , y_C , z_C in system *xyz*. Aside from these coordinates, focal length f_c and accuracy of calculations *delta* are needed.

Calculations of camera coordinates for setting presented in fig.3 are done by method *kalibracja*. This method created 5056 sets of points A,B,C. These sets were created by mentioned method from 24 points from template. Calculated camera coordinates for these sets were averaged. Selected points where taken outside of origin of coordinate system O_c, but in the closest neighborhood. Read coordinates of these points are burdened by errors caused by optical distortions. These points are located on the squares with side two and four times greater than length of side of basic square in the calibration template fig.1 with centers approximately overlapped with point O_c. Coordinates $z_A = z_B = z_C = 0$ *mm* . Fig.5 presents these points.

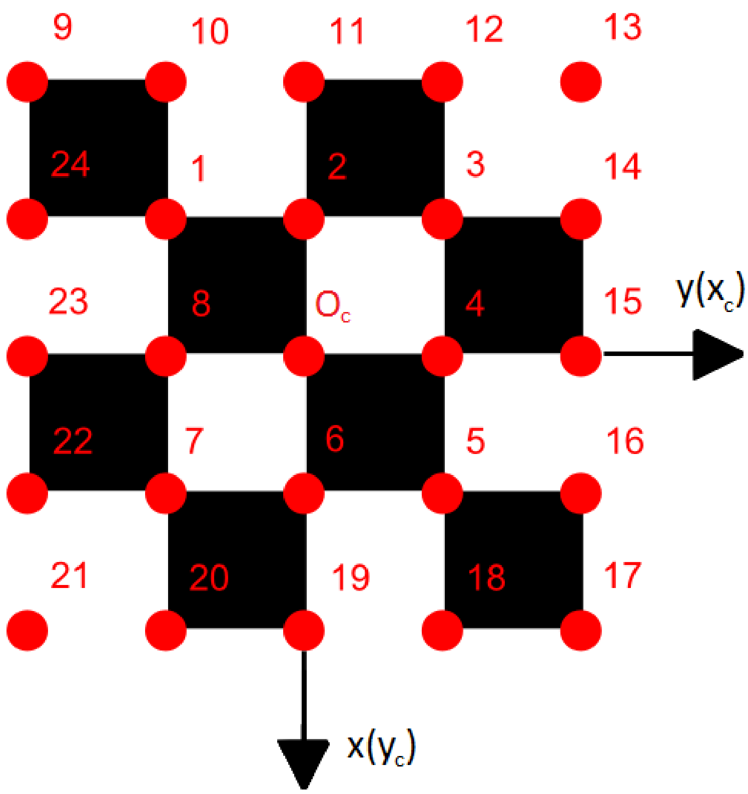


Fig.5. 24 used to generate 5056 sets

Camera coordinates α , β , γ , d_x , d_y , d_z were calculated for mentioned before input parameters with accuracy $delta$=10^{-6} *mm* [4].

Coordinates of position and orientation were calculated for camera setting 1 and 2. Values of these coordinates on account of high accuracy of type double in C# language, in order to presentation were rounded to 4 decimal places. Averaged coordinates for respectively setting 1 and 2 are represented by equations 2a and 3a, matrix $\mathbf{T}_c$ is presented by equations 2b and 3b.

Camera1:

$$\alpha = 225.8552^{\circ}, \ \beta = 1.7101^{\circ}, \ \gamma = 91.3053^{\circ}, \ d_x = 412.3035 \ mm, \ d_y = 19{,}5456 \ mm,$$

$$d_z = 399.7759 \ mm. \tag{2a}$$

$$\mathbf{T}_c = \begin{bmatrix} -0.0138 & 0.70643 & -0.7077 & 412.1308 \\ 0.9997 & -0.0031 & -0.0226 & 11.7709 \\ -0.0182 & -0.7078 & -0.7062 & 410.7867 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2b}$$

Camera2:

$$\alpha = 132.5487^{\circ}, \ \beta = -1.5697^{\circ}, \ \gamma = 91.2903^{\circ}, \ d_x = -456.4978 \ mm,$$

$$d_y = -22.5570\,mm,\ d_z = 433.8042\,mm. \tag{3a}$$

$$\mathbf{T}_c = \begin{bmatrix} -0.0157 & 0.6639 & 0.7476 & -460.2657 \\ 0.9997 & -0.0033 & 0.0237 & -15.32451 \\ 0.0183 & 0.7478 & -0.6637 & 422.90944 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3b}$$

3.1.2 Errors of optical distortions

In order to simplify description of characteristic points of template we apply concept of rows and columns. Rows are points located on lines parallel to axis *y* presented in fig.1. As we can see in the template are located 12 rows where each has 16 points. Columns are points located on lines parallel to axis *x*. Analogically in the template exists 16 columns with 12 points each. Point P_{ij} belongs to *i* row and *j* column. Coordinates read from camera matrix ${}^{c}x_c(i, j)$ and ${}^{c}y_c(i, j)$ of image of points P_{ij} in coordinate system x_c y_c are burdened by error caused by optical distortions $\Delta_*{}^{c}xc(i, j)$ and $\Delta_*{}^{c}y_c(i, j)$, which result from mathematical description of distortion described by factors k_1 , k_2 , k_3 , p_1 *i* p_2 [2,3]. These errors and parameters are represent by equations (4a) and (4b). Using coordinates read from camera andd equation (4c) we can calculate errors $\Delta^{c}x_c(i, j)$ i $\Delta^{c}y_c(i, j)$.

$$\Delta_*{}^{c}x_c(i,j)={}^{c}x_{ci}(i,j)[k_1{}^{c}r_{ci}(i,j)^2+k_2{}^{c}r_{ci}(i,j)^4+k_3{}^{c}r_{ci}(i,j)^6]+2p_1\cdot{}^{c}x_{ci}(i,j)\cdot{}^{c}y_{ci}(i,j)$$
$$+\,p_2[{}^{c}r_{ci}(i,j)^2+2\cdot{}^{c}x_{ci}(i,j)^2], \tag{4a}$$

$$\Delta_*{}^{c}y_c(i,j)={}^{c}y_{ci}(i,j)[k_1{}^{c}r_{ci}(i,j)^2+k_2{}^{c}r_{ci}(i,j)^4+k_3{}^{c}r_{ci}(i,j)^6]+2p_2\cdot{}^{c}x_{ci}(i,j)\cdot{}^{c}y_{ci}(i,j)$$
$$+\,p_1[{}^{c}r_{ci}(i,j)^2+2\cdot{}^{c}y_{ci}(i,j)^2], \tag{4b}$$

$$\Delta^{c}x_c(i,j)={}^{c}x_c(i,j)-{}^{c}x_{ci}(i,j),\ \Delta^{c}y_c(i,j)={}^{c}y_c(i,j)-{}^{c}y_{ci}(i,j),$$
$${}^{c}r_{ci}(i,j)^2={}^{c}x_{ci}(i,j)^2+{}^{c}y_{ci}(i,j)^2. \tag{4c}$$

In equations (4a-c) occurs ideal coordinate ${}^{c}x_{ci}(i, j)$ and ${}^{c}y_{ci}(i, j)$ without influence of optical distortions. These coordinates can be calculated using homogenous form $\mathbf{r}(i, j)$ of vector, which describes P_{ij} in reference system.

$$\mathbf{r}(i,j) = \begin{bmatrix} x(i,j) \\ y(i,j) \\ z(i,j) \\ 1 \end{bmatrix}. \tag{5}$$

Coordinates which occurs in the equation (5), can be presented using indexes of rows and columns in the following way: $x(i, j) = (i - 7)\cdot$*length of side of square in template in mm* and *y(i, j) = (j - 9)· length of side of square in template in mm.* Beginning of the reference system O is point P_{79}. We assume, that $z(i, j) = 0$. Using matrix $\mathbf{T}_c$ we can calculate the homogenous form of vector $^c\mathbf{r}(i, j)$ describing point P_{ij} in camera system $x_c\, y_c\, z_c$.

$$\mathbf{r}(i,j)=\begin{bmatrix} x(i,j) \\ y(i,j) \\ z(i,j) \\ 1 \end{bmatrix}=\mathbf{T}_c\,{}^c\mathbf{r}\rightarrow {}^c\mathbf{r}(i,j)=\begin{bmatrix} {}^c x(i,j) \\ {}^c y(i,j) \\ {}^c z(i,j) \\ 1 \end{bmatrix}=\mathbf{T}_c^{-1}\begin{bmatrix} x(i,j) \\ y(i,j) \\ z(i,j) \\ 1 \end{bmatrix}. \tag{6}$$

Using coordinate $^c x(i, j)$ from equation (6) we can calculate value of ideal coordinate $^c x_{ci}(i, j)$. Using geometrical dependencies presented in fig.6 we can delineate dependency (7a) describing mentioned before ideal coordinate.

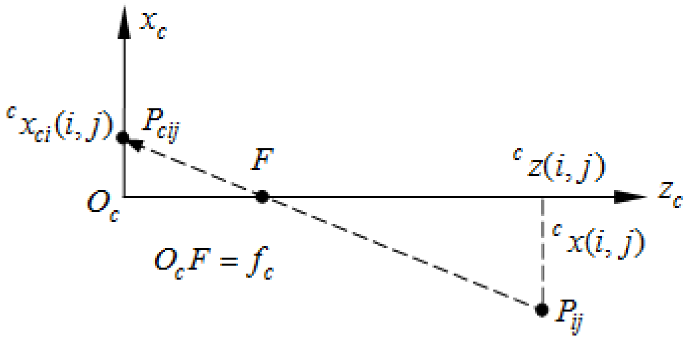


Fig.6. Geometrical dependencies P_{ij} and his image P_{cij}

$$\frac{{}^c x_{ci}(i,j)}{f_c}=\frac{-{}^c x(i,j)}{{}^c z(i,j)-f_c}\rightarrow {}^c x_{ci}(i,j)=-\frac{{}^c x(i,j)}{\dfrac{{}^c z(i,j)}{f_c}-1}. \tag{7a}$$

Analogically we can delineate equation (7b) describing second ideal coordinate $^c y_{ci}(i, j)$.

$${}^c y_{ci}(i,j)=-\frac{{}^c y(i,j)}{\dfrac{{}^c z(i,j)}{f_c}-1}. \tag{7b}$$

Using mentioned equations (6), (7a), (7b) we are able to calculate $\Delta^c x_c(i, j)$, $\Delta^c y_c(i, j)$ and $^c r_c(i, j)$ necessary in equations (4c). Now we can use equations (4a) and (4b) to calculate factors k_1 , k_2 , k_3 , p_1 and $p2$. Using sum presented in equation (8), we are able to calculate these factors using minimum square method. Results of this method for exemplary data are presented in equations (9a-10d) for proper camera setting presented in fig.3

$$S=\sum_{i=1}^{12}\sum_{j=1}^{16}\{[\Delta^c x_c(i,j)-\Delta_*{}^c x_c(i,j)]^2+[\Delta^c y_c(i,j)-\Delta_*{}^c y_c(i,j)]^2\} \tag{8}$$

Camera setting 1:

$$k_1 = 0.0007\ mm^{-2}, k_2 \div k_3 = 0,\ p_1 \div p_2 = 0; \quad \textbf{(9a)}$$

$$k_1 = -0.0014\ mm^{-2},\ k_2 = 0.0009\ mm^{-4},\ k_3 = 0,\ p_1 \div p_2 = 0; \quad \textbf{(9b)}$$

$$k_1 = -0.0013\ mm^{-2},\ k_2 = 0.0009\ mm^{-4},\ k_3 = 0,\ p_1 \div p_2 = 0; \quad \textbf{(9c)}$$

$$k_1 = -0.0007\ mm^{-2},\ k_2 = 0.0004\ mm^{-4},\ k_3 = 0.0001\ mm^{-6},\ p_1 = -0.0008\ mm^{-2},$$

$$p_2 = -0.0002\ mm^{-2}. \quad \textbf{(9d)}$$

Camera setting 2:

$$k_1 = -0.0036\ mm^{-2}, k_2 \div k_3 = 0,\ p_1 \div p_2 = 0; \quad \textbf{(10a)}$$

$$k_1 = 0.0015\ mm^{-2},\ k_2 = -0.0021\ mm^{-4},\ k_3 = 0,\ p_1 \div p_2 = 0; \quad \textbf{(10b)}$$

$$k_1 = 0.0044\ mm^{-2},\ k_2 = -0.0050\ mm^{-4},\ k_3 = 0.0006\ mm^{-6},\ p_1 \div p_2 = 0; \quad \textbf{(10c)}$$

$$k_1 = 0.0072\ mm^{-2},\ k_2 = -0.0046\ mm^{-4},\ k_3 = 0.0001\ mm^{-6},\ p_1 = -0.0058\ mm^{-2},$$

$$p_2 = -0.0002\ mm^{-2}. \quad \textbf{(10d)}$$

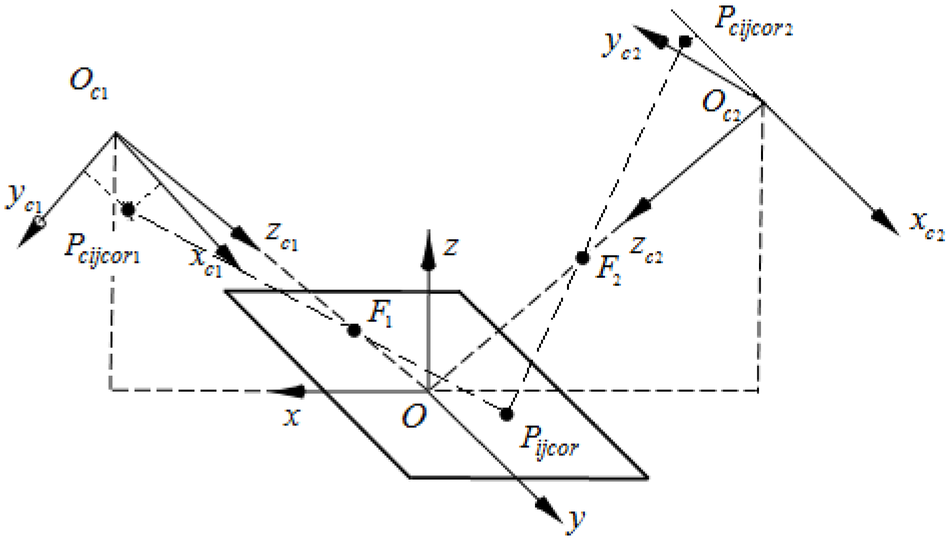
Presented equations describes errors using different amount of factor. Expressions (9a), (10a) - using one factor; (9b), (10b) - two; (9c), (10c) - three and (9d), (10d) - five.

3.1.3 Errors of coordinates calculations

Using factors described by expressions (9a)-(10d) we can calculate errors $\Delta_*{}^c xc(i, j)$ and $\Delta_*{}^c y_c(i, j)$ represented by equations (4a) and (4b). When we calculate those error we can use correction of coordinates of points ${}^c x_c(i, j)$ and ${}^c y_c(i, j)$. This correction works in following way: the values of mentioned errors $\Delta_*{}^c xc(i, j)$ and $\Delta_*{}^c y_c(i, j)$ are subtracted from read coordinates ${}^c x_c(i, j)$ and ${}^c y_c(i, j)$. New corrected coordinates are marked as ${}^c x_{ccor}(i, j)$ and ${}^c y_{ccor}(i, j)$, these coordinate are represented by equation (11).

$${}^c x_{ccor}(i,j) = {}^c x_c(i,j) - \Delta_*{}^c x_c(i,j),\ \ {}^c y_{ccor}(i,j) = {}^c y_c(i,j) - \Delta_*{}^c y_c(i,j) \quad \textbf{(11)}$$

Using corrected coordinates we are able to calculate coordinates of points in reference system *xyz*. Mark these coordinates as: $x_{cor}(i, j)$, $y_{cor}(i, j)$ i $z_{cor}(i, j)$. These coordinates describes point P_{ijcor} presented in fig.7.



$$P_{cijcor}({}^{c}x_{ccor}(i,j), {}^{c}y_{ccor}(i,j))$$

$$P_{ijcor}(x_{cor}(i,j), y_{cor}(i,j))$$

Fig.6. Coordinates x of points P_{ijcor} and P_{cijcor}

To calculate coordinates $x_{cor}(i, j)$, $y_{cor}(i, j)$ and $z_{cor}(i, j)$ we need coordinates of point P_{cijcor}: $x_{ccor}(i, j)$, $y_{ccor}(i, j)$, $z_{ccor}(i, j)$, and also x_F , y_F , z_F of focuses F in reference system. It is worth to mention, that $O_{c1}F_1=f_{c1}$ and $O_{c2}F_2=f_{c2}$ where f_{c1} and f_{c2} are the focal length of camera. Equations (12) describes these coordinates:

$$P_{F1=}\begin{bmatrix} x_{F1} \\ y_{F1} \\ z_{F1} \\ 1 \end{bmatrix} = \mathbf{T}_{c1}\begin{bmatrix} 0 \\ 0 \\ f_{c1} \\ 1 \end{bmatrix}, \; P_{F2} = \begin{bmatrix} x_{F2} \\ y_{F2} \\ z_{F2} \\ 1 \end{bmatrix} = \mathbf{T}_{c2}\begin{bmatrix} 0 \\ 0 \\ f_{c2} \\ 1 \end{bmatrix}. \qquad \textbf{(12)}$$

Equations (13),(14) describes respectively lines passing through points $P_{cijcor1}$, F_1 , P_{ijcor} and $P_{cijcor2}$, F_2 , P_{ijcor} .

$$\frac{x_{cor}(i,j) - x_{ccor1}(i,j)}{x_{F1} - x_{ccor1}(i,j)} = \frac{y_{cor}(i,j) - y_{ccor1}(i,j)}{y_{F1} - y_{ccor1}(i,j)} = \frac{z_{cor}(i,j) - z_{ccor1}(i,j)}{z_{F1} - z_{ccor1}(i,j)}. \qquad \textbf{(13)}$$

$$\frac{x_{cor}(i,j) - x_{ccor2}(i,j)}{x_{F2} - x_{ccor2}(i,j)} = \frac{y_{cor}(i,j) - y_{ccor2}(i,j)}{y_{F2} - y_{ccor2}(i,j)} = \frac{z_{cor}(i,j) - z_{ccor2}(i,j)}{z_{F2} - z_{ccor2}(i,j)}. \qquad \textbf{(14)}$$

Mentioned equations allows to create systems of three linear equations with three unknowns $x_{cor}(i,j)$, $y_{cor}(i,j)$, $z_{cor}(i,j)$. There can be 6 of such systems. Created method *odl0* calculates these equations. Accuracy of calculating coordinates on plane of template characteristic points can be described by absolute values of distance $\Delta r(i,j) =$

$\sqrt{\Delta x(i,j)^2 + \Delta y(i,j)^2}$, where $\Delta x(i,j) = | x(i,j) - x_{cor}(i,j) |$, $\Delta y(i,j) = | y(i,j) \; - y_{cor}(i,j) |$ and $\Delta z(i,j) = | z(i,j) - z_{cor}(i,j) |$ Results for exemplary data are presented below.

For setting 1 of camera 1 from fig. 3 and different set of factors obtained:

- $\max[\Delta r(i,j)] = \Delta r(2,1) = 1.2955\ mm$ for factors described by equations (9a),
- $\max[\Delta r(i,j)] = \Delta r(2,1) = 1.2947\ mm$ for factors described by equations (9b),
- $\max[\Delta r(i,j)] = \Delta r(1,3) = 0.7303\ mm$ for factors described by equations (9c),
- $\max[\Delta r(i,j)] = \Delta r(1,3) = 0.7303\ mm$ for factors described by equations (9d).

For setting 2 of camera 2 from fig. 3 and different set of factors obtained:

- $\max[\Delta r(i,j)] = \Delta r(12,16) = 4.5363\ mm$ for factors described by equations (10a),
- $\max[\Delta r(i,j)] = \Delta r(12,16) = 4.4833\ mm$ for factors described by equations (10b),
- $\max[\Delta r(i,j)] = \Delta r(10,1) = 2.5806\ mm$ for factors described by equations (10c),
- $\max[\Delta r(i,j)] = \Delta r(10,1) = 2.5806\ mm$ for factors described by equations (10d).

It is easy to notice, that accuracy of calculations of coordinates grows with the number of used factors of model of optical errors (4a) and (4b). Using just one factor the biggest errors occurs for both cameras, while using all five factor the values of errors were lowest.

3.2 Position of manipulated object

In this point calculations used in program *Object Coordinates* in order to obtain coordinates of position of object of manipulation are presented. Picture of object is presented in fig. 7 while scheme pointing its size can be found in fig.8.

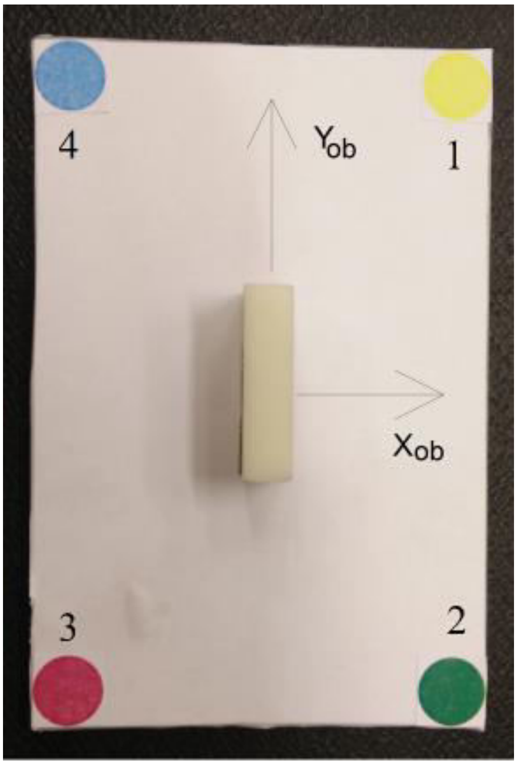


Fig.7. Object of manipulated with marked axes and characteristic points

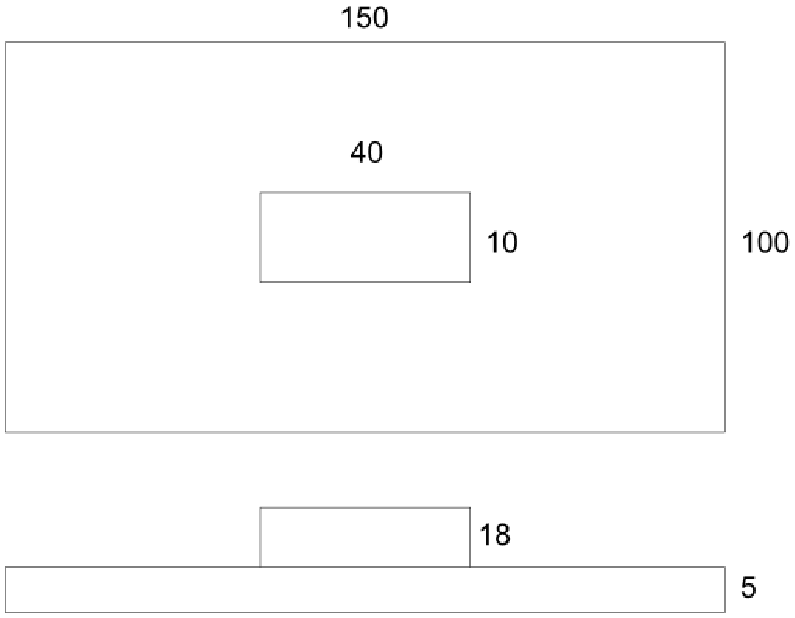


Fig.8. Size of object of manipulation

Input parameters for calculations in mentioned program are coordinates $x_{cor}, y_{cor}, z_{cor}$ of any three characteristic points of object, calculated using mentioned before method *odl0*. Mark calculated vectors of coordinates of appropriate points as $\vec{r}_1$, $\vec{r}_2$, $\vec{r}_3$ and $\vec{r}_4$, while distances between points are represented by vectors presented in equations (15a-d). Afterwards unit vectors $\vec{a}$, $\vec{b}$ are calculated according to equations (16) and (17).

$$\vec{r}_{12} = \vec{r}_1 - \vec{r}_2 , \quad \textbf{(15a)}$$

$$\vec{r}_{43} = \vec{r}_4 - \vec{r}_3 , \quad \textbf{(15b)}$$

$$\vec{r}_{14} = \vec{r}_1 - \vec{r}_4 , \quad \textbf{(15c)}$$

$$\vec{r}_{23} = \vec{r}_2 - \vec{r}_3 , \quad \textbf{(15d)}$$

$$\vec{a} = \frac{\vec{r}_{14}}{\|\vec{r}_{14}\|} = \frac{\vec{r}_{23}}{\|\vec{r}_{23}\|} , \quad \textbf{(16)}$$

$$\vec{b} = \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|} = \frac{\vec{r}_{43}}{\|\vec{r}_{43}\|} . \quad \textbf{(17)}$$

Like we can see unit vector $\vec{a}$ is parallel to axis x_{ob} of object, while unit vector $\vec{b}$ is parallel to axis y_{ob}. Using equations (18) unit vector $\vec{c}$ can be calculated.

$$\vec{c} = \vec{a} \times \vec{b} \quad \textbf{(18)}$$

Last result vector, is a vector of coordinates of the origin of object coordinate system $\vec{d}$. This vector depending on available points and vectors can be calculated in several ways. Exemplary solutions are presented by equation (19).

$$\vec{d} = \vec{r}_1 - \frac{1}{2}\vec{r}_{12} - \frac{1}{2}\vec{r}_{14} = \vec{r}_2 + \frac{1}{2}\vec{r}_{12} - \frac{1}{2}\vec{r}_{23} = \vec{r}_3 + \frac{1}{2}\vec{r}_{12} + \frac{1}{2}\vec{r}_{23} . \quad \textbf{(19)}$$

Final result of program is present as matrix of transformation **T**.

$$\mathbf{T} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} . \quad \textbf{(20)}$$

4 Software description

In this chapter software created within this project is presented. Two programs are described one *Object Coordinates* which mechanics were presented in previous chapter and program *Camera Coordinates* used for obtaining coordinates *x* and *y* of characteristic points in pixels.

4.1 Program *Camera Coordinates* obtaining data

In this point program used instead of software created by Artur Meller [1] is described. Reasons for which his software was replaced with another one are presented in chapter 5. This program has been written in Microsoft Visual Studio 2013 environment, using C# with .NET platform with additional library AForge.NET.

4.1.1 Graphical user interface

Figure 9 presents graphical user interface of the program.

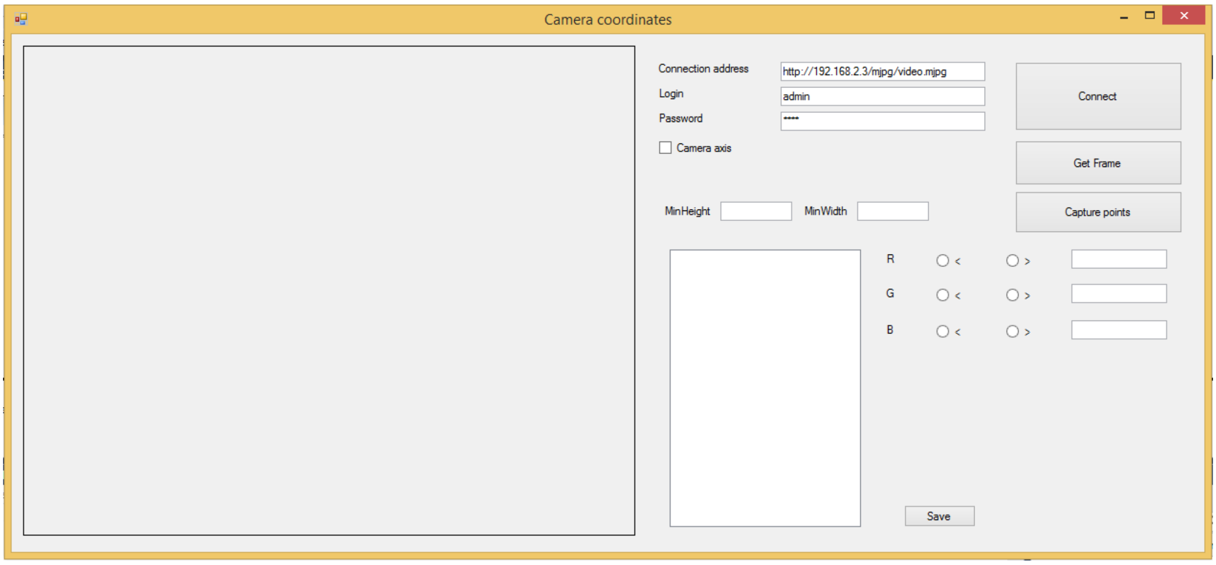


Fig.9. Program GUI

Program interface consists of several important parts. Left half of program is intended to displaying image from camera/ effect of binarization of image, in right top part are elements used to connection with camera, rest of the controls are responsible for settings used by binarization and displays/save coordinates of found points.

4.1.2 Method of operation

Creation of this program was not the main goal of this work, so this is simple software, which method of operation is presented below.

In order to use the program, first of all it is necessary to connect with camera. How it was mentioned before, elements in top right part of the program should be used (fig.10).

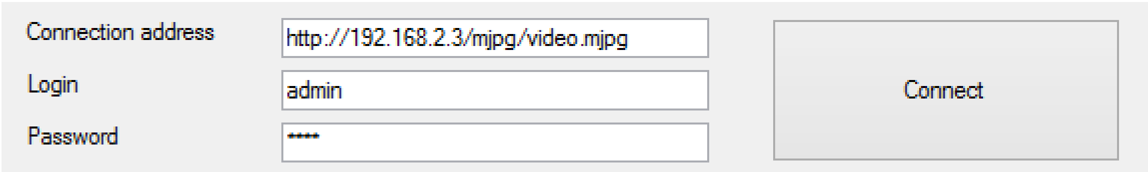


Fig.10. Connection with camera

After use of button "Connect" center of image from camera should be set approximately with center of reference system. In order to simplify this operation, control "Camera axis" can be used. When this controls is checked two axes appears on displayed image pointing center of image. If camera is set correctly "Get Frame" button should be used next. This button is used to capture frame for further manipulations. When it is done, appropriate values of RGB of searched color and "MinHeight", "MinWidth" filter parameters ,meaning minimal height and minimal width of objects, should be set. Filter is used to ignore unwanted object like single pixels. When all setting have proper value, button "Capture points" should be used which is responsible for binarization of image according to previously set RGB values and after that with use of AForge.NET library looks for characteristic points. In case of not proper amount of searched points (program has been specified for cooperation with program calulculatin coordinates of position and orientation) the message box with warning will appear. Examples of use are presented in chapter 5. If proper number of objects has been found, button "Save" should be used which saves data in text file.

4.2 Program *Object Coordinates* calculating coordinates of position and orientation

In this point is presented program which mechanics has been discussed in chapter 3.

4.2.1 Graphical user interface

In figures 11 and 12 the graphical user interface of the program is presented. Like you can see the interface is mostly split in 2 similar parts, each is responsible for another camera.

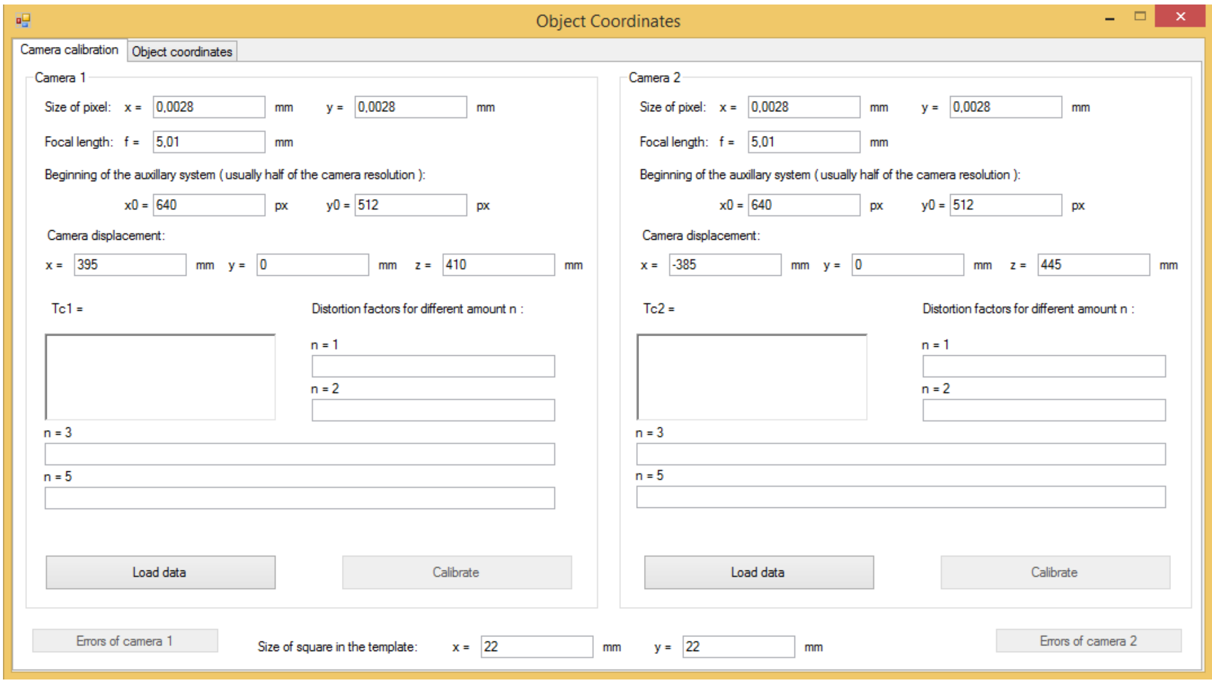


Fig.11.GUI of calibration part

To start working with this application, first the camera parameters (default values are set of cameras edimax IC-7100P), displacement (x of first camera should be positive value and of second camera should be negative) and information about template should be given. Size of squares means a distance between centers of characteristic points. Afterward the specially formatted text file should be loaded (2 rows with 192 columns of numeric values, separated by tab, with "," as decimal separator, where first row are the coordinate x and second row y of characteristic points, read from camera in pixels) using button "Load data", which is unblocking "Calibrate" button. After clicking "Calibrate" button camera will be calibrated, the matrix $\mathbf{T}_c$ of transformation and 1, 2, 3 and 5 distortion factors will be shown. Button "Errors of camera x" displays window with calculated values α , β , γ , d_x , d_y , d_z and accuracy analysis for 5 factors. After calibration of two cameras "Object coordinates" should be opened.

Similar to previous tab, interface is divided for two cameras separately. In this part the values of coordinates x and y of characteristic points of object in pixels should be given. Program needs just 3 of such points, but they should be given in pairs for two cameras, in opposite case program will display error.

After filling all field, button "Calculate" should be used, which is displaying result matrix (20).

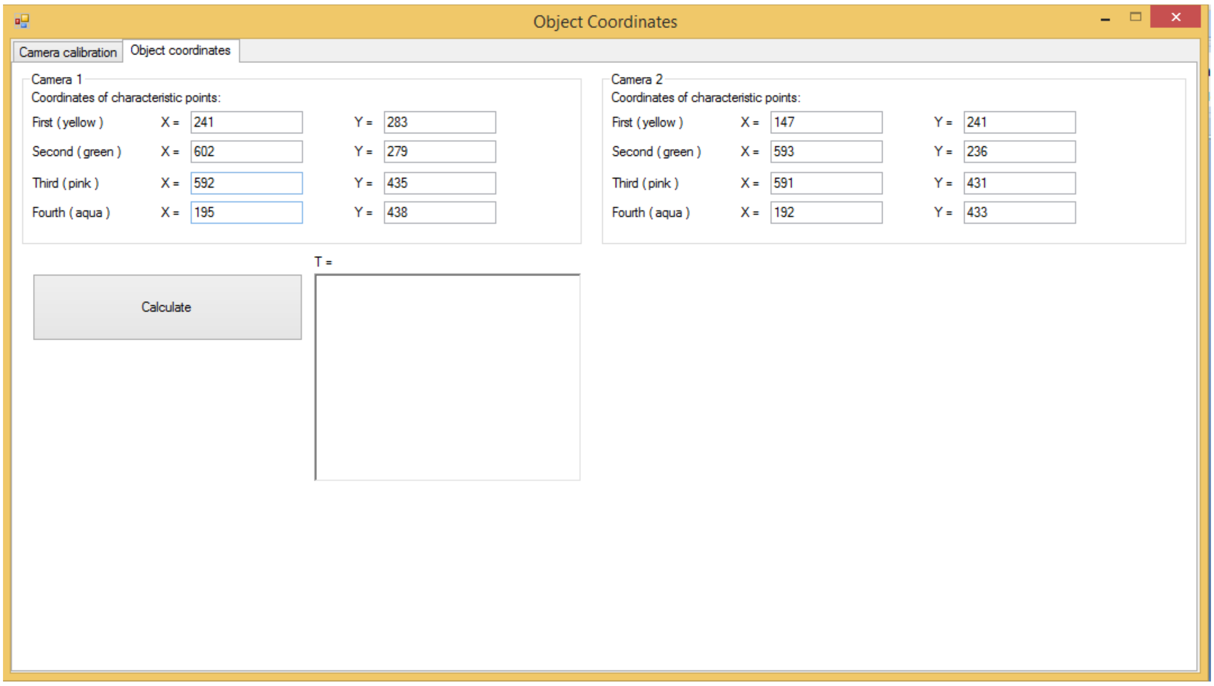


Fig.12.GUI of position calculating part

5 Application tests

In this chapter, tests of created, during this project, software has been described.

5.1 Program obtaining data - test

First test was attempt to binarization of image of characteristic points of calibration template. In the beginning software made by Artur Meller [1] was used. Unfortunately implemented method in this program does not allowed to simultaneously getting rid of unwanted objects and leave all characteristic points. In fig.13a and fig.13b are presented images from two cameras, respectively binarized images fig. 4a and 4b.

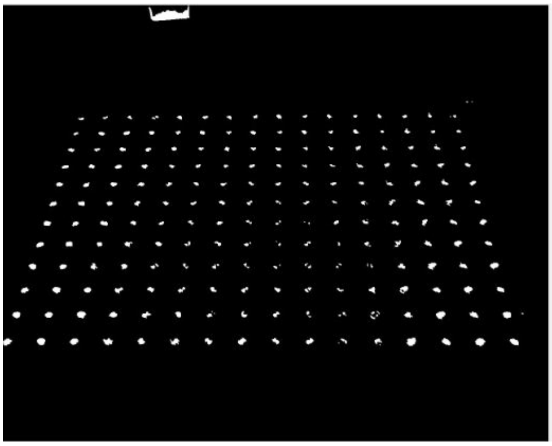


Fig.13a. Binaryzacja obrazu rys.4a

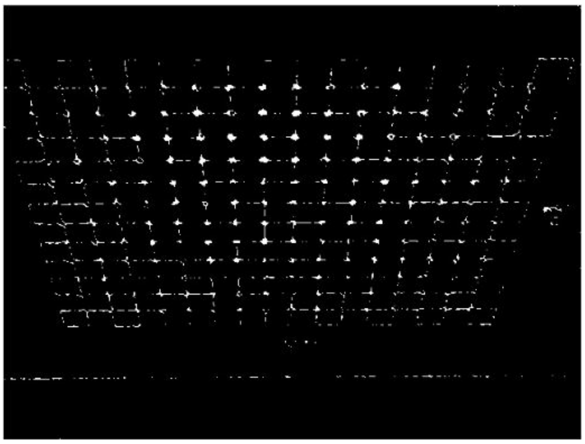


Fig.13b. Binaryzacja obrazu rys.4b

Like we can see above image, though the erosion and dilatation was used, contains a lot of unwanted objects and ,how can be seen fig. 13b where occurred slightly different illumination, some of characteristic points were omitted. This is reason why mentioned before program has been created, based on RGB values. Program is slower, but its results are better. In fig 14a and 14b binarized image is presented.

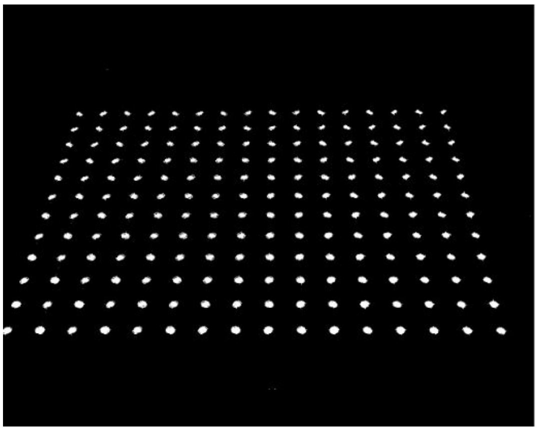


Fig.14a. Binaryzacja obrazu rys.4a

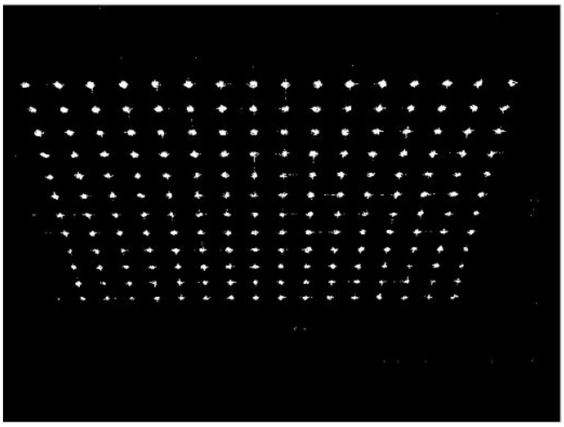
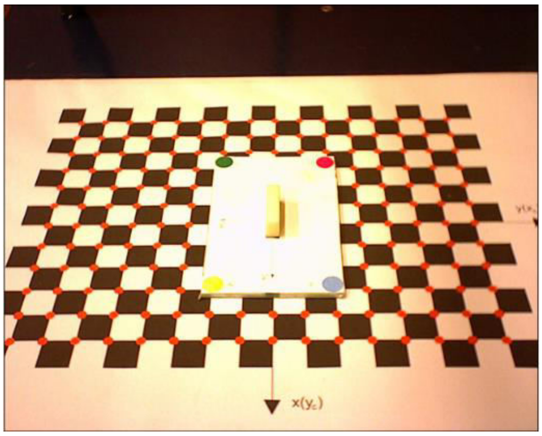


Fig.14b. Binaryzacja obrazu rys.4b

Used filter allows to omit small unwanted object like single pixels, allowed to obtain proper amount of points coordinates from these two images.

Next step of tests was checking capturing coordinates of characteristic points of the object. Unfortunately because of high influence of illumination on observed points, the automated process of obtaining data cannot be applied.

Figures below are presenting image from cameras and results of binarization of appropriate characteristic points.



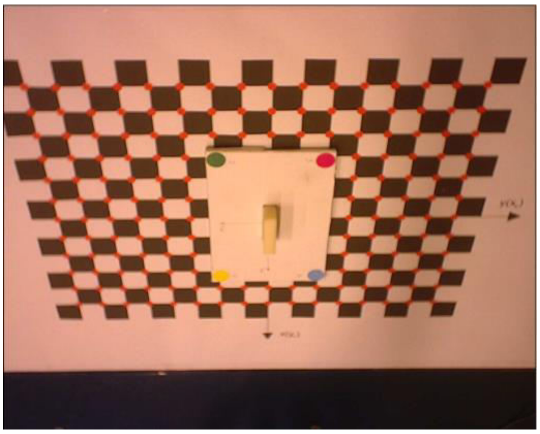


Fig.15a. View from camera 1

Fig.15b. View from camera 2

In next part, results of binarization of specified characteristic points of object are presented.

For point 1 (yellow):

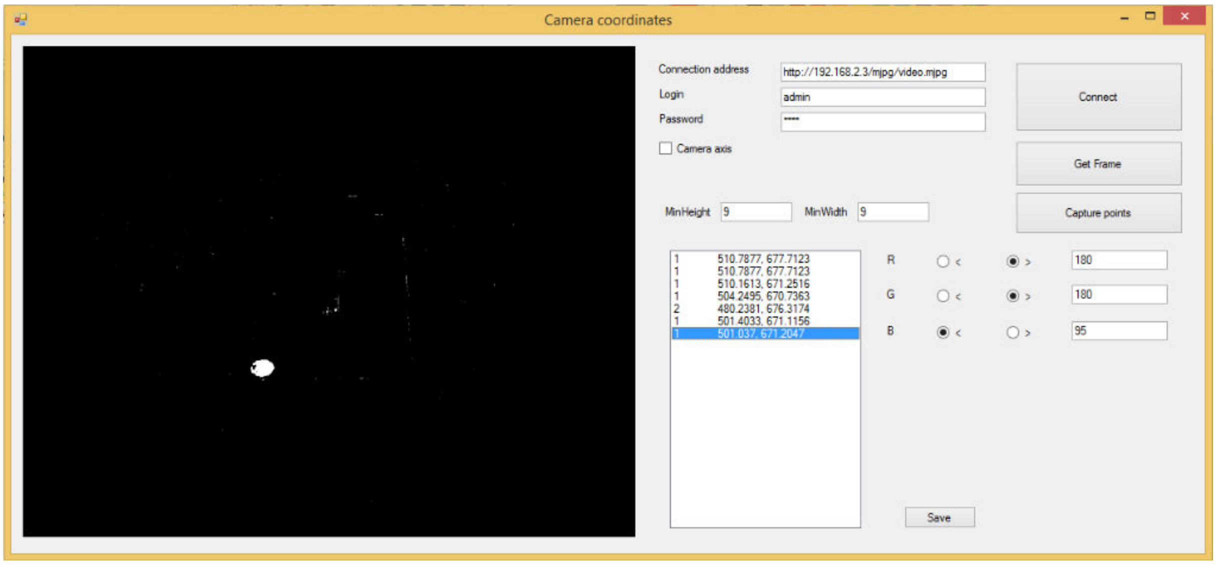


Fig.16a. Binarization of point 1 of image 15a

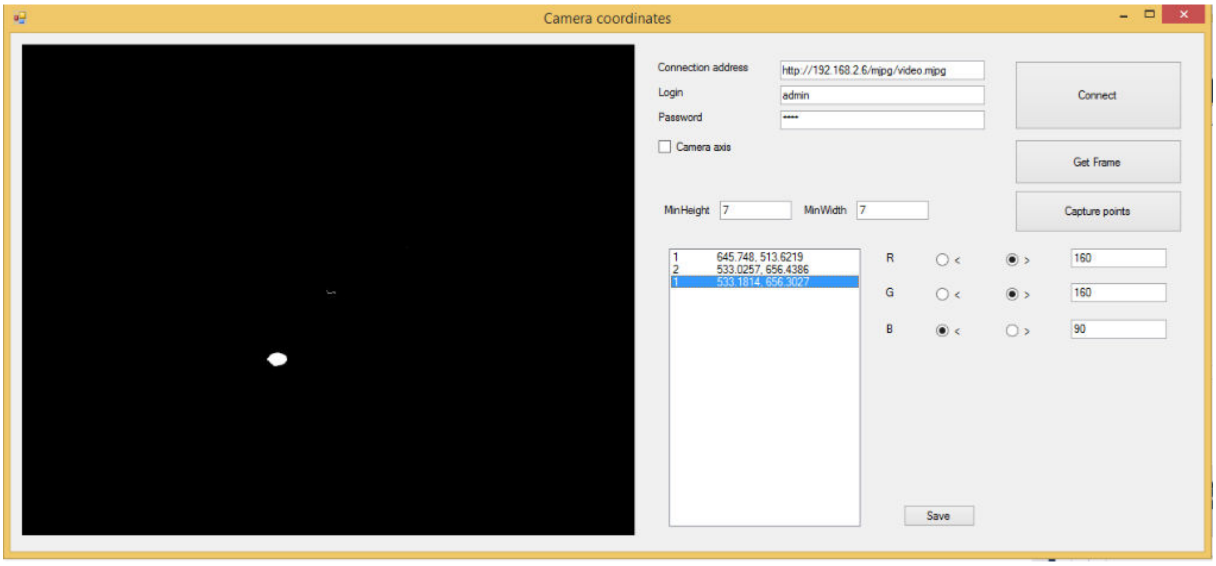


Fig.16b. Binarization of point 1 of image 15b

For point 2 (green):

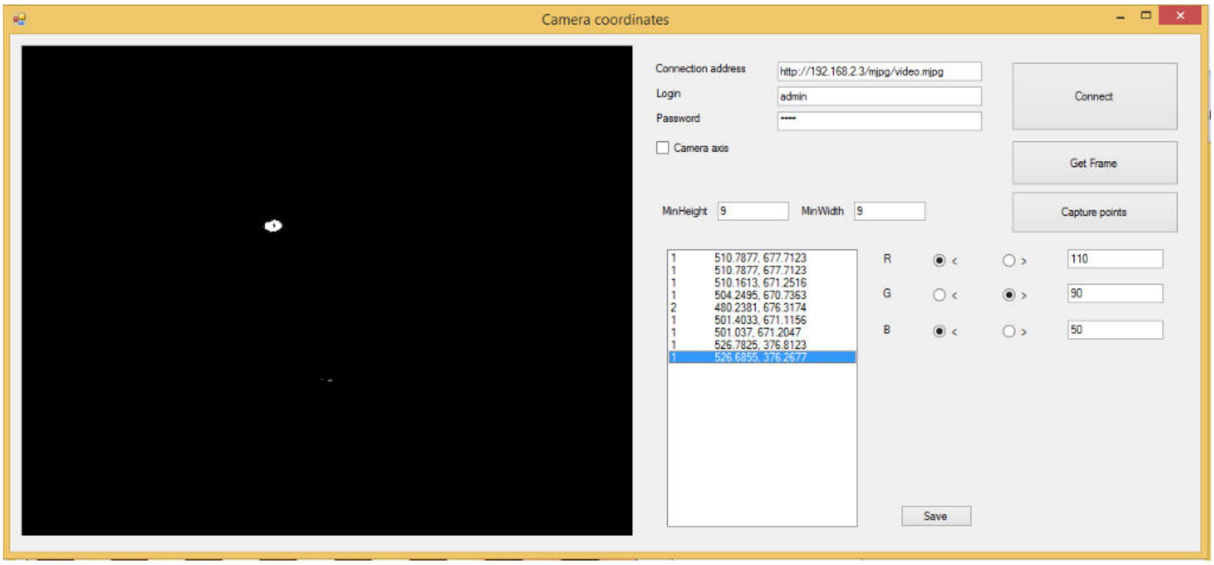


Fig.17a. Binarization of point 2 of image 15a

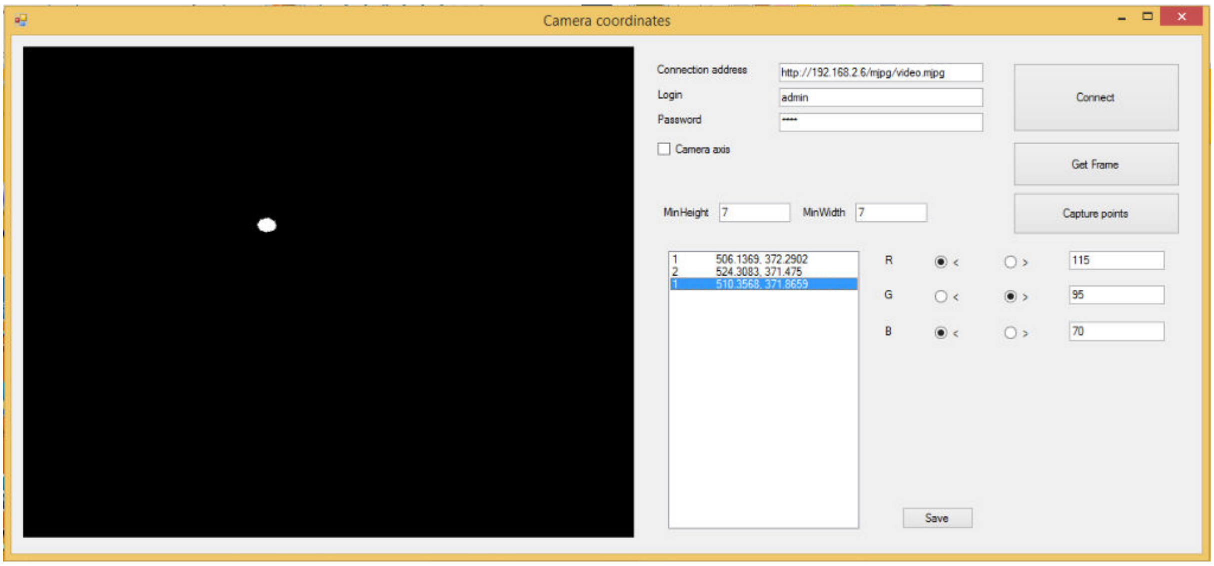


Fig.17b. Binarization of point 2 of image 15b

For point 3 (pink):

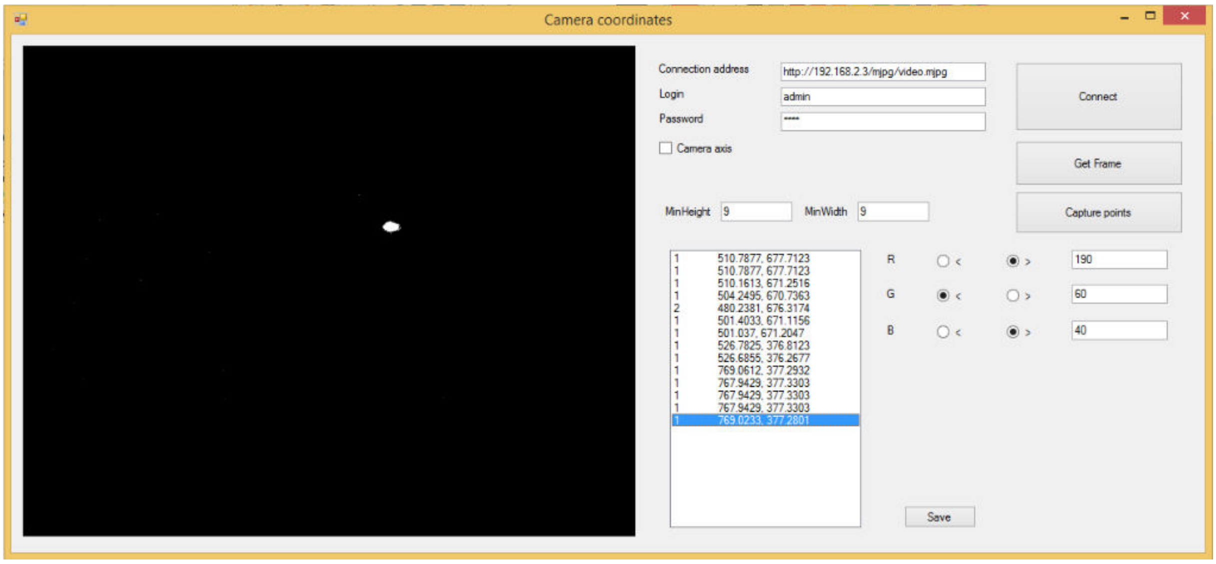


Fig.18a. Binarization of point 3 of image 15a

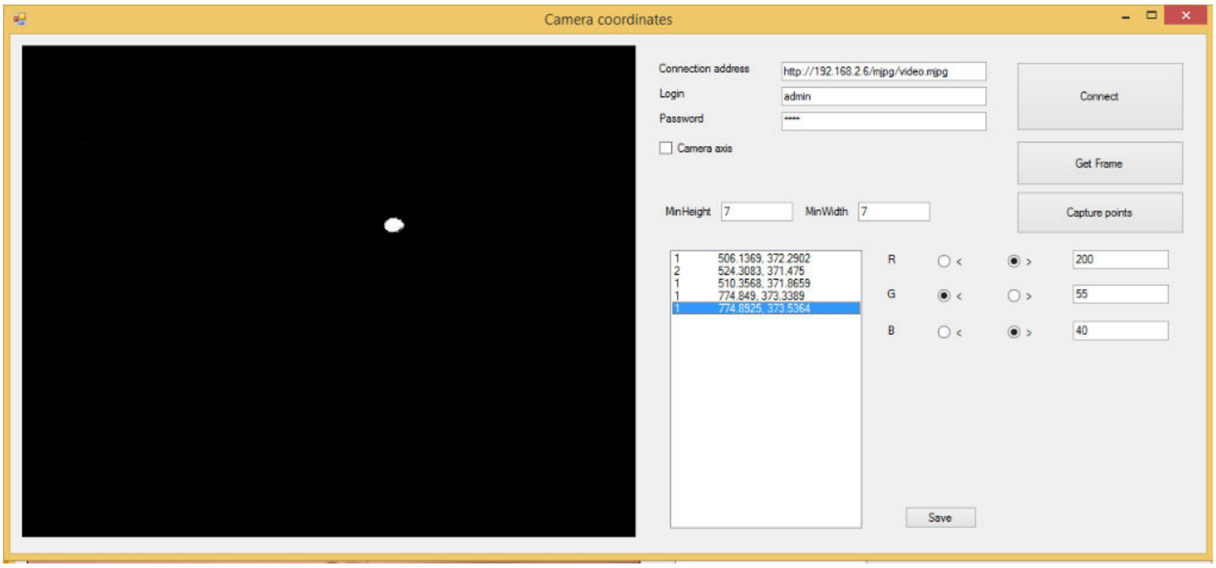


Fig.18b. Binarization of point 3 of image 15b

For point 4 (bright blue, aqua)

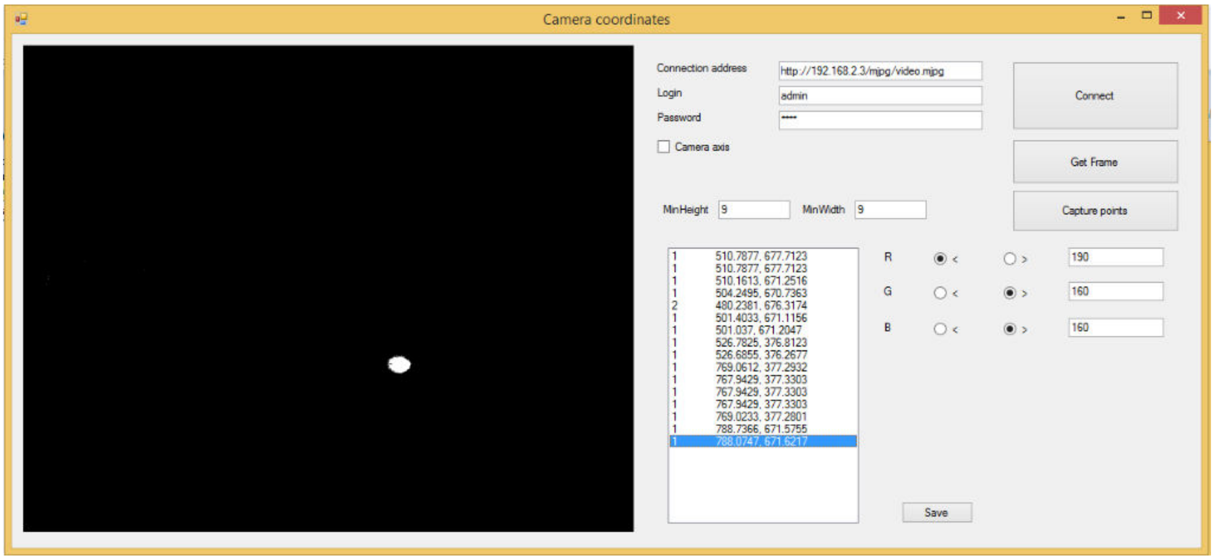


Fig.19a. Binarization of point 4 of image 15a

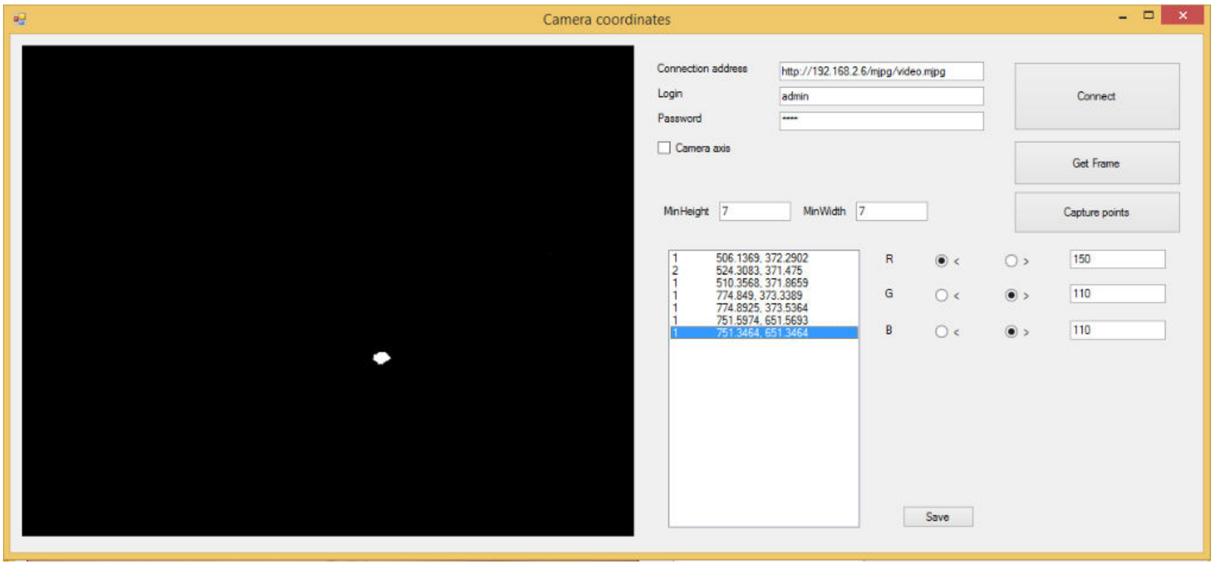


Fig.19b. Binarization of point 4 of image 15b

Like we can see in the figures above, using proper values of filters, good results can be obtained omitting unwanted objects. Unfortunately, how can be seen in the figures, effective values of RGB for the same colors obtained from 2 cameras are different. It is caused probably, because of illumination. Cameras are directed in different angles to template surface, which causes different reflection of light which varies brightness level of camera.

5.2 Program calculating coordinates of position and orientation - test

Next step was testing the main program *Object Coordinates* using data obtained from previous tests of first program *Camera Coordinates*.

5.2.1 Calibration

In this point tests of calibration part of mentioned program are presented. Here is presented one test of calibration for camera settings from fig. 15a and 15b (using image without object of manipulation). Results of test based on image in fig. 4a and 4b were presented as examples in chapter 3. Fig. 20 is presenting results of current test.

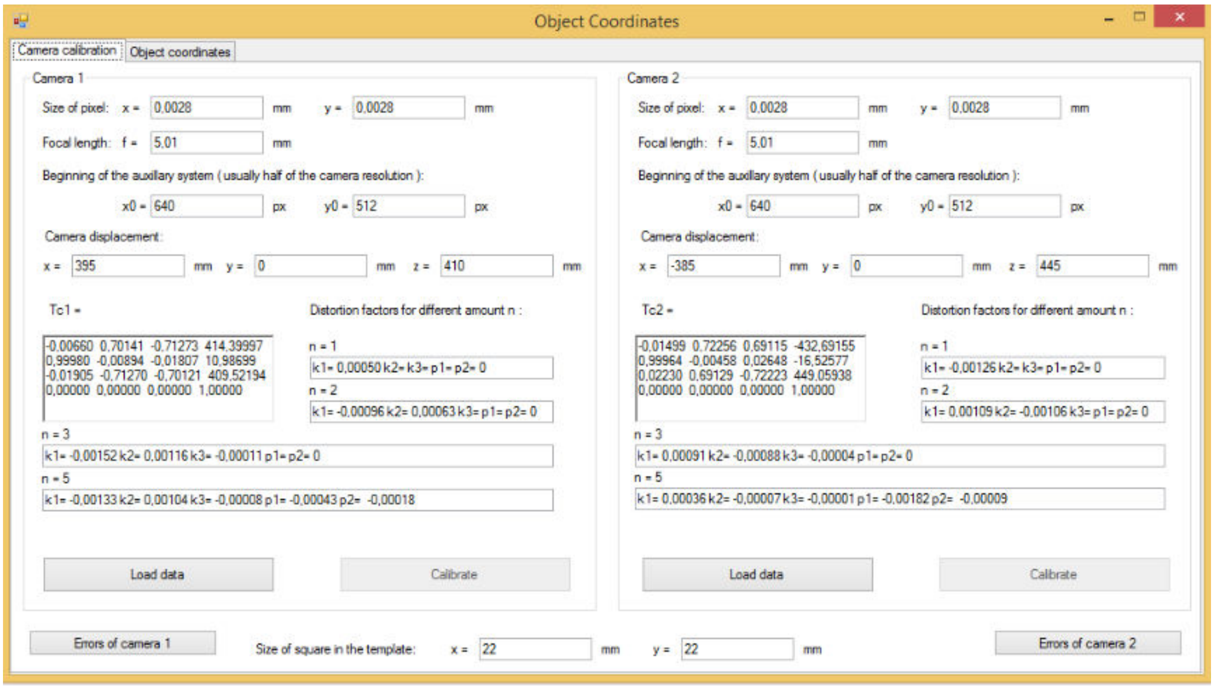
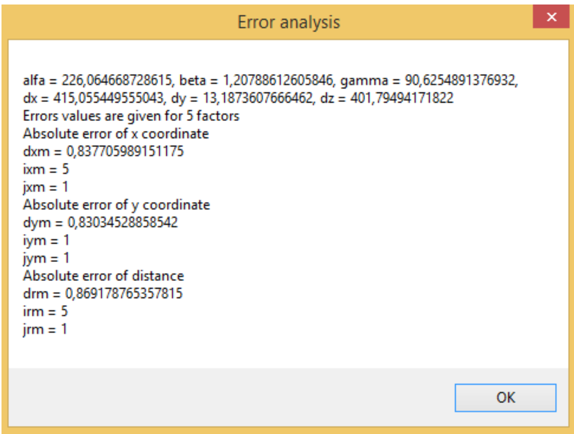


Fig.20. Results of calibration test

Software is calibrating camera and calculate 5 factors of optical distortions. In the fig.21a and 21b accuracy of calibration can be observed.



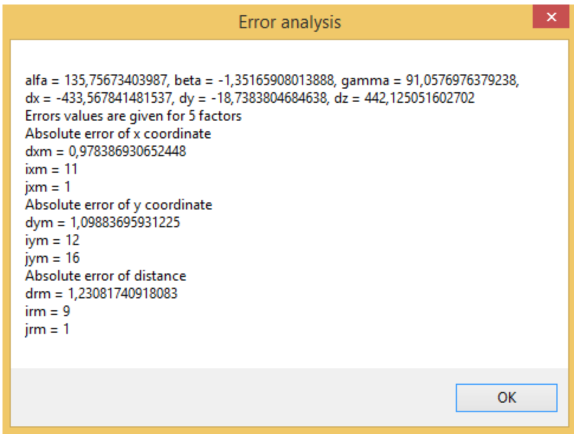


Fig.21a. Accuracy of calibration of camera 1 *Fig.21b. Accuracy of calibration of camera 2*

Like we can see, the greatest inaccuracy of calculation of points position on plane is equal around 1.23 *mm* what is a good result.

5.2.2 Object position

Taking into account specific position of object of manipulation, theoretical result table has been created and compared with program output. Theoretical matrix has been presented in equation (21).

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

Starting from last column: point d get such data, because center of object has been placed approximately in the center of template, value d_z=5 mm, because height of base of object has around 5 mm. Unit vector $\vec{c}$ gets such data, because has the same direction as axis z of reference system. Unit vector $\vec{b}$ is overlapping with axis x, while unit vector $\vec{a}$ is parallel to axis y, but is directed opposite, that is why the value in matrix is negative. In fig.22 results of the program has been presented.

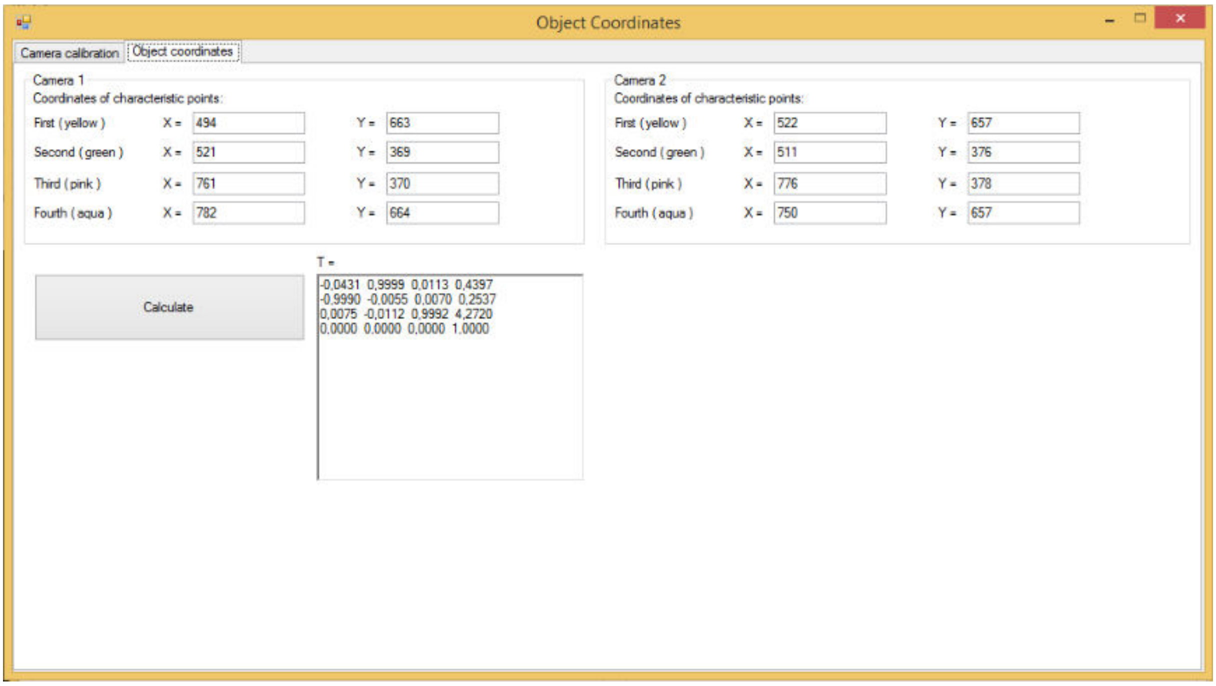


Fig.22. Program test results

How we can see result approximately is equal to our theoretical answer.

6 Summary

Nowadays vision systems determining position and detecting object are widely recognized in industry. Unfortunately notably such devices and software are generally highly specialized and because of that expensive. Presented solution specifying position of object can be used to all cameras with known focal length and pixel size, reaching fairly good accuracy.

Aim of the project has been realized with use of language C# in WinForms and AForge.NET library used to communication with camera and detection of characteristic objects.

7 References

[1] A.Meller *Integration of vision detection of manipulation object position software with Adept robot controller*.

[2] Beyer H.A.: *Geometric and Radiometric Analysis of CCD-Camera Based Photogrammetric Closed-Range System.* Dissertation, No. 9701, ETH, Zurich (1992)

[3] Google: Kalibracja kamery. etacar.put.poznan.pl/marcin Kiełczewski (2013) (in Polish)

[4] T. Szkodny *The Algorithm Camera Computing the Object Location.* Proc. ICIRA 2013, Part I, LNAI 8102, Springer-Verlag (2013)

[5] T. Szkodny : Podstawy robotyki. Wydawnictwo Politechniki Śląskiej. Gliwice, Poland (2012) (in Polish)

[6] Craig J. J.: Introduction to Robotics. 2-nd Edition. Addison Wesley Publ. Comp. (1986)

[7] T. Szkodny, A.Meller, K.Palka *Accuracy of Determining the Coordinates of Points Observed by Camera. International Conference on Intelligence Robotics and Applications. Guangzhou, China 2014.*