



**POLITECHNIKA ŚLĄSKA**  
**WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI**

Projekt inżynierski

Algorytmy wizyjnej detekcji pozycji obiektu manipulacji

Autor: Krzysztof Palka

Kierujący Pracą: dr hab. inż. Tadeusz Szkodny

Konsultant: mgr inż. Adrian Łęgowski

Gliwice, Luty 2015

## Spis treści

1	Wstęp .....	3
2	Cel i zakres prac .....	4
2.1	Cele projektu .....	5
2.2	Zastosowane elementy .....	5
3	Model matematyczny zastosowany w programie.....	6
3.1	Kalibracja .....	6
3.1.1	Położenie i orientacja kamery .....	6
3.1.2	Błędy zniekształceń optycznych .....	9
3.1.3	Błędy obliczeń współrzędnych .....	11
3.2	Położenie obiektu manipulacji .....	13
4	Opis Oprogramowania .....	15
4.1	Program pozyskujący dane.....	15
4.1.1	Graficzny interfejs użytkownika .....	16
4.1.2	Sposób działania .....	16
4.2	Program obliczający współrzędne położenia i orientacji .....	17
4.2.1	Graficzny interfejs użytkownika .....	17
5	Testy Aplikacji.....	18
5.1	Test programu pozyskującego dane.....	19
5.2	Test programu obliczającego położenie obiektu.....	24
5.2.1	Kalibracja .....	24
5.2.2	Położenie obiektu.....	25
6	Podsumowanie .....	26
7	Bibliografia.....	27

# 1 Wstęp

Współcześnie jednym z najważniejszych składników stanowisk zrobotyzowanych są systemy wizyjne za pomocą których przekazywane są odpowiednie współrzędne określające położenie oraz orientacje obiektów manipulacji. Parametry te są obliczane przez odpowiednie oprogramowanie. Przy projektowaniu tego oprogramowania uwzględniono błędy współrzędnych odczytywanych z matrycy kamery. Błędy odczytu z matrycy powodują niedokładności obliczeń współrzędnych punktów w układzie odniesienia, skojarzonego ze stanowiskiem technologicznym. Do określenia dokładności obliczania współrzędnych obserwowanych punktów była niezbędna analiza błędów. Błędy te spowodowane są: błędami odczytu współrzędnych z matrycy kamery, zniekształceniami optycznymi kamery, błędami parametrów opisujących układ optyczny kamery i błędami obliczeń. Przy projektowaniu układu wizyjnego dążono do minimalizacji wymienionych błędów.

Podczas pracy wykorzystany został szablon do kalibracji przedstawiony na rys.1 z czerwonymi markerami w narożnikach czarnych oraz białych kwadratów. Na rysunku zaznaczono osie, gdzie  $x, y$  oznaczają osie układu odniesienia, a  $x_c, y_c$  oznaczają osie układu współrzędnych kamery. Zaznaczono także orientacyjne miejsca umiejscowienia 2 kamer.

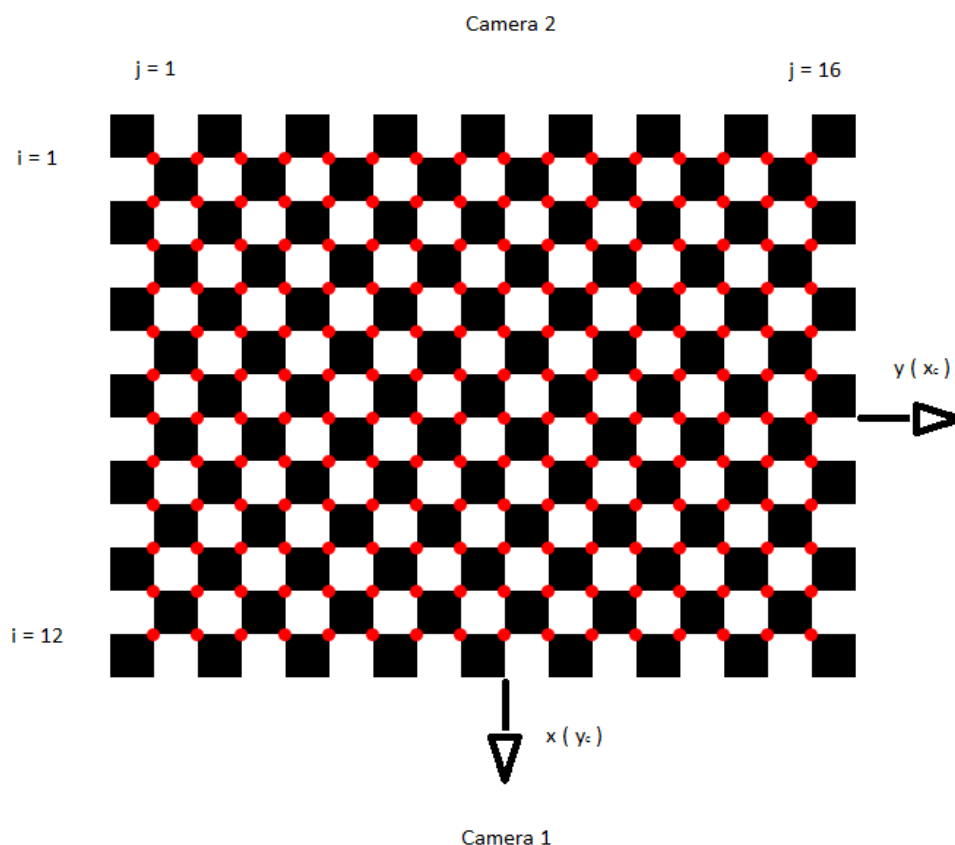


Fig. 1. Szablon do kalibracji kamery

Do odczytania współrzędnych punktów można użyć dowolnego programu graficznego, jednak taki sposób obciążony ryzykiem popełniania błędów. Żeby zmniejszyć szanse pojawienia się takich błędów oraz zautomatyzować pobieranie tych współrzędnych wykorzystano program z pracy inżynierskiej Artura Mellera[1]. Współrzędne te następnie skorygowano, aby zmniejszyć błędy spowodowanych wspomnianymi wyżej zniekształceniami za pomocą modelu matematycznego przedstawionego w pracach [2,3].

Do obliczenia współrzędnych  $x_c$   $y_c$   $z_c$  układu współrzędnych kamery oraz  $x$   $y$   $z$  układu odniesienia zastosowano algorytm *Camera* [4]. Długość ogniskowej i wymiar piksela w celu zminimalizowania błędów zostały przyjęte z dokumentacji technicznej kamery.

## **2 Cel i zakres prac**

W tym rozdziale przedstawiono podstawowe założenia, które powinny być spełnione w ramach tego projektu.

### **2.1 Cele projektu**

Celem tego projektu było opracowanie oprogramowania obliczającego pozycję obiektu manipulacji na podstawie współrzędnych punktów charakterystycznych otrzymanych z programu zewnętrznego. Opracowany został algorytm kalibrujący kamerę, aby następnie używając skalibrowanych danych uzyskać pozycje obiektu w postaci macierzy, którą można łatwo przekształcić do postaci wymaganej przez robota. Podczas prac zrealizowano następujące zadania:

- Opracowanie i wykonanie obiektu oraz szablonu do kalibracji kamer,
- Opracowanie algorytmów kalibracji geometrycznej kamer oraz ich implementacja w komputerze zewnętrznym,
- Kalibracja geometryczna kamer w środowisku programowych komputera zewnętrznego,
- Opracowanie algorytmów stereowizyjnego uzyskania pozycji obserwowanego obiektu i ich implementacja w komputerze zewnętrznym,
- Zaprojektowanie stanowiska i testy aplikacji.

### **2.2 Zastosowane elementy**

Podczas pracy nad projektem wykorzystano:

- Kamery sieciową EDIMAX IC-7100P (Rys.2),



*Fig. 2. Kamera EDIMAX IC-7100P*

- Komputer z systemem operacyjnym Windows oraz oprogramowaniem Microsoft Visual Studio,
- Router.

### **3 Model matematyczny zastosowany w programie**

W tym rozdziale zaprezentowano wzory oraz obliczenia wykorzystane w oprogramowaniu tworzonym w niniejszym projekcie.

#### **3.1 Kalibracja**

W punkcie tym zostały opisane wszystkie działania wykonywane przed przystąpieniem do ustalania pozycji obiektu, takie jak kalibracja oraz korekta współrzędnych. W części przedstawiono przykładowe wyniki kalibracji.

##### **3.1.1 Położenie i orientacja kamery**

W punkcie tym zaprezentowano obliczenia współrzędnych położenia i orientacji kamer ustawionych pod dwoma różnymi kątami do powierzchni szablonu z rys.1. Układ współrzędnych kamery  $x_c$   $y_c$   $z_c$  został skojarzony ze środkiem matrycy. Ustawienia kamer pokazane zostały na rys. 3.

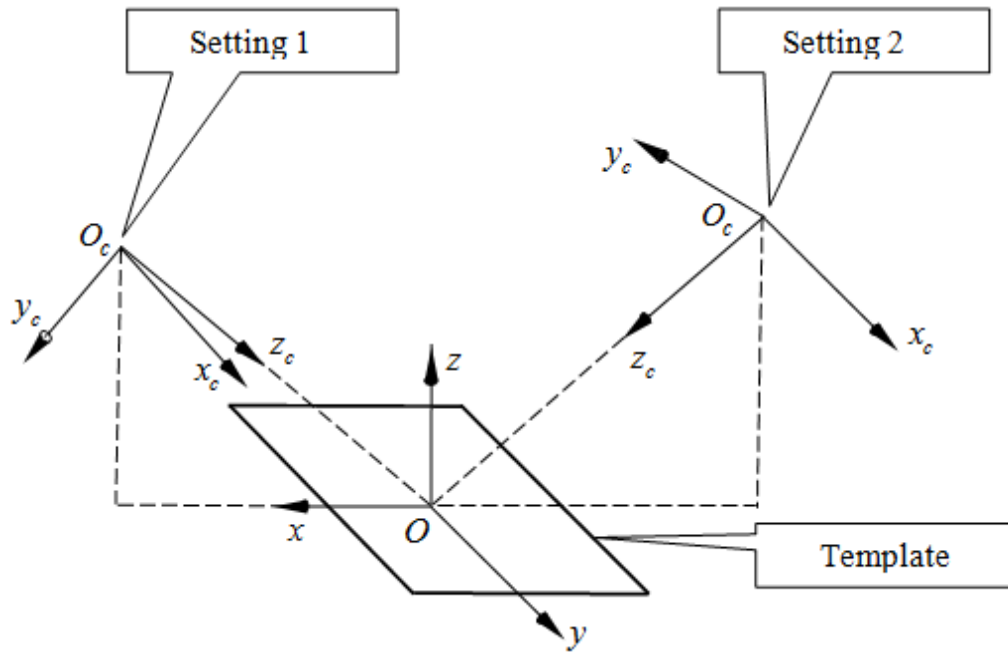


Fig.3. Ustawienia kamer

Obraz szablonu otrzymany z kamer przy podanych ustawieniach 1 oraz 2 prezentują odpowiednio rys.4a. oraz rys.4b.

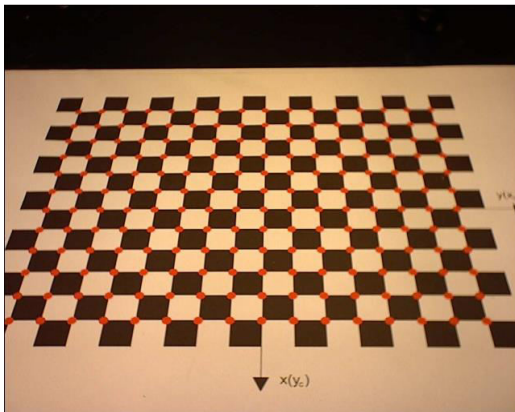


Fig.4a. Obraz z ustawienia 1

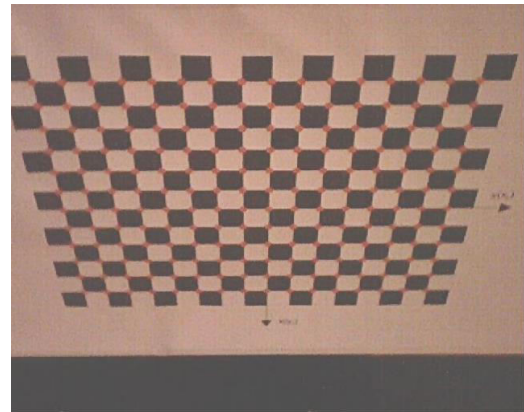


Fig.4b. Obraz z ustawienia 2

Współrzędne wszystkich punktów charakterystycznych z szablonu uzyskujemy w pikselach wykorzystując program omówiony szerzej w części 5 niniejszej pracy. Z danych technicznych zastosowanych kamer Edimax IC-7100P wynika długość ogniskowej  $f_c = 5.01 \text{ mm}$  i wymiary piksela równe  $2.8 \cdot 10^{-3} \text{ mm} \times 2.8 \cdot 10^{-3} \text{ mm}$ . Gdy przemnożymy współrzędne w pikselach przez odpowiedni wymiar piksela uzyskujemy współrzędne odczytane z obrazu w  $\text{mm}$  w układzie kamery  $x_c y_c$ . Współrzędne tych punktów w układzie odniesienia  $xy$  łatwo wyznaczyć, ponieważ znajdują się w narożnikach kwadratów o boku znanej długości.

Do opisu położenia i orientacji układu kamery  $x_c y_c z_c$  względem układu odniesienia  $xyz$  wykorzystana została macierz jednorodna  $\mathbf{T}_c$  przekształceń opisana w równaniu (1)

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

Zapis ten reprezentuje kolejne przekształcenia względem układu odniesienia  $xyz$ . Wymienione wyżej przekształcenia to: obrót wokół osi  $x$  o kąt  $\alpha$ , obrót wokół osi  $y$  o kąt  $\beta$ , obrót wokół osi  $z$  o kąt  $\gamma$ , przemieszczenie  $d_z$  wzdłuż osi  $z$ , przemieszczenie  $d_y$  wzdłuż osi  $y$  oraz przemieszczenie  $d_x$  wzdłuż osi  $x$  [5,6,7].

Do wyznaczenia współrzędnych wykorzystano algorytm *Camera* [4]. Parametry wejściowe tego algorytmu to wspomniane wyżej współrzędne  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $d_x$ ,  $d_y$ ,  $d_z$ ; współrzędne punktów A, B, C z szablonu:  ${}^c z_A$ ,  ${}^c z_B$ ,  ${}^c z_C$  w układzie  $x_c y_c z_c$ , a także  ${}^c x_{Ac}$ ,  ${}^c y_{Ac}$ ,  ${}^c x_{Bc}$ ,  ${}^c y_{Bc}$ ,  ${}^c x_{Cc}$ ,  ${}^c y_{Cc}$  w układzie  $x_c y_c z_c$  (odczytane z kamery) oraz  $x_A$ ,  $y_A$ ,  $z_A$ ,  $x_B$ ,  $y_B$ ,  $z_B$ ,  $x_C$ ,  $y_C$ ,  $z_C$  w układzie  $xyz$ . Oprócz tych współrzędnych potrzebna jest także długość ogniskowej  $f_c$  oraz dokładność obliczeń  $\delta$ .

Obliczenia współrzędnych kamery dla ustawień zaprezentowanych w rys.3 wykonała metoda kalibracji. Metoda ta utworzyła 5056 zbiorów punktów A, B, C. Zbiory te zostały utworzone przez wspomnianą metodę z 24 punktów szablonu. Obliczone współrzędne kamer dla tych zbiorów zostały uśrednione. Wybrano punkty poza początkiem układu współrzędnych  $O_c$ , ale w jego najbliższym otoczeniu. Współrzędne odczytane tych punktów są obciążone błędami, spowodowanymi zniekształceniami optycznymi. Punkty te leżą na kwadratach o bokach dwu- oraz czterokrotnie większym niż długość boku podstawowego kwadratu w szablonie rys.1 ze środkami w przybliżeniu pokrywającymi się z punktem  $O_c$ . Współrzędne  $z_A = z_B = z_C = 0 \text{ mm}$ . Rys.5 przedstawia te punkty.

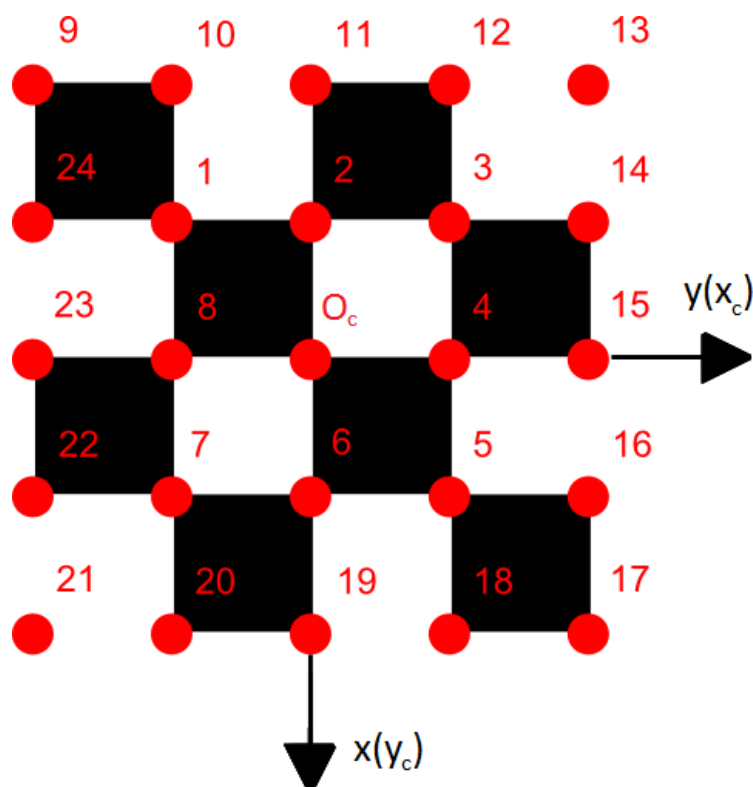


Fig.5. 24 punkty wykorzystane do wygenerowania 5056 zbiorów

Dla wspomnianych powyżej początkowych parametrów wejściowych obliczono współrzędne kamery  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $d_x$ ,  $d_y$ ,  $d_z$  z dokładnością  $\delta=10^{-6} \text{ mm}$  [4].

Dla kamer z ustawieniami 1 oraz 2 dokonano obliczenia współrzędnych orientacji i położenia. Współrzędne te z uwagi na wysoką dokładność typu double w języku C# w celu prezentacji zostały zaokrąglone do 4 miejsca po przecinku, współrzędne uśrednione dla odpowiednio ustawienia 1 i 2 reprezentują równania 2a oraz 3a, macierz  $\mathbf{T}_c$  została zaprezentowana w równaniach 2b oraz 3b.

Kamera1:

$$\alpha = 225.8552^\circ, \beta = 1.7101^\circ, \gamma = 91.3053^\circ, d_x = 412.3035 \text{ mm}, d_y = 19.5456 \text{ mm},$$

$$d_z = 399.7759 \text{ mm}. \quad (2a)$$

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

Kamera2:

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

$$d_y = -22.5570 \text{ mm}, d_z = 433.8042 \text{ mm}. \quad (3a)$$

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

### 3.1.2 Błędy zniekształceń optycznych

W celu uproszczenia opisu punktów charakterystycznych szablonu zastosujemy pojęcie wierszy oraz kolumn. Wiersze to punkty leżące na liniach równoległych do osi  $y$  przedstawionej na rys.1. Z tego rysunku wynika że w szablonie znajduje się 12 wierszy po 16 punktów każdy. Kolumny to punkty leżące na liniach równoległych do osi  $x$ . Analogicznie w szablonie występuje 16 kolumn po 12 punktów każda. Punkt  $P_{ij}$  należy to  $i$ -tego wiersza i  $j$ -tej kolumny. Z powodu zniekształceń optycznych współrzędne odczytane z matrycy kamery  ${}^c x_c(i, j)$  i  ${}^c y_c(i, j)$  obrazu punktów  $P_{ij}$  w układzie współrzędnych  $x_c, y_c$  obciążone są błędami  $\Delta^c x_c(i, j)$  i  $\Delta^c y_c(i, j)$ , które wynikają z opisu matematycznego zniekształceń za pomocą współczynników  $k_1, k_2, k_3, p_1$  i  $p_2$  [2,3]. Te błędy i parametry opisane są za pomocą równań (4a) i (4b). Ze współrzędnych odczytanych z kamery, za pomocą równania (4c) można wyznaczyć błędy  $\Delta^c x_c(i, j)$  i  $\Delta^c y_c(i, j)$ .



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

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

$$\Delta^c x_c(i, j) = {}^c x_c(i, j) - {}^c x_{ci}(i, j), \quad \Delta^c y_c(i, j) = {}^c y_c(i, j) - {}^c y_{ci}(i, j),$$

$${}^c r_{ci}(i, j)^2 = {}^c x_{ci}(i, j)^2 + {}^c y_{ci}(i, j)^2. \quad (4c)$$

W równaniach (4a-c) występują współrzędne idealne  ${}^c x_{ci}(i, j)$  i  ${}^c y_{ci}(i, j)$  z brakiem wpływu zniekształceń optycznych. Współrzędne te można uzyskać wykorzystując postać jednorodną  $\mathbf{r}(i, j)$  wektora, który opisuje punkt  $P_{ij}$  w układzie odniesienia.

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

Współrzędne występujące w równaniu (5), możemy przedstawić używając indeksów wierszy i kolumna w następujący sposób:  $x(i, j) = (i - 7) \cdot \text{długość boku kwadratu w mm na szablonie}$  oraz  $y(i, j) = (j - 9) \cdot \text{długość boku kwadratu w mm na szablonie}$ . Początek układu odniesienia  $O$  stanowi punkt  $P_{79}$ . Jako, że wszystkie punkty leżą na płaszczyźnie szablonu  $xy$  przyjmujemy, że  $z(i, j) = 0$ . Posiadając macierz  $\mathbf{T}_c$  jesteśmy w stanie obliczyć postać jednorodną wektora  ${}^c \mathbf{r}(i, j)$  wektora opisującego punkt  $P_{ij}$  w układzie kamery  $x_c y_c z_c$ .

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

Stosując współrzędną  ${}^c x(i, j)$  z równania (6) możemy obliczyć wartość współrzędnej idealnej  ${}^c x_{ci}(i, j)$ . Wykorzystując zależności geometryczne przedstawione na rys.6 możemy wyznaczyć zależność (7a) opisującą wymienioną wyżej współrzędną idealną.

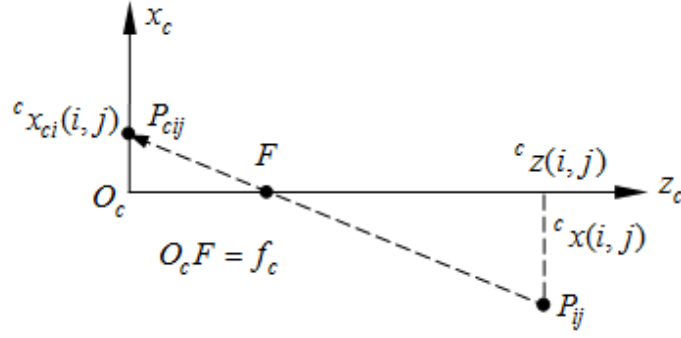


Fig.6. Zależności geometryczne punktu  $P_{ij}$  i jego obrazu  $P_{cij}$

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

Analogicznie możemy wyznaczyć wzór (7b) opisujący drugą współrzędną idealną  ${}^c y_{ci}(i, j)$ .

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

Korzystając z powyższych równań (6), (7a), (7b) mamy możliwość wyliczenia  $\Delta^c x_c(i, j)$ ,  $\Delta^c y_c(i, j)$  i  ${}^c r_c(i, j)$  potrzebne w równaniach (4c). Dzięki temu jesteśmy w stanie zastosować równania (4a) i (4b) do obliczenia współczynników  $k_1$ ,  $k_2$ ,  $k_3$ ,  $p_1$  i  $p_2$ . Za pomocą sumy zaprezentowanej w równaniu (8), jesteśmy w stanie wyliczyć te współczynniki używając metody minimalno-kwadratowej. Wyniki tej metody dla przykładowych danych przedstawiają równania (9a-10d) dla odpowiednich ustawień kamery przedstawionych na rys.3.

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

Ustawienie kamery 1:

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

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

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

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

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

Ustawienie kamery 2:

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

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

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

$$k_1 = 0.0072 \text{ mm}^{-2}, k_2 = -0.0046 \text{ mm}^{-4}, k_3 = 0.0001 \text{ mm}^{-6}, p_1 = -0.0058 \text{ mm}^{-2}, \\ p_2 = -0.0002 \text{ mm}^{-2}. \quad (10d)$$

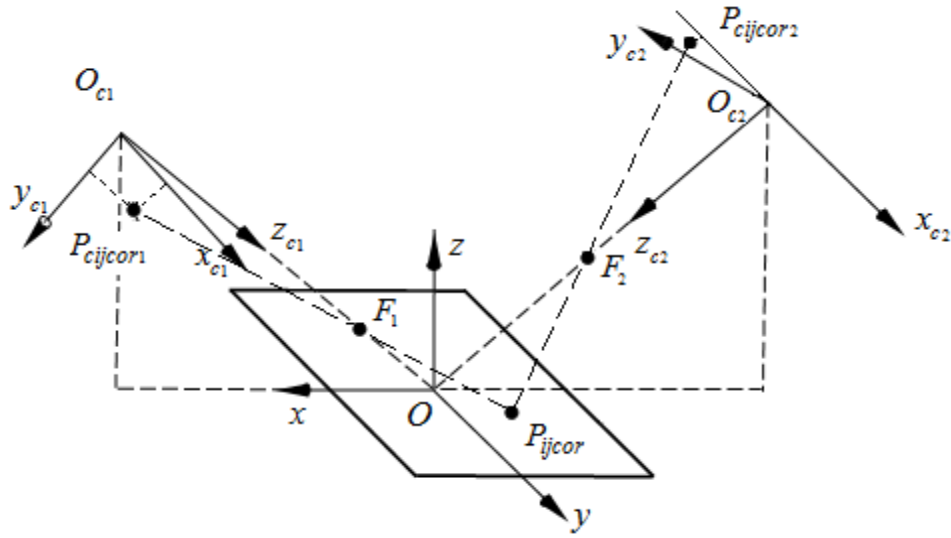
Przedstawione równania opisują błędy za pomocą różnej ilości współczynników. Wyrażenia (9a), (10a) - za pomocą jednego współczynnika; (9b), (10b) - dwóch; (9c), (10c) - trzech oraz (9d), (10d) - pięciu.

### 3.1.3 Błędy obliczeń współrzędnych

Stosując współczynniki opisane wyrażeniami (9a)-(10d) możemy obliczyć błędy  $\Delta^c x_c(i, j)$  i  $\Delta^c y_c(i, j)$  przedstawionych za pomocą równań (4a) i (4b). Po wyznaczeniu tych błędów możemy zastosować korektę współrzędnych punktów  $^c x_c(i, j)$  i  $^c y_c(i, j)$ . Korekta polega na odjęciu wspomnianych błędów  $\Delta^c x_c(i, j)$  i  $\Delta^c y_c(i, j)$  od współrzędnych odczytanych  $^c x_c(i, j)$  i  $^c y_c(i, j)$ . Nowe poprawione współrzędne oznaczmy jako  $^c x_{ccor}(i, j)$  i  $^c y_{ccor}(i, j)$ , współrzędne te są oznaczone przez równanie (11).

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

Stosując skorygowane współrzędne jesteśmy w stanie obliczyć współrzędne punktów w układzie odniesienia  $xyz$ . Przyjmijmy następujące oznaczenia tych współrzędnych:  $x_{cor}(i, j)$ ,  $y_{cor}(i, j)$  i  $z_{cor}(i, j)$ . Współrzędne te opisują punkt  $P_{ijcor}$  przedstawiony na rys.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. Współrzędne  $x$  punktów  $P_{ijcor}$  i  $P_{cijcor}$

Aby obliczyć współrzędne  $x_{cor}(i, j)$ ,  $y_{cor}(i, j)$  i  $z_{cor}(i, j)$  potrzebne są współrzędne punktu  $P_{cijcor}$ , czyli  $x_{ccor}(i, j)$ ,  $y_{ccor}(i, j)$ ,  $z_{ccor}(i, j)$ , a także  $x_F$ ,  $y_F$ ,  $z_F$  ognisk  $F$  w układzie odniesienia. Warto zaznaczyć, że odcinki  $O_{c1}F_1=f_{c1}$  oraz  $O_{c2}F_2=f_{c2}$  gdzie  $f_{c1}$  i  $f_{c2}$  to ogniskowe kamery. Równania (12) opisują te współrzędne:

$$P_{F1} = \begin{bmatrix} x_{F1} \\ y_{F1} \\ z_{F1} \\ 1 \end{bmatrix} = T_{c1} \begin{bmatrix} 0 \\ 0 \\ f_{c1} \\ 1 \end{bmatrix}, \quad P_{F2} = \begin{bmatrix} x_{F2} \\ y_{F2} \\ z_{F2} \\ 1 \end{bmatrix} = T_{c2} \begin{bmatrix} 0 \\ 0 \\ f_{c2} \\ 1 \end{bmatrix}. \quad (12)$$

Równania (13),(14) opisują odpowiednio proste przechodzące przez punkty  $P_{cijcor1}$ ,  $F_1$  i  $P_{ijcor}$  oraz  $P_{cijcor2}$ ,  $F_2$  i  $P_{ijcor}$ .

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

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

Powyższe równania pozwalają utworzyć układy trzech równań liniowych z trzema niewiadomymi  $x_{cor}(i, j)$ ,  $y_{cor}(i, j)$ ,  $z_{cor}(i, j)$ . Można utworzyć 6 takich układów. Opracowany program *odl0* rozwiązuje te równania. Dokładność wyznaczania współrzędnych punktów na płaszczyźnie szablonu możemy opisać bezwzględnymi wartościami odległości

$\Delta r(i, j) = \sqrt{\Delta x(i, j)^2 + \Delta y(i, j)^2}$ , gdzie  $\Delta x(i, j) = |x(i, j) - x_{cor}(i, j)|$  i  $\Delta y(i, j) = |y(i, j) - y_{cor}(i, j)|$ . Poniżej przedstawiono wyniki dla przykładowych danych.

Dla ustawienia 1 kamery 1 z rys.3 oraz różnych zbiorów współczynników otrzymano:

- $\max[\Delta r(i, j)] = \Delta r(2,1) = 1.2955 \text{ mm}$  dla współczynników opisanych równaniami (9a),
- $\max[\Delta r(i, j)] = \Delta r(2,1) = 1.2947 \text{ mm}$  dla współczynników opisanych równaniami (9b),
- $\max[\Delta r(i, j)] = \Delta r(1,3) = 0.7303 \text{ mm}$  dla współczynników opisanych równaniami (9c),
- $\max[\Delta r(i, j)] = \Delta r(1,3) = 0.7303 \text{ mm}$  dla współczynników opisanych równaniami (9d).

Dla ustawienia 2 kamery 2 z rys.3 oraz różnych zbiorów współczynników otrzymano:

- $\max[\Delta r(i, j)] = \Delta r(12,16) = 4.5363 \text{ mm}$  dla współczynników opisanych równaniami (10a),
- $\max[\Delta r(i, j)] = \Delta r(12,16) = 4.4833 \text{ mm}$  dla współczynników opisanych równaniami (10b),
- $\max[\Delta r(i, j)] = \Delta r(10,1) = 2.5806 \text{ mm}$  dla współczynników opisanych równaniami (10c),
- $\max[\Delta r(i, j)] = \Delta r(10,1) = 2.5806 \text{ mm}$  dla współczynników opisanych równaniami (10d).

Jak łatwo zauważyć, dokładność obliczeń współrzędnych rośnie razem z ilością użytych współczynników modelu błędów optycznych (4a) oraz (4b). Przy użyciu tylko jednego współczynnika pojawiły się największe błędy dla obu kamer. Natomiast przy użyciu wszystkich pięciu współczynników błędy były najmniejsze.

### 3.2 Położenie obiektu manipulacji

W niniejszym punkcie przedstawiono obliczenia stosowane w programie *Object Coordinates* w celu pozyskania współrzędnych położenia obiektu manipulacji. Zdjęcie obiektu ukazane jest na rys.7 natomiast jego wymiary przedstawia rys.8.

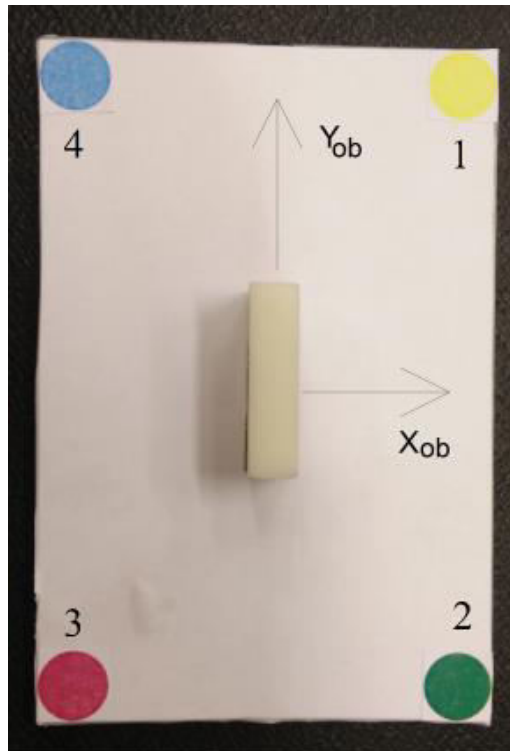


Fig.7. Obiekt manipulacji razem z osiami oraz punktami charakterystycznymi

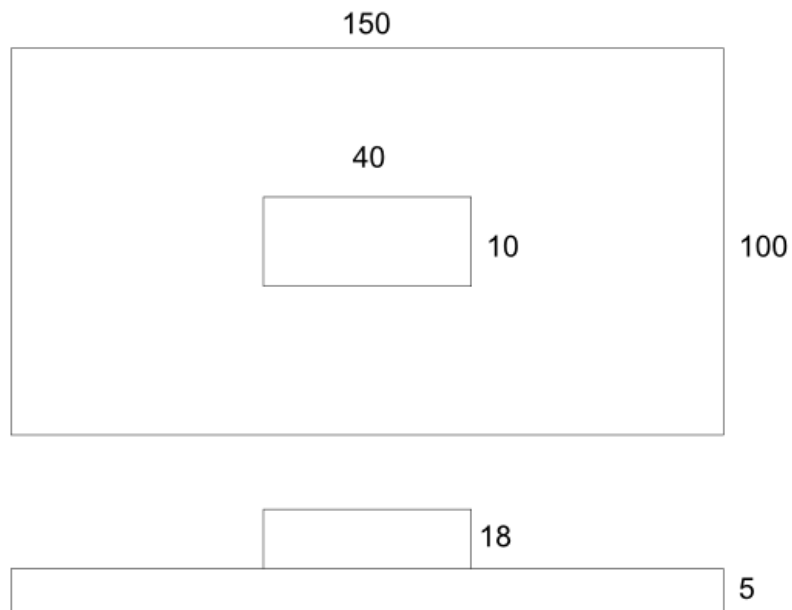


Fig.8. Wymiary obiektu manipulacji

Parametrami wejściowymi wyżej wspomnianego programu dla obliczeń są współrzędne  $x_{cor}, y_{cor}, z_{cor}$  dowolnych trzech punktów charakterystycznych obiektu, obliczonych za pomocą wspomnianego w poprzednim punkcie programu *odl0*. Niech obliczone wektory współrzędnych odpowiednich punktów opisują w układzie odniesienia *xyz* odpowiednie wektory  $\vec{r}_1, \vec{r}_2, \vec{r}_3$  i  $\vec{r}_4$ . Natomiast odległości pomiędzy odpowiednimi punktami są

reprezentowane przez wektory przedstawione w równaniach (15a-d). Następnie obliczone zostają wersory  $\vec{a}$ ,  $\vec{b}$  według równań (16) i (17).

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

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

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

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

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

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

Wersor  $\vec{a}$  jest równoległy do osi  $x_{ob}$  obiektu, natomiast wersor  $\vec{b}$  jest równoległy do osi  $y_{ob}$ . Za pomocą równania (18) obliczono wersor  $\vec{c}$ .

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

Ostatni wektor wynikowy, jest wektorem współrzędnych początku układu współrzędnych obiektu  $\vec{d}$ . Wektor ten w zależności od obliczonych punktów oraz wektorów może być wyliczony na różne sposoby. Przykładowe rozwiązania prezentuje równanie (19).

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

Ostateczny wynik programu, prezentowany jest jako macierz przekształceń  $\mathbf{T}$ .

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

## 4 Opis Oprogramowania

W niniejszym rozdziale zaprezentowano utworzone w ramach tego projektu oprogramowanie. Omówiony został tu program *Object Coordinates*, którego działanie opisane jest w poprzednim rozdziale oraz program *Camera Coordinates* za pomocą którego pozyskiwane zostały współrzędne  $x$  oraz  $y$  punktów charakterystycznych w pikselach.

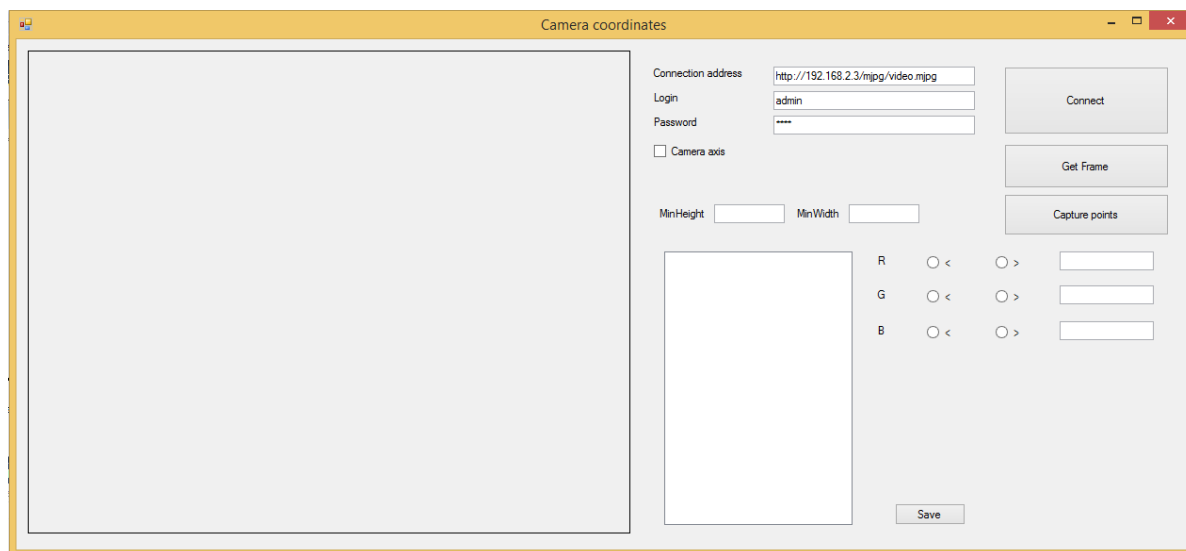
### 4.1 Program *Camera Coordinates* pozyskujący dane

W tym punkcie opisany został program zastosowany zamiast oprogramowania Artura Mellera [1]. Powody dla których z niego zrezygnowano zaprezentowane zostały w rozdziale

5. Program został napisany w środowisku Microsoft Visual Studio 2013, w języku C# .NET z wykorzystaniem biblioteki AForge.NET.

#### 4.1.1 Graficzny interfejs użytkownika

Rysunek 9 przedstawia graficzny interfejs programu.



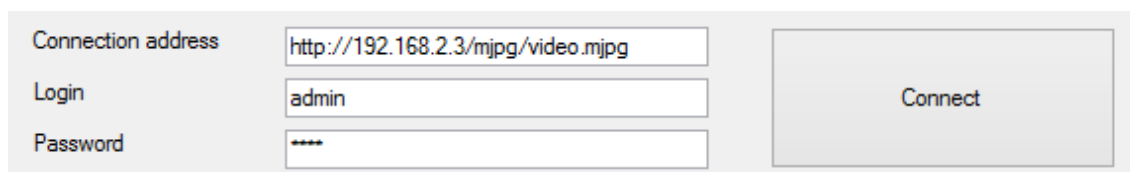
*Fig.9. GUI programu*

Interfejs programu składa się na kilka istotnych części. Po lewej pole przeznaczone na wyświetlenie obrazu z kamery/efektu binaryzacji obrazu, w prawej górnej części elementy wykorzystywane do połączenia z kamerą, prawy dolny róg to kontrolki odpowiedzialne za zmianę ustawień binaryzacji oraz wyświetlanie i zapisywanie współrzędnych znalezionych punktów.

#### 4.1.2 Sposób działania

Jako, że stworzenie tego programu nie było głównym celem tej pracy, to jest to prosty program, którego sposób działania przedstawiono pokrótce poniżej.

W celu skorzystania z programu należy najpierw połączyć się z kamerą. Jak wspomniano powyżej należy użyć elementów w prawy górnym rogu aplikacji rys.10.



*Fig.10. Połączenie z kamerą*

Po użyciu przycisku "Connect" należy ustawić środek kamery w przybliżeniu ze środkiem układu odniesienia. W celu ułatwienia tej operacji, należy użyć kontrolki "Camera axis", po



zaznaczeniu której na wyświetlanym obrazie pojawią się 2 żółte linie wskazujące jego środek. Po odpowiednim ustawieniu kamery należy użyć przycisku "Get Frame" służącego do przechwycenia klatki obrazu w celu poddania obróbce, następnie z użyciem elementów poniżej należy ustawić odpowiednie wartości RGB poszukiwanego koloru oraz "MinHeight" i "MinWidth" oznaczające odpowiednio minimalną wysokość oraz szerokość wyszukiwanych obiektów, służące do odfiltrowania elementów niepożądanych jak np. wolne piksele. Po tym kroku należy użyć przycisku "Capture points" który zbinaryzuje wg wartości RGB oraz następnie z pomocą biblioteki AForge.NET wyszuka punkty charakterystyczne. W razie nieodpowiedniej liczby wyszukanych punktów (program został przygotowany do współpracy z programem podającym współrzędne położenia obiektu) zostanie wywołany błąd z ostrzeżeniem. Przykłady użycia programu zostały zaprezentowane w rozdziale 5.

Jeśli odpowiednia liczba punktów została odnaleziona należy użyć przycisku "Save" do zapisania ich w pliku tekstowym.

## 4.2 Program *Object coordinates* obliczający współrzędne położenia i orientacji

W tym punkcie przedstawiono program którego działanie zostało opisane w rozdziale 3.

### 4.2.1 Graficzny interfejs użytkownika

Na rysunkach 11 oraz 12 przedstawiono graficzny interfejs użytkownika programu. Jak można niżej zauważyć interfejs w większości został podzielony na 2 podobne do siebie części każda odpowiedzialna za inną z dwóch kamer.

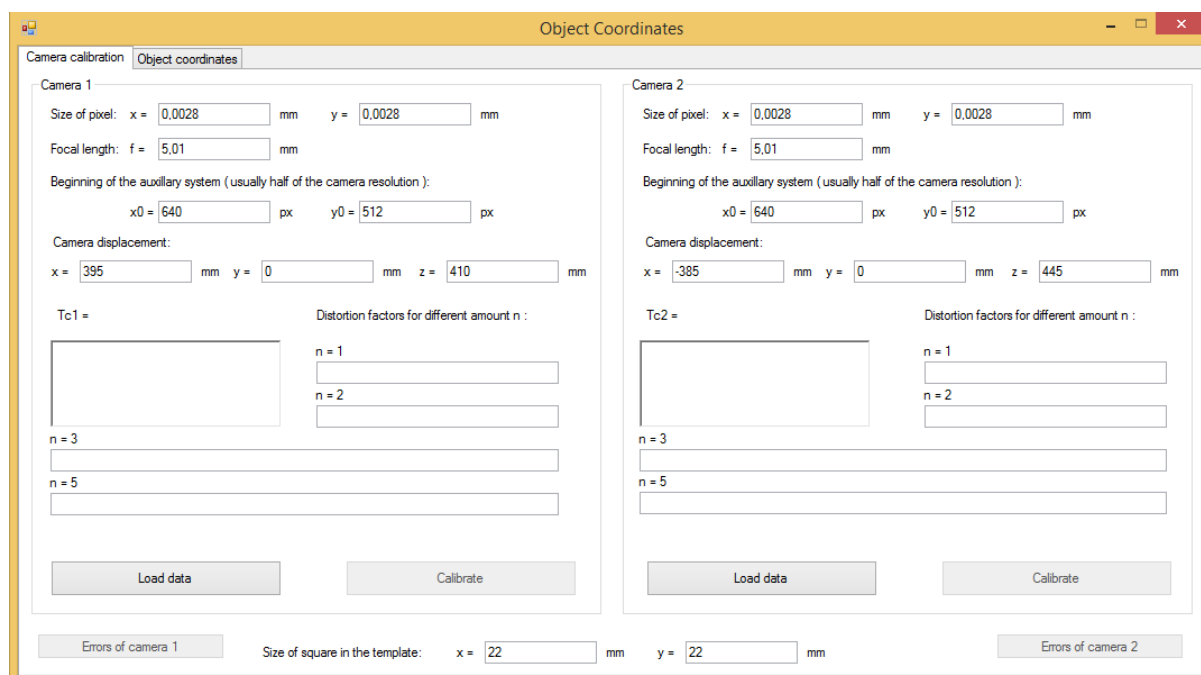


Fig. 11. GUI części kalibracyjnej

Należy zacząć od podania parametrów kamery (domyślne wartości pojawiają się dla kamer Edimax IC-7100P), jej przesunięcia (wartość  $x$  pierwszej kamery powinna mieć wartość dodatnią natomiast druga ujemną), a także danych dotyczących szablonu. Wielkość boku kwadratu szablonu to inaczej odległość między środkami punktów charakterystycznych. Następnie należy załadować odpowiednio sformatowany plik tekstowy (2 rzędy po 192 kolumny wartości liczbowych oddzielonych tabulacją, gdzie pierwszy rząd oznacza współrzędne  $x$  a drugi  $y$  punktów charakterystycznych odczytanych z kamery w pikselach) za pomocą przycisku "Load data", spowoduje to odblokowanie odpowiedniego przycisku "Calibrate" po naciśnięciu którego kamera zostanie skalibrowana, zostanie wyświetlona macierz  $T_c$  przekształceń oraz 1, 2, 3 oraz 5 współczynników zniekształceń. Po naciśnięciu przycisku "Errors of camera  $x$ " wyświetlone zostanie okno z obliczonymi wartościami  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $d_x$ ,  $d_y$ ,  $d_z$  oraz analizą błędów dla 5 współczynników. Po skalibrowaniu należy przejść do zakładki "Object coordinates".

Podobnie jak poprzednia, ta też została podzielona na 2 części za względu na kamery. W tej części należy podać odpowiednie wartości współrzędnych  $x$  oraz  $y$  punktów charakterystycznych obiektu manipulacji w pikselach. Do działania programu wystarczy podać jedynie 3 takie punkty, jednak należy je podać parami dla obu kamer w przeciwnym wypadku program wyświetli błąd.

Po wypełnieniu pól należy nacisnąć przycisk "Calculate" który poda w odpowiednim polu macierz wynikową (20).

The screenshot shows the 'Object Coordinates' window with the following data in the input fields:

Camera	Point	X	Y
Camera 1	First (yellow)	241	283
	Second (green)	602	279
	Third (pink)	592	435
	Fourth (aqua)	195	438
Camera 2	First (yellow)	147	241
	Second (green)	593	236
	Third (pink)	591	431
	Fourth (aqua)	192	433

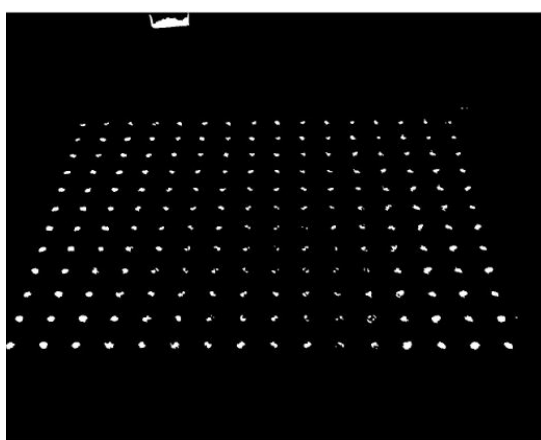
Fig.12. GUI części uzyskiwania położenia

## 5 Testy Aplikacji

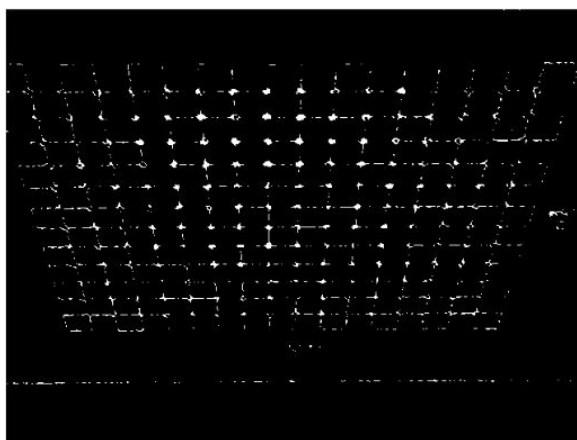
W tym rozdziale zostały opisane testy oprogramowania, które jest przedmiotem tej pracy.

### 5.1 Test programu pozyskującego dane

Pierwszym testem była próba binaryzacji obrazu punktów charakterystycznych szablonu. Na początku użyto wspomnianego wyżej programu Artura Meller [1]. Niestety zastosowana tam metoda nie pozwalała na jednoczesne pozbycie się obiektów niepożądanych przy pozostawieniu wszystkich punktów charakterystycznych. Na rys.13a oraz rys.13b zaprezentowano obrazy z dwóch różnych kamer, odpowiednio rys.4a i 4b, poddanych binaryzacji podanym programem.

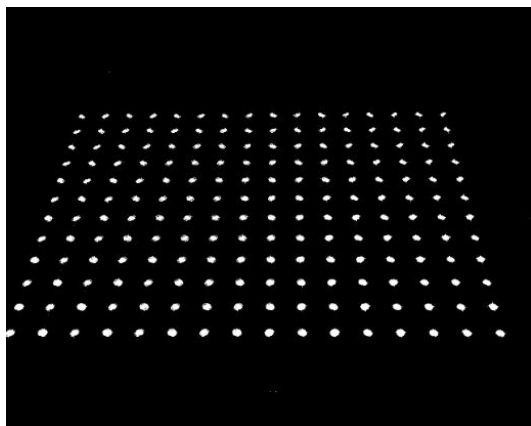


*Fig.13a. Binarzacja obrazu rys.4a*

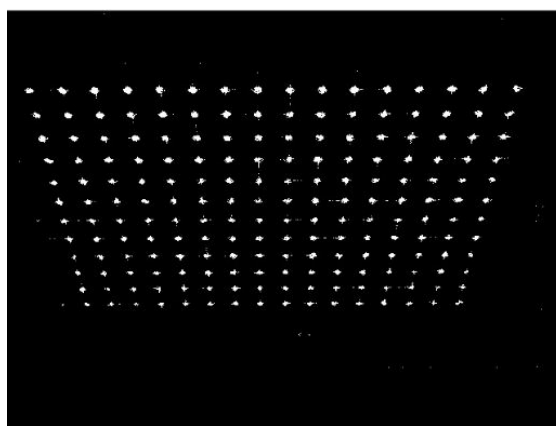


*Fig.13b. Binarzacja obrazu rys.4b*

Jak widać powyżej zbinaryzowany obraz, mimo zastosowania operacji erozji i dylatacji, zawiera dużo niepożądanych obiektów oraz jak można zauważyć na rys.13b gdzie wystąpiło inne oświetlenie część punktów charakterystycznych została pominięta. Dlatego stworzono wspomniany powyżej program, operujący wartościami RGB. Działanie programu jest wolniejsze jednak gwarantuje lepsze wyniki. Na rys.14a oraz 14b zaprezentowano zbinaryzowany obraz dla tych samych obrazów z kamery.



*Fig.14a. Binarzacja obrazu rys.4a*

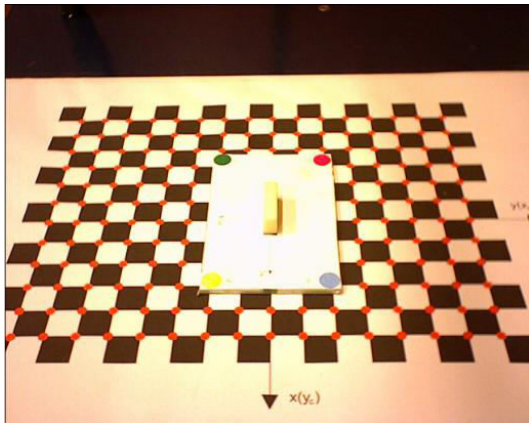


*Fig.14b. Binarzacja obrazu rys.4b*

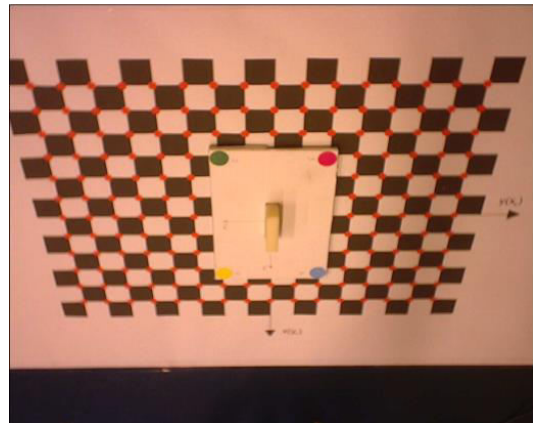
Dzięki zastosowaniu filtra, można pominąć pozycje małych niepożądanych obiektów, dzięki czemu z obu obrazów pozyskano odpowiednią liczbę współrzędnych punktów.

Następnym krokiem testów było sprawdzenie przechwytywania wartości punktów charakterystycznych obiektu. Niestety ze względu na duży wpływ oświetlenia na obserwowane punkty nie udało się zautomatyzować pobierania danych.

Poniżej zamieszczono rysunki przedstawiające obraz z kamery oraz wyniki binaryzacji dla odpowiednich punktów charakterystycznych.



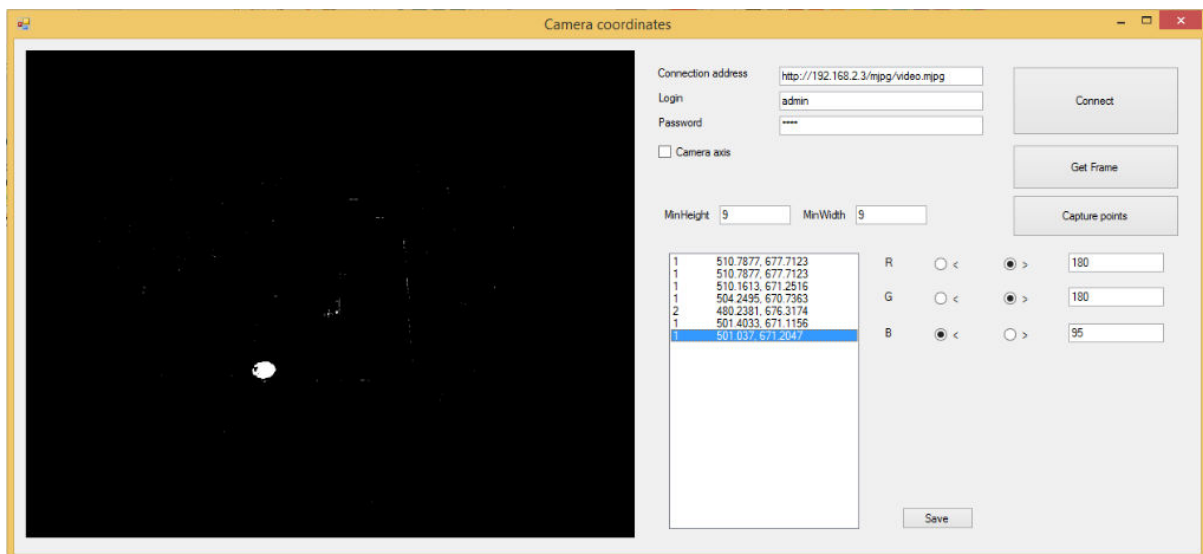
*Fig.15a. Widok z kamery 1*



*Fig.15b. Widok z kamery 2*

W następnej części zaprezentowano wyniki binaryzacji dla poszczególnych punktów charakterystycznych obiektu manipulacji.

Dla punktu pierwszego ( żółtego ):



*Fig.16a. Binaryzacja punktu 1 widoku 15a*

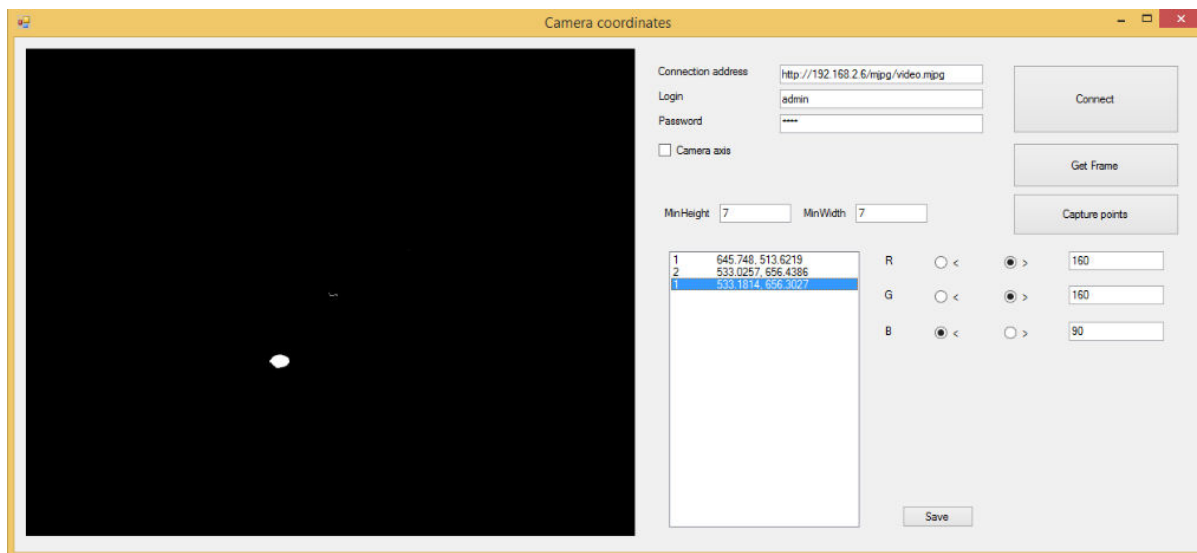


Fig.16b. Binaryzacja punktu 1 widoku 15b

Dla punktu drugiego ( zielonego ):

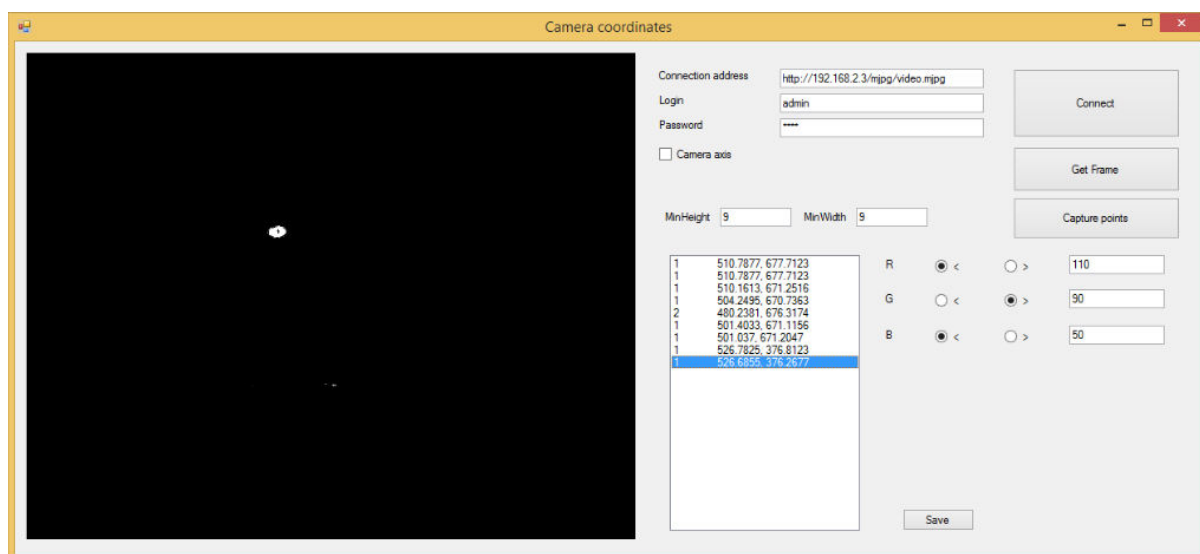
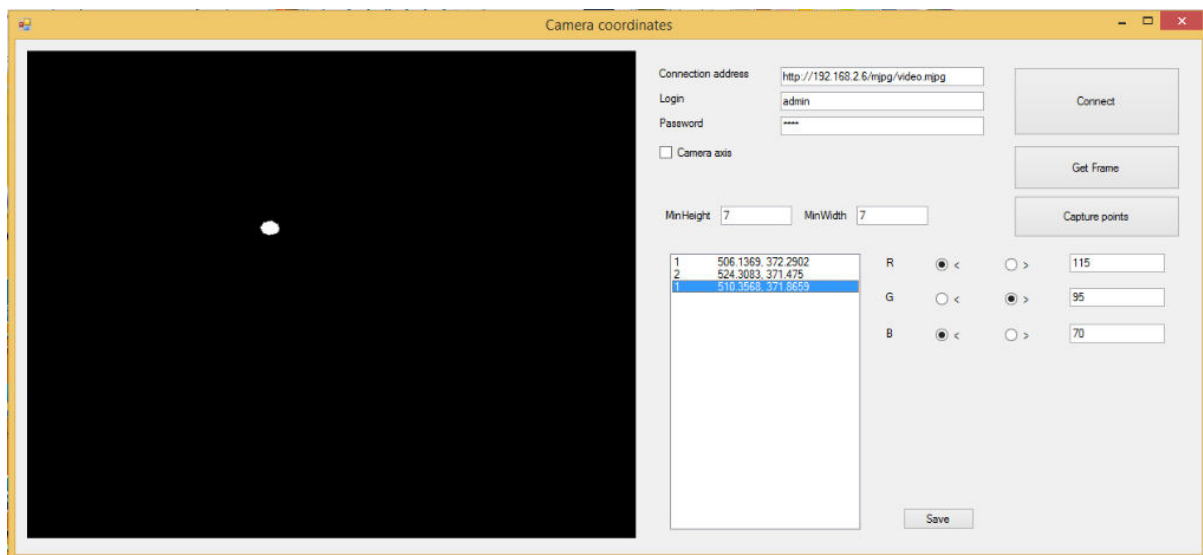
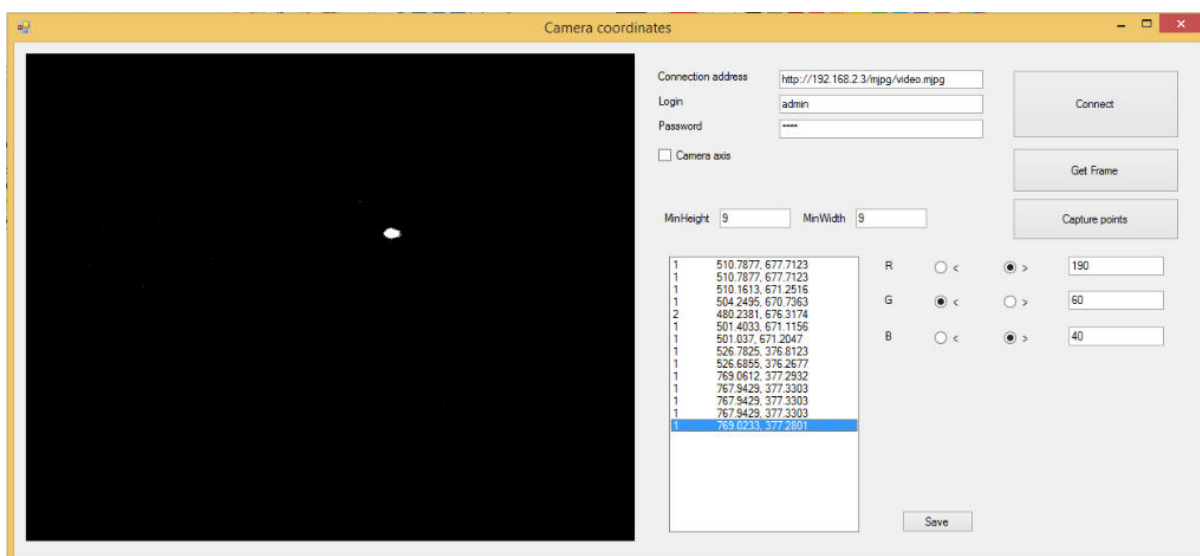


Fig.17a. Binaryzacja punktu 2 widoku 15a

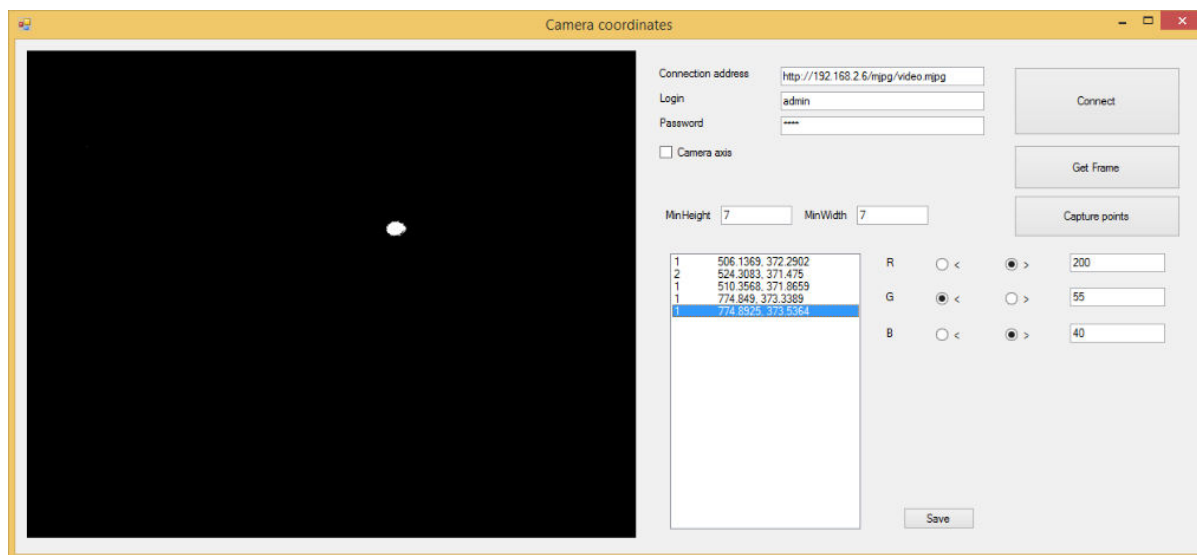


*Fig.17b. Binaryzacja punktu 2 widoku 15b*

Dla punktu trzeciego ( różowego ):

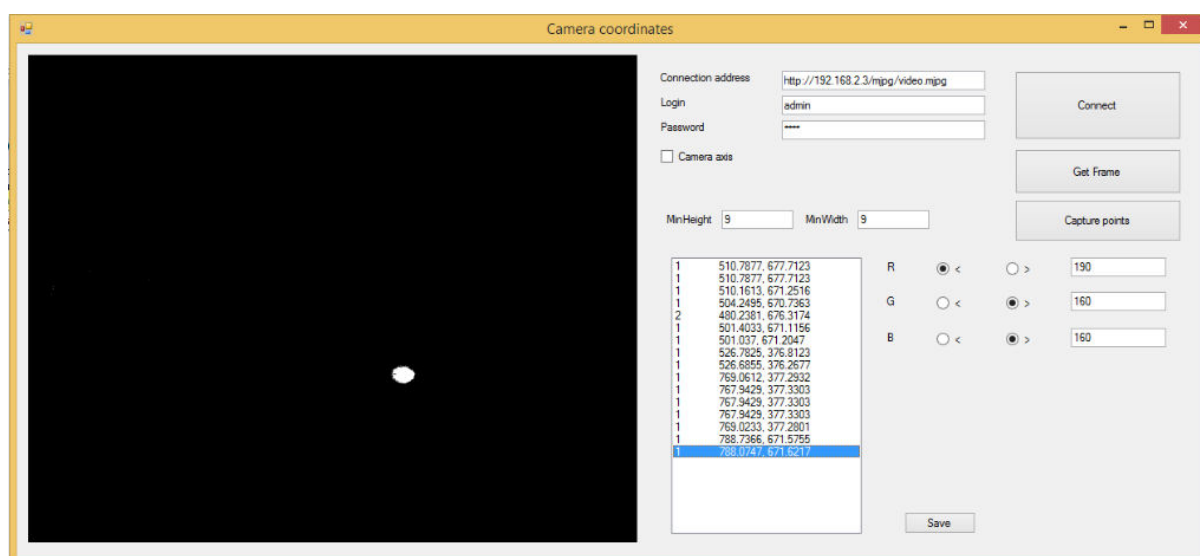


*Fig.18a. Binaryzacja punktu 3 widoku 15a*



*Fig.18b. Binaryzacja punktu 3 widoku 15b*

Dla punktu czwartego ( jasnoniebieskiego ):



*Fig.19a. Binaryzacja punktu 4 widoku 15a*

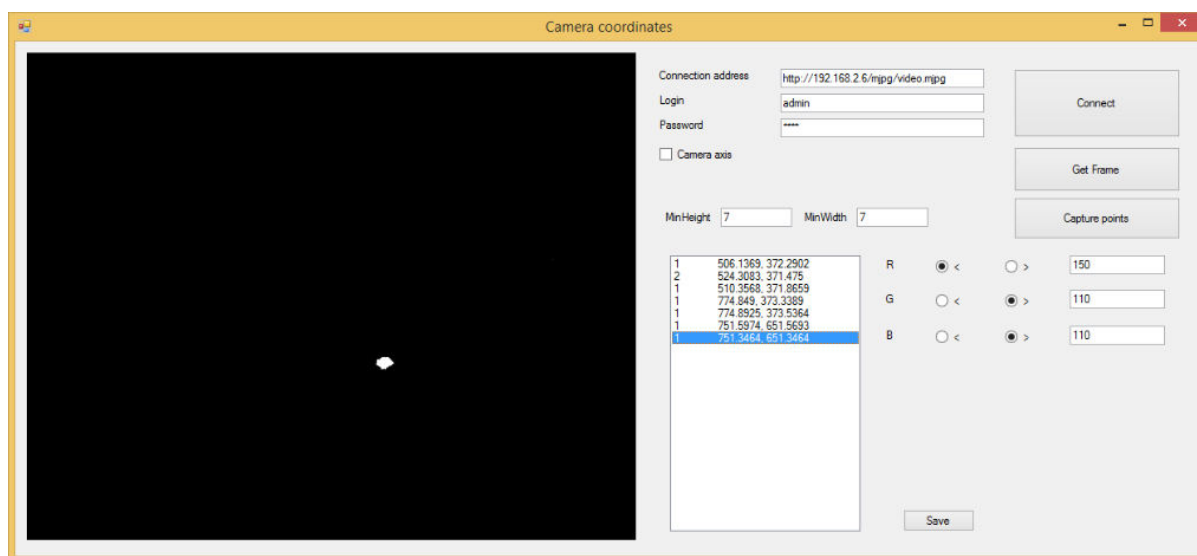


Fig.19b. Binaryzacja punktu 4 widoku 15b

Jak widać na powyższych obrazkach dzięki zastosowaniu odpowiednich wartości filtrów, można uzyskać dobre wyniki z pominięciem niepożądanych obiektów. Niestety jak widać na powyższych rysunkach efektywne wartości RGB dla tych samych barw pozyskiwanych z 2 różnych kamer są inne. Jest to prawdopodobnie spowodowane oświetleniem. Kamery są skierowane pod różnymi kątami do powierzchni szablonu, przez co odbite promienie inaczej je oświetlają.

## 5.2 Test programu obliczającego położenie obiektu

Następnym krokiem było przetestowanie głównego programu *Object coordinates* w oparciu o dane uzyskane z testów poprzedniego programu *Camera Coordinates*.

### 5.2.1 Kalibracja

W tym punkcie zaprezentowano testy części kalibracyjnej programu *Object Coordinates*. Zaprezentowano tutaj jeden test kalibracji dla ustawienia kamer z rys.15a oraz 15b (wykorzystując obraz bez obiektu manipulacji). Wyniki testu z obrazu rys. 4a oraz 4b zostały przedstawione jako przykłady w rozdziale 3. Rys. 20 przedstawia wyniki aktualnego testu.



Fig.20. Wyniki testu kalibracji

Program dokonuje kalibracji kamery i oblicza 5 współczynników zniekształceń optycznych. Na rys.21a ora 21b można zaobserwować dokładność kalibracji.

Fig.21a. Dokładność kalibracji kamery 1

Fig.21b. Dokładność kalibracji kamery 2

Jak widać powyżej największa niedokładność obliczeń położenia punktów na płaszczyźnie szablonu wynosi około 1.23 mm co jest dobrym wynikiem.

## 5.2.2 Położenie obiektu

Biorąc pod uwagę specyficzne położenie obiektu manipulacji utworzono teoretyczną macierz wynikową i porównano ją z odpowiedzią programu. Teoretyczna macierz została zaprezentowana w równaniu (21).

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

Zacznijmy od ostatniej kolumny. Punkt  $d$  przyjął takie dane, ponieważ obiekt został ułożony w przybliżeniu na środku szablonu, wartość  $d_z = 5 \text{ mm}$ , ponieważ wysokość podstawy obiektu ma taką wysokość. Wersor  $\vec{c}$  przyjął taki dane, ponieważ ma ten sam zwrot i kierunek co oś  $z$  układu odniesienia. Kierunek oraz zwrot wersora  $\vec{b}$  pokrywa się z osią  $x$  układu odniesienia, natomiast wersor  $\vec{a}$  jest równoległy do osi  $y$ , jednak ma wartość ujemną, ponieważ posiada przeciwny zwrot. Na rys. 22 zaprezentowano wyniki programu.

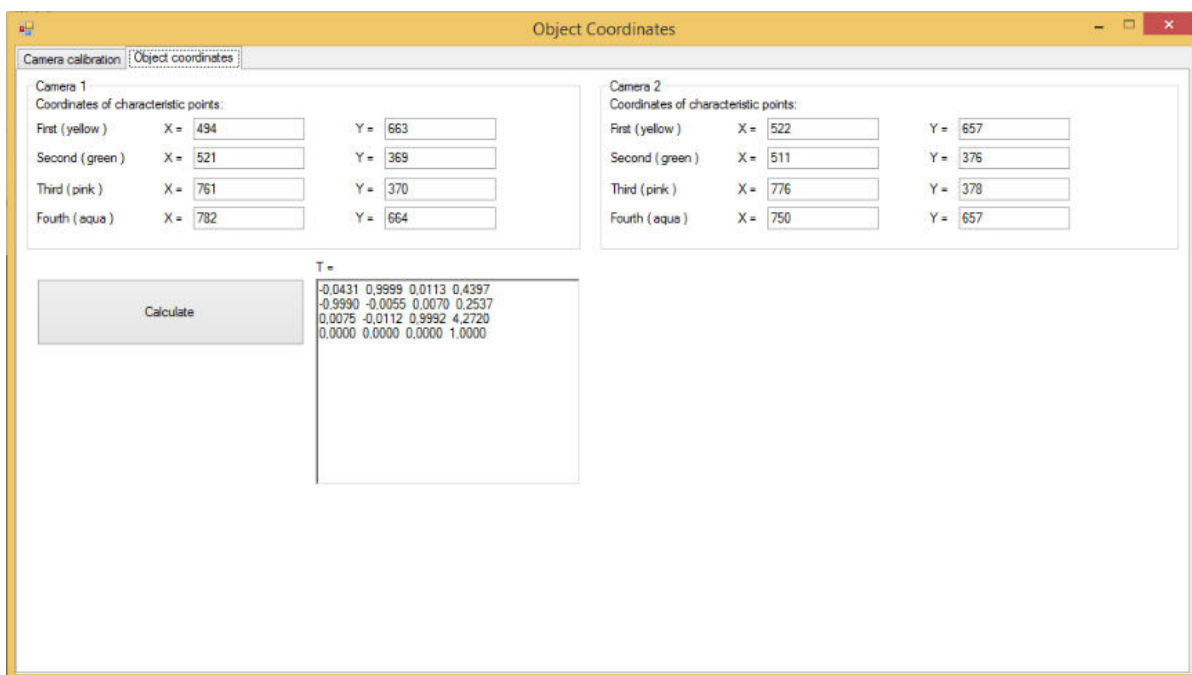


Fig.22. Wyniki testu programu

Jak widać wynik programu w przybliżeniu odpowiada teoretycznej odpowiedzi.

## 6 Podsumowanie

Najistotniejszymi informacjami wizyjnymi dla potrzeb sterowania robotami przemysłowymi są współrzędne położenia i orientacji (pozycji) obserwowanych obiektów manipulacji. Współcześnie systemy wizyjne określające pozycje oraz rozpoznające obiekt cieszą się dużym uznaniem w środowisku przemysłowym. Niestety w znacznej mierze takie urządzenia oraz oprogramowanie jest najczęściej wysoce wyspecjalizowane, a przez to drogie. Zaprezentowane rozwiązanie określania położenia obiektu można zastosować do każdej kamery, której znana jest długość ogniskowej oraz wymiar piksela, osiągając przy tym stosunkowo dużą dokładność.

Cel projektu został zrealizowany z wykorzystaniem języka C# w formie WindowsForms i biblioteki AForge.NET zastosowanej do komunikacji z kamerą oraz do wykrywania obiektów charakterystycznych.

## 7 Bibliografia

- [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](http://etacar.put.poznan.pl/marcin) Kielczewski (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.