

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA AUTOMATYKI



PRACA MAGISTERSKA

PIOTR KAPCIA

**STEROWANIE MANIPULATOREM ROBOTA Z UŻYCIEM
SYSTEMU WIZYJNEGO**

PROMOTOR:

dr inż. Paweł Rotter

Kraków 2012

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

.....

PODPIS

AGH
University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical
Engineering

DEPARTMENT OF AUTOMATICS



MASTER OF SCIENCE THESIS

PIOTR KAPCIA

VISION SYSTEM FOR MANIPULATOR CONTROL

SUPERVISOR:
Paweł Rotter Ph.D

Krakow 2012

Serdecznie dziękuję promotorowi za
życzliwą atmosferę sprzyjającą pracy
naukowej.

Spis treści

1. Wstęp	6
1.1. Warstwa fizyczna	8
1.2. Warstwa oprogramowania.....	11
1.2.1. Przetwarzanie współbieżne	12
1.2.2. Optymalizacja kodu	12
2. Manipulator robota	13
2.1. Rzeczywisty manipulator robota	13
2.2. Wirtualny manipulator robota.....	15
2.2.1. Kinematyka prosta	16
2.2.2. Kinematyka odwrotna dla kątów RPY.....	18
2.2.3. Kinematyka odwrotna	19
3. System wizyjny	21
3.1. Stany systemu wizyjnego.....	21
3.2. Procedura uruchamiania system sterującego	22
3.3. Sterowanie za pomocą gestów	23
3.3.1. Lokalizacja	23
3.3.2. Sterowanie systemem oraz efektozem.....	26
3.4. Sterowanie za pomocą wskaźnika	32
3.4.1. Czteropunktowa metoda identyfikacji transformacji stereowizyjnej	32
3.4.2. Kalibracja	35
3.4.3. Sterowanie manipulatorem robota	39
4. Opis programu	54
4.1. Graficzny interfejs użytkownika (GUI)	54
4.2. Skróty klawiaturowe	59
5. Podsumowanie	61

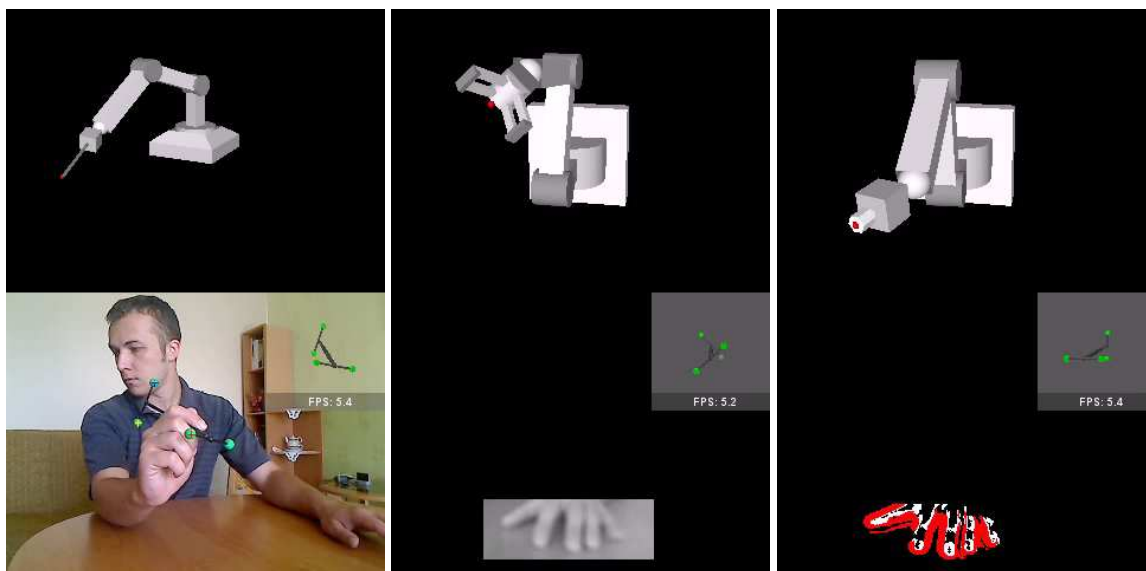
1. Wstęp

Celem niniejszej pracy jest sterowanie manipulatorem robota z wykorzystaniem systemu wizyjnego. Wykonany system wizyjny składa się z dwóch podsystemów realizujących następujące zadania:

1. Sterowanie systemem oraz efektoem za pomocą gestów
2. Sterowanie manipulatorem za pomocą wskaźnika

Dodatkowo, na potrzeby testowania systemu, został utworzony moduł pozwalający na tworzenie wirtualnych manipulatorów. Wykorzystywane będą dwa modele manipulatorów: manipulator robota terenowego oraz manipulator robota podwodnego. Potencjalnym zastosowaniem systemu jest sterowanie tego typu manipulatorami robotów mobilnych.

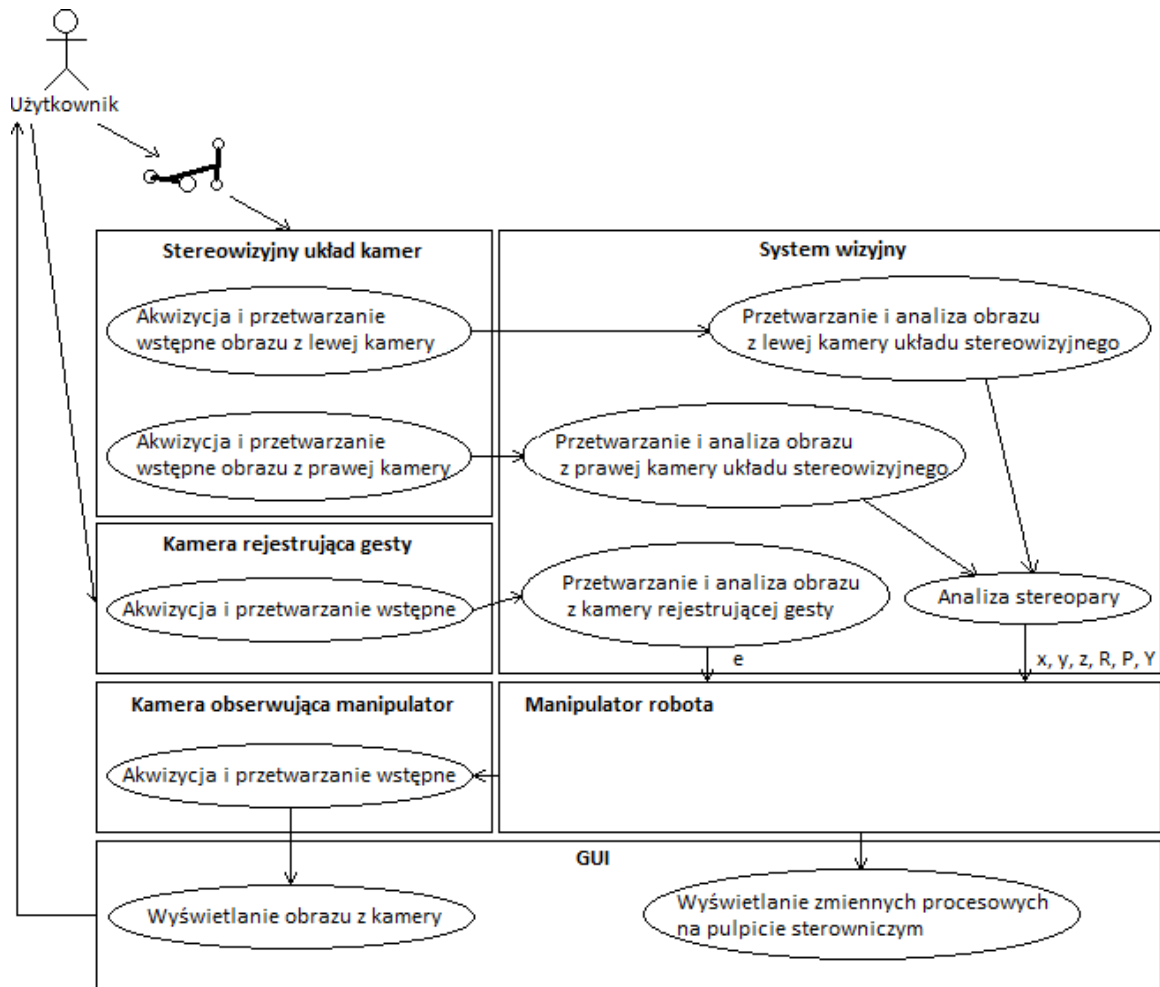
Poniżej przedstawiono przykładowe ramki obrazujące działanie systemu.



Rysunek 1.1: Przykładowe ramki

Jak już wspomniano, pozycja i orientacja zadawane są przy użyciu wskaźnika, zaś sterowanie efektoem oraz systemem odbywa się za pomocą gestów. Czerwony punkt, za którym nadaża manipulator, odpowiada zadanej pozycji. Obserwując ustawienie wirtualnego wskaźnika, można odczytać aktualnie zadawaną orientację oraz to jak podsystem stereowizyjny "widzi" wskaźnik trzymany przez użytkownika. Obiekt wskaźnika, który nie jest widoczny dla kamer, jest oznaczany jako półprzezroczysty. Pozy-

cja takiego obiektu jest odtwarzana na podstawie pozycji trzech pozostałych obiektów, z wykorzystaniem specjalnych algorytmów, które zostaną omówione w następnych rozdziałach.



Rysunek 1.2: Diagram przypadków użycia

Na powyższym rysunku został przedstawiony diagram przypadków użycia [17][7] opisujący system sterowania. Akwizycja i przetwarzanie wstępne są realizowane przez zestaw kamer oraz ich sterowniki. W systemie wizyjnym równolegle odbywają się obliczenia dla dwóch wyżej wspomnianych podsystemów (pkt 1 oraz 2). Zwracane przez nie wartości są wysyłane do manipulatora:

- e - zmienna sterująca efektozem
- x, y, z - pozycja
- RPY - orientacja:
 - R - obrót (*ang. Roll*)
 - P - nachylenie (*ang. Pitch*)
 - Y - odchylenie (*ang. Yaw*)

Na końcu, informacja o pozycji, orientacji oraz działaniu efektora, trafia do użytkownika w postaci obrazu z kamery, która obserwuje manipulator (lub wyrenderowanego obrazu, jeśli jest to wirtualny manipulator).

1.1. Warstwa fizyczna

W skład zestawu systemu wizyjnego wchodzi:

- wzorzec kalibracyjny
- trzy kamery umieszczone na statywie
- wskaźnik

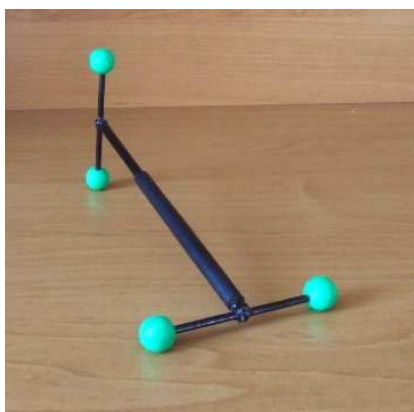
W związku z tym, że kamery układu stereowizyjnego mają zablokowaną możliwość regulacji, kalibracja jest przeprowadzana tylko raz. Ponowna kalibracja byłaby konieczna tylko w przypadku uszkodzenia lub zastosowania innego zestawu kamer oraz statywu.



Rysunek 1.3: Wzorzec kalibracyjny

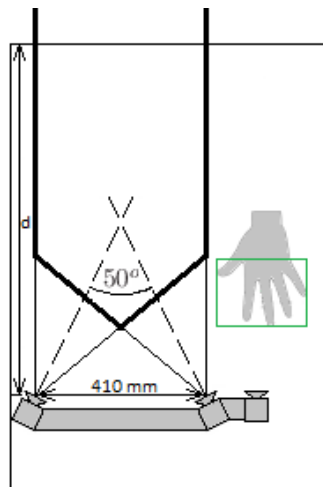


Rysunek 1.4: Układ kamer umieszczonych na statywie



Rysunek 1.5: Wskaźnik

Poniżej przedstawiono układ przestrzeni roboczych obydwu podsystemów - przestrzenie te nie kolidują ze sobą.



Rysunek 1.6: Układ przestrzeni roboczych

Ustawienie kamer układu stereowizyjnego [9] w taki sposób, że ich osie optyczne przecinają się pod kątem 50° , miało na celu optymalnie wykorzystać dostępną przestrzeń i jednocześnie zapewnić korzystne warunki dla transformacji stereowizyjnej, tak by odczytane pozycje obiektów wskaźnika jak najwierniej odzwierciedlały te rzeczywiste.

Odległość statywu z kamerami od brzegu stołu różni się w zależności od osoby użytkującej system. W tym przypadku odległość d wynosiła 80 cm .

1.2. Warstwa oprogramowania

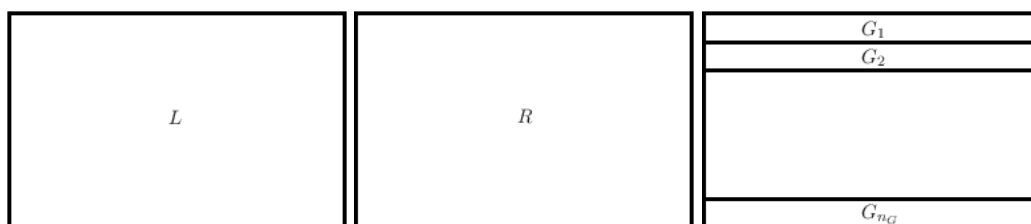
Zasadnicza część programu została napisana w języku Java [5]. Pewne funkcjonalności wymagały zastosowania C++ [6], np. w celu przechwycenia obrazu z kamer z wykorzystaniem biblioteki OpenCV. Poniżej lista zastosowanych technologii oraz narzędzi:

- Języki programowania: Java, C++
- Platforma: Java SE
- IDE: Eclipse
- JNI (Java Native Interface) - interfejs programistyczny umożliwiający uruchamianie aplikacji Java wraz z bibliotekami napisanymi w innych językach programowania (np. C++)
- OpenCV (Open Source Computer Vision Library) - biblioteka funkcji wykorzystywanych podczas obróbki obrazu
- JOGL (Java OpenGL) - biblioteka udostępniająca API OpenGL w Javie
- SWT (Standard Widget Toolkit) - biblioteka graficzna
- EJML (Efficient Java Matrix Library) - biblioteka umożliwiająca przeprowadzanie operacji na macierzach
- Xuggle - biblioteka umożliwiająca rejestrowanie obrazu
- NSIS (Nullsoft Scriptable Install System) - narzędzie umożliwiające tworzenie instalatorów programów
- MATLAB - środowisko wspomagające obliczenia naukowe i inżynierskie

1.2.1. Przetwarzanie współbieżne

Aby jak najefektywniej wykorzystać moc rdzeni procesora, zastosowano mechanizm wielowątkowości [14]. W tym przypadku, do obliczeń stosowany był dwurdzeniowy procesor, jednak nie przeszkodziło to w zasymulowaniu pracy procesora o większej ilości rdzeni. Algorytm przydziela odpowiednią ilość wątków dla trzech grup obrazów: L, R, G.

- L - obraz z lewej kamery stereo
- R - obraz z prawej kamery stereo
- G - obraz z kamery rejestrującej gesty



Rysunek 1.7: Współbieżne przetwarzanie obrazów z trzech kamer

Jeśli tylko operacja jest podzielna, kilka wątków przetwarza współbieżnie różne fragmenty tego samego obrazu. Podejście to można zastosować do następujących operacji: odejmowanie obrazów, filtracja dolnoprzepustowa oraz detekcja krawędzi. Wszystkie znajdują się w podsystemie przetwarzającym obraz z kamery rejestrującej gesty (pkt 1). Przykładowy podział wątków (dla $n = 8$):

- L - 1 wątek
- R - 1 wątek
- G - 6 wątków

Przetwarzanie obrazów odbywa się etapami. Przed każdym etapem następuje rozdzielenie (ang. *fork*), natomiast po ukończeniu etapu następuje scalenie sterowania (ang. *join*).

1.2.2. Optymalizacja kodu

Aby jeszcze dodatkowo skrócić czas przetwarzania obrazów, postarano się zoptymalizować kod programu - m.in. zastosowano łączenie pętli (ang. *loop fusion*) - oraz spłaszczanie pętli (ang. *loop collapsing*). Przykładowo, w jednej pętli zerowane są piksele z kilku obrazów równocześnie, dla tych obrazów które tego wymagają.

Kolejnym usprawnieniem było ominięcie kopiowania obrazów, służących wyłącznie do prezentacji, wtedy kiedy nie są obserwowane - jak np. dwóch ostatnich obrazów z grupy G .

2. Manipulator robota

2.1. Rzeczywisty manipulator robota

Aby sterować rzeczywistym manipulatorem, system powinien być wyposażony w moduł umożliwiający zadawanie pozycji oraz monitorowanie zmiennych procesowych. Dodatkowo użytkownik potrzebuje obserwować manipulator wykonujący jego polecenia, zatem pojawia się potrzeba zainstalowania kamery obserwującej działanie manipulatora.

Aby zrealizować w/w cele, zasymulowano 2 manipulatory - jeden w pliku DLL (napisany w C++), drugi w MATLABie (komunikacja poprzez protokół TCP/IP). Do systemu dołączono również czwartą kamerę, której zadaniem jest obserwowanie manipulatora.

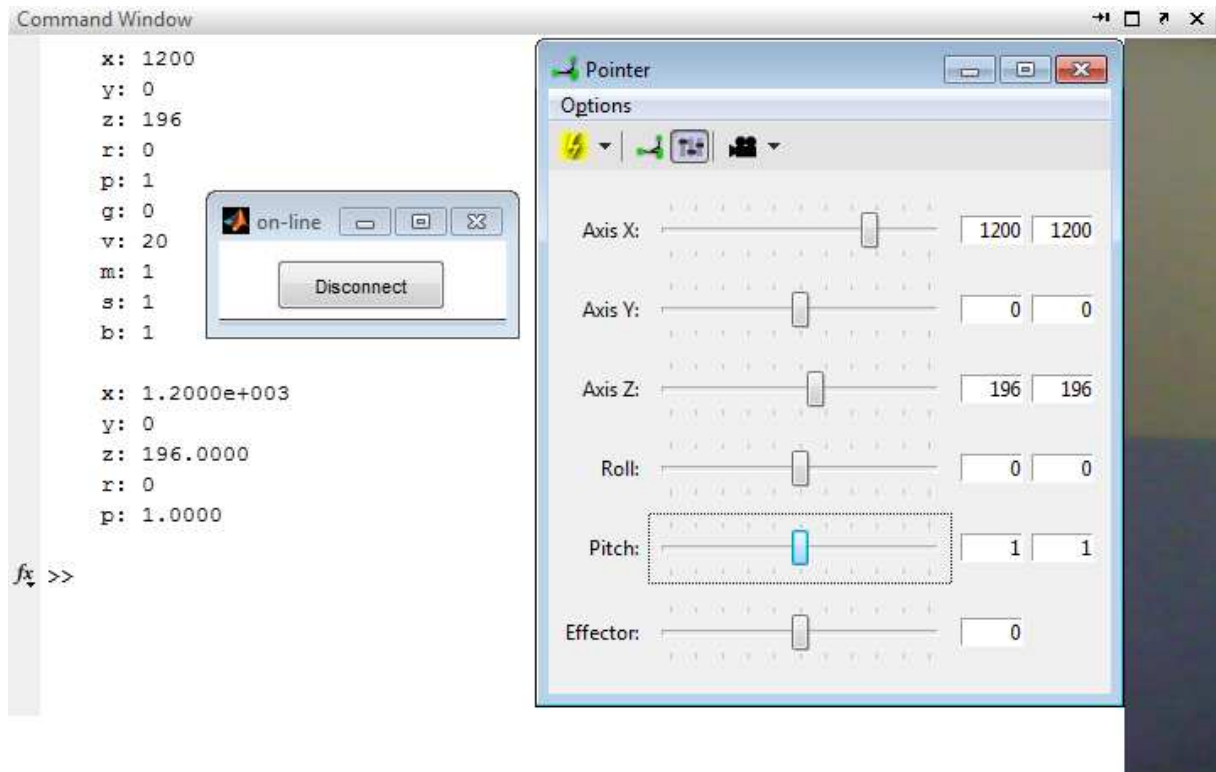
1. Moduł sterujący manipulatorem.

Domyślnie sterowanie odbywa się przy pomocy protokołu TCP/IP. Po stronie systemu sterującego, znajduje się serwer działający na porcie 9000, który oczekuje na połączenie z klientem (w tym przypadku symulatorem manipulatora napisanym w MATLABie).

Jeśli jednak w odpowiednim katalogu zostanie umieszczony plik DLL (ang. *Dynamic-link library*) - *Com.dll*, wówczas sterowanie będzie odbywać się za pośrednictwem właśnie tego pliku. Może być w nim zaimplementowany dowolny sposób komunikacji z manipulatorem - obecnie jest to jedynie testowy symulator manipulatora.

2. Kamera obserwująca manipulator.

Domyślnie strumień wideo pobierany jest bezpośrednio z portu USB. Dołączając plik *Cam.dll* można przesyłać strumień wideo dowolną drogą, w zależności od implementacji.



Rysunek 2.1: Sterowanie manipulatorem o 5 stopniach swobody (w MATLABie) za pośrednictwem protokołu TCP/IP

Na powyższym rysunku widoczne jest GUI głównej aplikacji, tj. pulpit sterowniczy oraz obraz z kamery obserwującej manipulator (fragment widoczny po prawej stronie). W tle znajduje się wiersz poleceń MATLABa oraz GUI z przyciskiem *Połącz/Rozłącz*.

1. Procedura uruchamiania: najpierw uruchamiany jest serwer, następnie klient.
2. Procedura wyłączenia: jeśli jedna ze stron zakończy połączenie, druga również automatycznie kończy pracę.

Żeby zasymulować pracę manipulatora, nie było konieczności wiernego odtwarzania dynamiki oraz pracy serwomechanizmów. Zamiast tego, jedynie zasymulowano aperiodyczne dochodzenie przegubów manipulatora do wartości zadanych. Równocześnie został uzyskany efekt quasiliniowego ruchu całego manipulatora. Efekt ten uzyskano poprzez mnożenie uchybu przez arbitralnie dobrany współczynnik wzmocnienia k oraz odejmowanie tej wartości od bieżącej wartości procesowej:

$$e_i = w_i - y_i \quad , \text{dla } i = 1, 2, \dots, 5 \quad (2.1)$$

$$y_i := y_i - ke_i \quad , \text{dla } i = 1, 2, \dots, 5 \quad (2.2)$$

e_i - uchyb w danym złączu

w_i - wartość zadana danego złącza

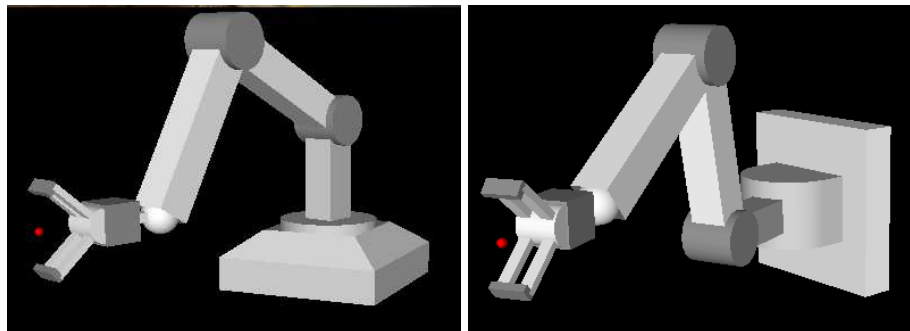
y_i - zmienna procesowa danego złącza

k - współczynnik wzmocnienia ($k = 0,05$, taki sam dla wszystkich złączy)

Algorytm ten zastosowano również w głównej części programu, napisanej w języku Java, tj. w manipulatorach wirtualnych, mających 6 stopni swobody.

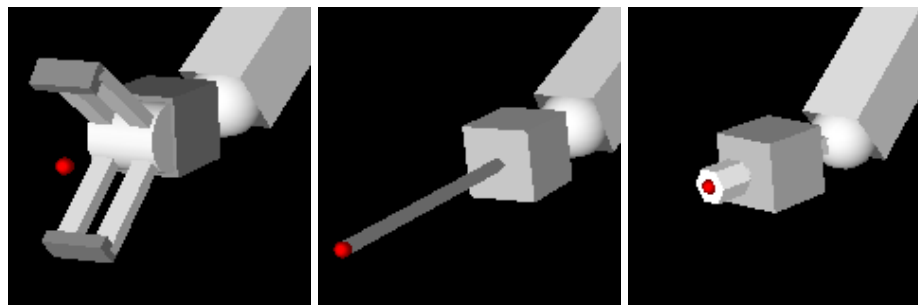
2.2. Wirtualny manipulator robota

Wirtualne manipulatory robotów zostały zaprojektowane w celu testowania systemu wizyjnego. W tym podrozdziale zostaną równoległe omówione dwa typy manipulatorów: manipulator robota terenowego oraz manipulator robota podwodnego.



Rysunek 2.2: Wirtualne manipulatory robota terenowego oraz podwodnego

W poprzednim podrozdziale 2.1 został omówiony sposób dochodzenia manipulatorów do zadanej pozycji oraz orientacji, bez konieczności modelowania ich dynamiki. Tutaj zastosowano ten sam algorytm.



Rysunek 2.3: Trzy rodzaje efektorów: chwytak, wiertarka oraz wkrętak

Manipulator może być wyposażony w dowolny rodzaj efektora, spośród niżej wymienionych:

- chwytak - sterowanie 2-stanowe
- chwytak - sterowanie 3-stanowe
- chwytak - sterowanie ciągłe
- wiertarka - sterowanie 2-stanowe
- wkrętak - sterowanie 3-stanowe
- wkrętak - sterowanie ciągłe

Zastosowane manipulatory są typu RRR, wszystkie ich złącza są rotacyjne, posiadają 6 stopni swobody, osiągalne przestrzenie robocze są przestrzeniami toroidalnymi. Zastosowana kiść sferyczna upraszcza analizę oraz umożliwia osiągnięcie zadanego punktu z określoną orientacją. Niejednoznaczność rozwiązania odwrotnego została usunięta dzięki ograniczeniom mechanicznym w konstrukcji manipulatorów. Poniższe tabele szczegółowo opisują parametry manipulatorów.

	$\theta_{i,min}[^\circ]$	$\theta_{i,max}[^\circ]$	$l_j[mm]$
1	-180	180	400
2	-60	90	500
3	0	130	500
4	-90	90	-
5	-90	90	-
6	-180	180	40

Rysunek 2.4: Parametry manipulatora robota terenowego

	$\theta_{i,min}[^\circ]$	$\theta_{i,max}[^\circ]$	$l_j[mm]$
1	-90	90	400
2	-30	210	500
3	0	130	500
4	-90	90	-
5	-90	90	-
6	-180	180	40

Rysunek 2.5: Parametry manipulatora robota podwodnego

2.2.1. Kinematyka prosta

Proste zadanie kinematyki polega na obliczeniu pozycji oraz orientacji efektora (końcówki roboczej manipulatora), na podstawie zmiennych konfiguracyjnych (złączowych).

Aby ujednolicić sposób przypisywania układów współrzędnych w poszczególnych parach kinematycznych oraz uprościć obliczenia, zastosowano notację Denavita-Hartenberga [4][19][18], która obejmuje następujące przekształcenia:

1. Obrót wokół osi Z o kąt θ_i (równoległość osi X_i i X_{i-1})
2. Przesunięcie wzdłuż osi Z o d_i
3. Przesunięcie wzdłuż osi X o a_i
4. Obrót wokół osi X o kąt α_i

$$\mathbf{T}_{i-1}^i = \mathbf{T}_{rot,Z,\theta_i} \mathbf{T}_{tr,Z,d_i} \mathbf{T}_{tr,X,a_i} \mathbf{T}_{rot,X,\alpha_i} \quad (2.3)$$

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

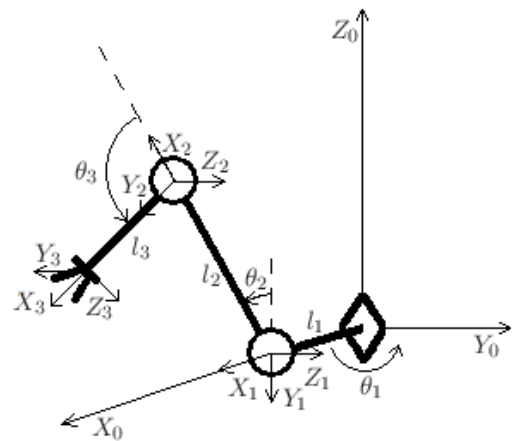
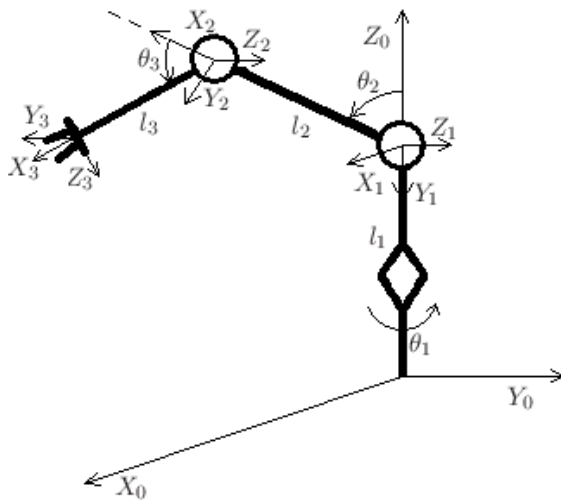
Poniższe tabele przedstawiają zestaw parametrów D-H dla poszczególnych członów i przegubów obydwu manipulatorów.

	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	-90°
2	$\theta_2 - 90^\circ$	0	l_2	0°
3	θ_3	0	l_3	-90°
4	θ_4	0	0	-90°
5	$\theta_5 + 90^\circ$	0	0	90°
6	$\theta_6 - 90^\circ$	l_6	0	-90°
6'	-90°	0	0	0°

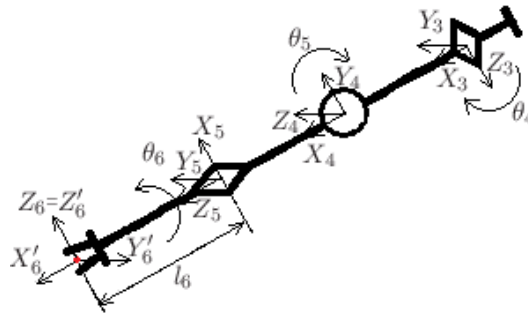
Rysunek 2.6: Parametry D-H dla manipulatora robota terenowego

	θ_i	d_i	a_i	α_i
1	θ_1	0	l_1	-90°
2	$\theta_2 - 90^\circ$	0	l_2	0°
3	θ_3	0	l_3	-90°
4	θ_4	0	0	-90°
5	$\theta_5 + 90^\circ$	0	0	90°
6	$\theta_6 - 90^\circ$	l_6	0	-90°
6'	-90°	0	0	0°

Rysunek 2.7: Parametry D-H dla manipulatora robota podwodnego



Rysunek 2.8: Łańcuchy kinematyczne manipulatorów



Rysunek 2.9: Łańcuch kinematyczny kiści sferycznej oraz efektora

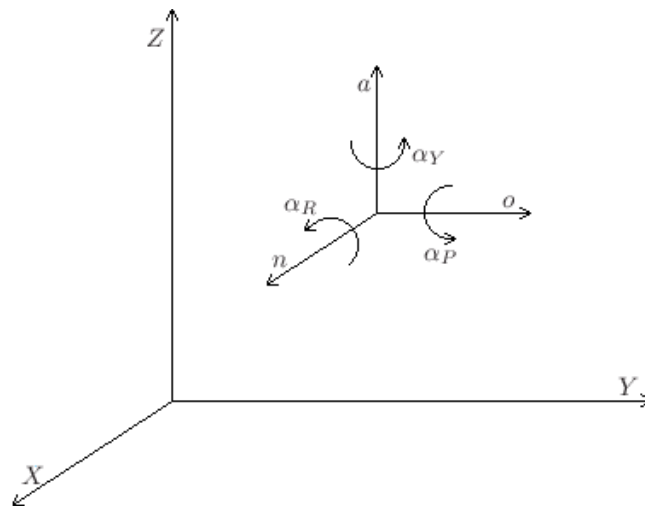
Dla uproszczenia rozważań przyjęto, że T_5^6 obejmuje przekształcenie 6 i 6', gdzie 6' koryguje orientację efektora: $T_{\text{rot},Z,-90^\circ}$.

$$T_0^6 = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 \quad (2.5)$$

2.2.2. Kinematyka odwrotna dla kątów RPY

Jednym ze sposobów zapisu orientacji są kąty RPY (ang. *Roll-Pitch-Yaw*). Jednak aby móc wyrazić orientację w tej postaci, należy wcześniej obliczyć zadanie kinematyki odwrotnej dla kątów RPY. Mając rozwiązanie zadania prostego kinematyki, jesteśmy w stanie wyznaczyć te kąty.

Aby obliczenia przeprowadzać w zakresie od $-\pi$ do π , do obliczeń zastosowano odpowiednik $\arctan - \text{atan2}^1$.



Rysunek 2.10: Kąty RPY

Układ współrzędnych noa odpowiada układowi $X_6'Y_6'Z_6'$ z poprzedniego podrozdziału 2.2.1 (rys. 2.9). Z poniższego układu równań należy wyliczyć szukane α_R , α_P oraz α_Y [8][15]:

$$\text{RPY}(\mathbf{r}_a, \mathbf{r}_o, \mathbf{r}_n) = \text{Rot}(\mathbf{a}, \alpha_Y) \text{Rot}(\mathbf{o}, \alpha_P) \text{Rot}(\mathbf{n}, \alpha_R) \quad (2.6)$$

$$\text{Rot}(\mathbf{a}, \alpha_Y)^{-1} \text{RPY}(\mathbf{r}_a, \mathbf{r}_o, \mathbf{r}_n) = \text{Rot}(\mathbf{o}, \alpha_P) \text{Rot}(\mathbf{n}, \alpha_R) \quad (2.7)$$

¹Metoda atan2 znajduje się w jednej z podstawowych bibliotek języka Java - `java.lang.Math`

Pamiętając, że macierz $\mathbf{Rot}(\mathbf{a}, \alpha_Y)$ jest macierzą ortogonalną [19], stąd $\mathbf{Rot}(\mathbf{a}, \alpha_Y)^{-1} = \mathbf{Rot}(\mathbf{a}, \alpha_Y)^T$, równanie można zapisać w postaci²:

$$\begin{bmatrix} C_Y & -S_Y & 0 & 0 \\ S_Y & C_Y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_P & 0 & S_P & 0 \\ 0 & 1 & 0 & 0 \\ -S_P & 0 & C_P & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_R & -S_R & 0 \\ 0 & S_R & C_R & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} n_x C_Y + n_y S_Y & o_x C_Y + o_y S_Y & a_x C_Y + a_y S_Y & 0 \\ n_y C_Y - n_x S_Y & o_y C_Y - o_x S_Y & a_y C_Y - a_x S_Y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_P & S_P S_R & S_P C_R & 0 \\ 0 & C_R & -S_R & 0 \\ -S_P & C_P S_R & C_P C_R & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Rozwiązując układy równań otrzymano³:

$$(2, 1) : \alpha_Y = \operatorname{atan2}\left(\frac{n_y}{n_x}\right) \quad (2.10)$$

$$(3, 1) \text{ oraz } (1, 1) : \alpha_P = \operatorname{atan2}\left(\frac{-n_z}{n_x C_Y + n_y S_Y}\right) \quad (2.11)$$

$$(2, 2) \text{ oraz } (2, 3) : \alpha_R = \operatorname{atan2}\left(\frac{a_x S_Y - a_y C_Y}{o_y C_Y - o_x S_Y}\right) \quad (2.12)$$

2.2.3. Kinematyka odwrotna

Odwrotne zadanie kinematyki zostało zrealizowane przy użyciu metod numerycznych, z uwagi na typ manipulatora oraz dużą ilość stopni swobody. Zastosowano dwie alternatywne metody numeryczne: metodę jacobianu odwrotnego oraz metodę jacobianu transponowanego [3]. W systemie istnieje możliwość wyboru, którą z tych metod chcemy zastosować do celu sterowania wirtualnym manipulatorem.

Wyżej wymienione metody różnią się między sobą głównie zbieżnością oraz podatnością manipulatora na wpadanie w oscylacje. Algorytmy implementujące te metody zostały wyposażone w mechanizm zapobiegający oscylacjom, np. w metodzie jacobianu odwrotnego zostało to uzyskane dzięki kontroli zbieżności. W metodach numerycznych, obliczających odwrotne zadanie kinematyki, istnieje silna zależność rozwiązania od punktu startowego. Dla obydwu manipulatorów, punkt ten został dobrany arbitralnie.

Celem algorytmu jest obliczenie wartości w przestrzeni konfiguracyjnej, przy spełnieniu określonej dokładności w przestrzeni kartezjańskiej.

²Przykładowo zapis C_R oznacza $\cos(\alpha_R)$

³Przykładowo zapis (2,1) oznacza drugi wiersz, pierwszą kolumnę

Oznaczenia:

\mathbf{t} - wektor docelowego położenia w układzie kartezjańskim $([x, y, z, R, P, Y]^T)$

\mathbf{s} - wektor położenia w układzie kartezjańskim

$\boldsymbol{\theta}$ - wektor położenia w układzie konfiguracyjnym

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \quad (2.13)$$

$$\mathbf{e} = \mathbf{t} - \mathbf{s} \quad (2.14)$$

1. Równanie jacobianu odwrotnego:

$$\Delta\boldsymbol{\theta} = \alpha \mathbf{J}^{-1} \mathbf{e} \quad (2.15)$$

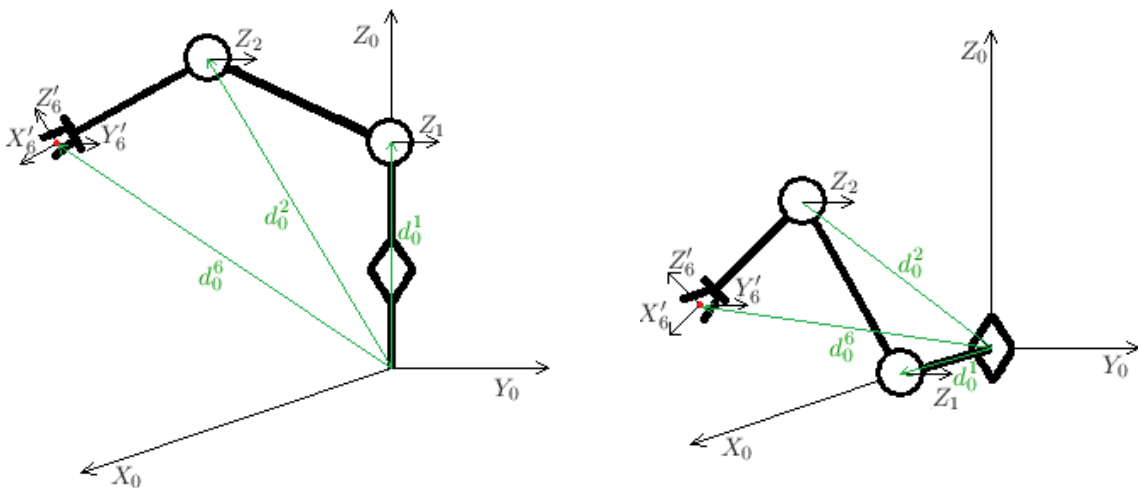
$$\alpha = \begin{cases} 0,5 & \text{dla } |t_6| < 85 \\ 0,0003 & \text{dla } |t_6| \geq 85 \end{cases} \quad (2.16)$$

2. Równanie jacobianu transponowanego:

$$\Delta\boldsymbol{\theta} = \alpha \mathbf{J}^T \mathbf{e} \quad (2.17)$$

$$\alpha = \frac{\langle \mathbf{e}, \mathbf{J}\mathbf{J}^T \mathbf{e} \rangle}{\langle \mathbf{J}\mathbf{J}^T \mathbf{e}, \mathbf{J}\mathbf{J}^T \mathbf{e} \rangle} \quad (2.18)$$

Obliczanie jacobianu:



Rysunek 2.11: Łańcuchy kinematyczne manipulatorów

$$\mathbf{J} = [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \mathbf{J}_3 \quad \mathbf{J}_4 \quad \mathbf{J}_5 \quad \mathbf{J}_6] \quad (2.19)$$

Wszystkie złącza są obrotowe dlatego dla każdego złącza stosowany jest wzór [20]:

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_i \times (\mathbf{d}_0^6 - \mathbf{d}_0^i) \\ \mathbf{z}_i \end{bmatrix} \quad (2.20)$$

Powyższe wektory uzyskuje się z macierzy:

$$\mathbf{T}_0^i = \begin{bmatrix} \mathbf{R}_0^i & \mathbf{d}_0^i \\ 0 & 1 \end{bmatrix} \quad (2.21)$$

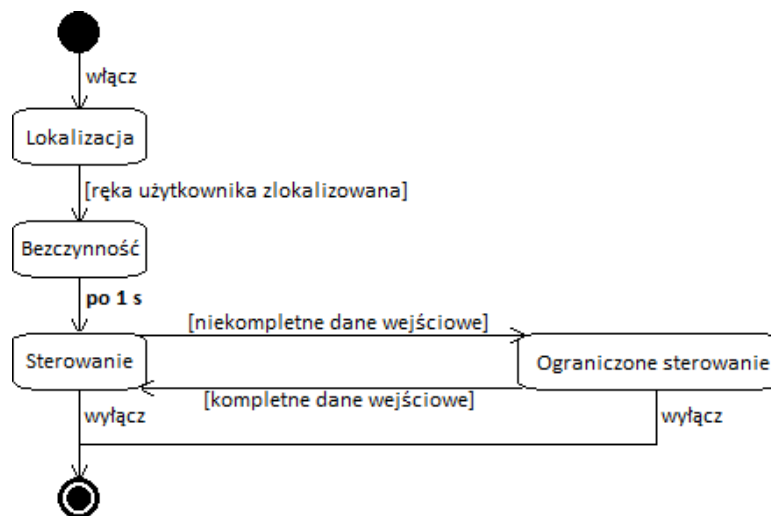
gdzie \mathbf{z}_i jest trzecią kolumną macierzy orientacji \mathbf{R}_0^i .

3. System wizyjny

3.1. Stany systemu wizyjnego

Opisany we wstępie system wizyjny może znajdować się w następujących stanach:

- *Lokalizacja* - system lokalizuje rękę użytkownika
- *Bezczynność* - użytkownik otrzymuje 1 sekundę na zatrzymanie ręki
- *Sterowanie* - sterowanie manipulatorem, systemem, oraz efektem
- *Ograniczone sterowanie* - sterowanie samym systemem oraz efektem (ma miejsce kiedy podsystem stereowizyjny nie jest w stanie odczytać pozycji oraz orientacji wskaźnika)

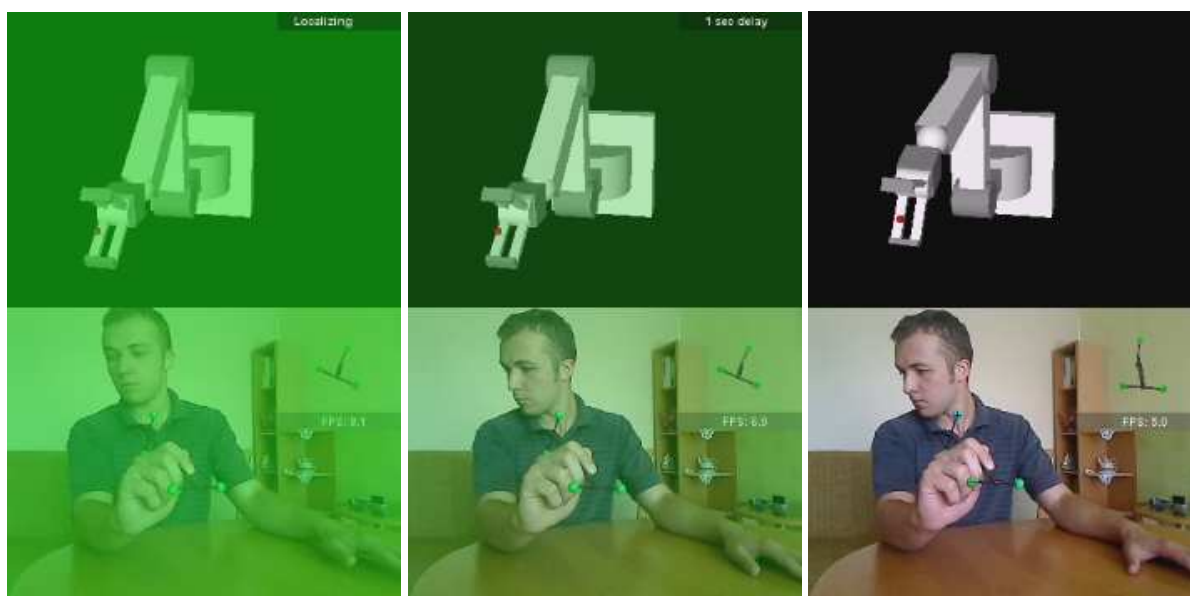


Rysunek 3.1: Diagram maszyny stanowej

3.2. Procedura uruchamiania system sterującego

Pierwszym etapem procedury uruchamiania systemu sterującego jest lokalizacja ręki. Po ukończeniu tego etapu użytkownik otrzymuje 1 sekundę na zatrzymanie ręki i dopiero wtedy zostaje uruchomiony system sterujący (przejdzie do stanu *Sterowanie*). Początkowymi wartościami zadawanymi manipulatorowi są wartości z pulpitu sterowniczego, czyli ostatnia pozycja manipulatora¹ oraz wartość zadana dla efektora.

W procedurze uruchamiania systemu sterującego istotne jest aby wskaźnik był trzymany w odpowiedni sposób. Mianowicie, w momencie przejścia systemu wizyjnego do stanu *Sterowanie* dwa przednie obiekty wskaźnika, tj obiekt nr 1 oraz obiekt nr 2, znajdują się niżej niż pozostałe dwa.



Rysunek 3.2: Uruchamianie systemu sterującego

W prawym górnym rogu ekranu znajduje się informacja o stanie systemu. Dodatkowo, aby ułatwić użytkownikowi zorientowanie się, w jakim aktualnie stanie znajduje się system, nakładana jest na obraz półprzezroczysta zielona warstwa - inny odcień dla stanu *Lokalizacja*, inny dla stanu *Bezczynność*.

W momencie przejścia w stan *Sterowanie* zaczyna się przetwarzanie stereopary. Dopiero wtedy wysyłane są do manipulatora jakiegokolwiek komendy (widoczna zmiana orientacji manipulatora na ostatnim ujęciu).

¹Orientacja celowo została pominięta, ponieważ sterowanie przy ustawionym offsecie dla orientacji byłoby nieintuicyjne.

3.3. Sterowanie za pomocą gestów

Sterowanie systemem oraz efektozem odbywa się przy pomocy gestów wykonywanych lewą ręką. Na tym etapie przetwarzany jest tylko dolny fragment obrazu, ponieważ tam system spodziewa się wykrycia ręki użytkownika.

3.3.1. Lokalizacja

Celem lokalizacji jest dalsze ograniczenie powierzchni przetwarzanego okna w celu zminimalizowania zapotrzebowania na moc obliczeniową.

Przetwarzanie obrazu

Obrazami wejściowymi jest para: bieżący obraz oraz obraz z poprzedniej iteracji.



Rysunek 3.3: Obraz wejściowy - poprzednia klatka



Rysunek 3.4: Obraz wejściowy

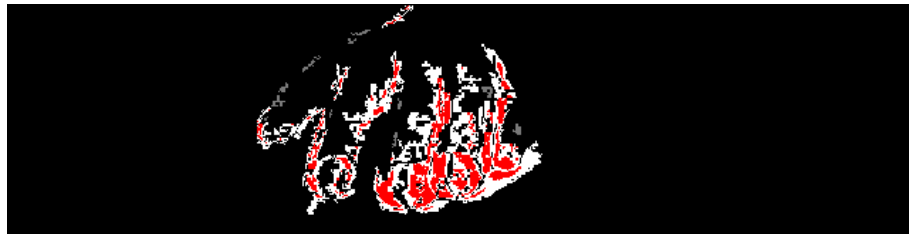
1. Różnica obrazów:



Rysunek 3.5: Różnica obrazów

W celu detekcji ruchu wykonywane jest odejmowanie obrazów. Obliczenia wykonywane są w przestrzeni barw RGB .

$$R = |R_{curr} - R_{prev}|, \quad G = |G_{curr} - G_{prev}|, \quad B = |B_{curr} - B_{prev}| \quad (3.1)$$

2. Rozrost obszaru (ang. *region growing*) [1]:

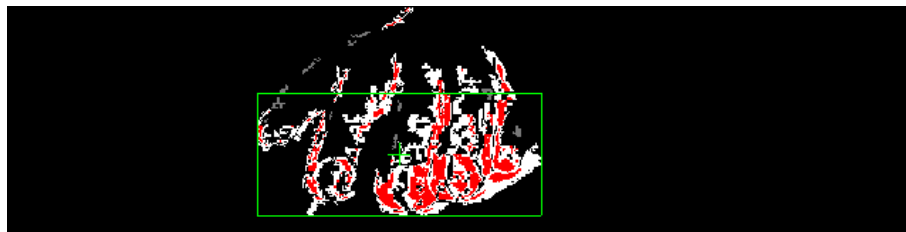
Rysunek 3.6: Rozrost obszaru

Rozrost obszaru może być traktowany jako dodanie analizy sąsiedztwa do techniki progowania. Obszar jest tworzony przez dwie klasy punktów: ziarna oraz punkty dołączane do obszaru. Ziarna muszą spełniać bardziej rygorystyczne kryteria niż punkty dołączane. Punktem startowym do rozrostu danego obszaru jest pierwsze napotkane ziarno, nieprzypisane wcześniej nigdzie do żadnego obszaru. Kolorem czerwonym oznaczane są ziarna, kolorem białym punkty przyłączone do obszaru. Stosowane jest 8-sąsiedztwo. Obliczenia są prowadzone w przestrzeni HSV . Poniżej lista zastosowanych kryteriów:

- Ziarna:
 - dolny próg dla V
 - dolny próg dla $\frac{S}{V}$.
- Punkty dołączane do obszaru:
 - dolny próg dla V
 - dolny próg dla S
 - dolny próg dla $\frac{S}{V}$.

Został ustawiony dodatkowy warunek, tj. dolny próg dla kanału S , ponieważ szum na obrazie, często ma kolor zbliżony do koloru szarego.

3. Obraz wynikowy:



Rysunek 3.7: Rozrost obszaru wraz z rozwiązaniem



Rysunek 3.8: Obraz wejściowy wraz z rozwiązaniem

W końcowym etapie przetwarzania została wyznaczona ramka odpowiadająca obrysowi ręki użytkownika.

Analiza obrazu

Ramka otrzymana w procesie przetwarzania jest następnie poddawana analizie. Parametrem stałym ramki jest jej wysokość - tutaj $0,18h$, gdzie h jest wysokością obrazu. Zmiennymi są pozycja, tj. środek ramki, oraz szerokość. Aby zminimalizować wpływ zakłóceń na otrzymany wynik, sprawdzane są kryteria:

- górny próg dla zmiany szerokości ramki względem poprzedniej iteracji
- histereza dla szerokości ramki
- dolny i górny próg dla $\frac{w}{h}$, gdzie w to szerokość a h wysokość ramki
- dolny próg dla $\frac{n_s}{n}$, gdzie n_s to ilość ziaren, n to pole analizowanego obszaru

Jeśli którekolwiek z nich nie zostanie spełnione, wówczas licznik iteracji zakończonych sukcesem zostanie wyzerowany. Aby uruchomić system sterujący, tj. przejść ze stanu *Lokalizacja* do stanu *Sterowanie*, wymagana jest określona liczba iteracji zakończonych sukcesem (tutaj $n = 5$).

3.3.2. Sterowanie systemem oraz efektozem

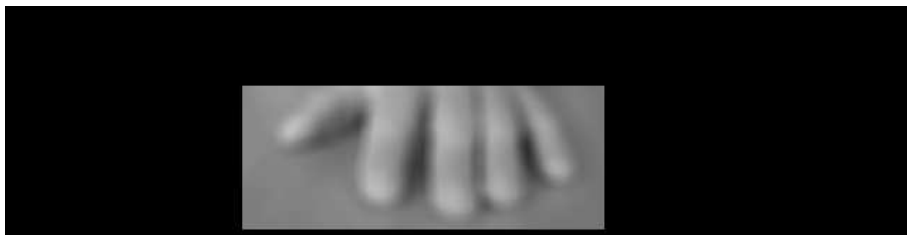
Okno wyznaczone podczas lokalizacji, na bieżąco dostosowuje się kształtami oraz pozycją do ręki wykrytej na obrazie z kamery. Jak już wspomniano, stwarza to możliwość sterowania systemem oraz efektozem przy stosunkowo niewielkim koszcie obliczeniowym.

Przetwarzanie obrazu



Rysunek 3.9: Obraz wejściowy

1. Filtracja dolnoprzepustowa [16]



Rysunek 3.10: Filtracja dolnoprzepustowa

Aby usunąć elementy obrazu o wysokiej częstotliwości, tj. pozbyć się artefaktów występujących na powierzchni stołu, małych obiektów, a także odfiltrować szum, zastosowano filtr uśredniający o masce 11x11:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

2. Detekcja krawędzi



Rysunek 3.11: Detekcja krawędzi

W celu detekcji krawędzi zastosowano operator Sobela [2]. Największy spośród modułów gradientu, zwracanych przez maski, określa wartość danego piksela w przetworzonym obrazie:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

Jeśli najjaśniejszy z pikseli obrazu przetworzonego (L_{max}) będzie poniżej zadanego progu, wówczas system zrywa komunikację z manipulatorem. Powodem tego jest zbyt niska wiarygodność systemu wizyjnego w danych warunkach oświetleniowych.

3. Rozrost obszaru

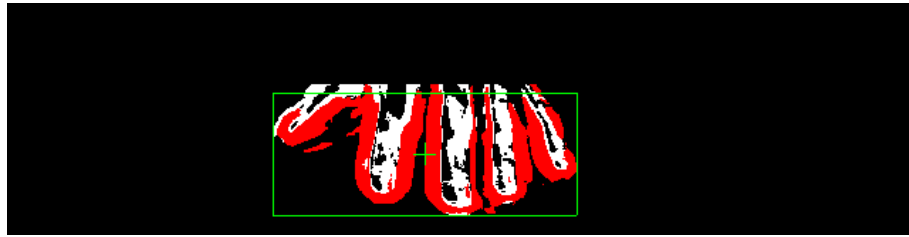


Rysunek 3.12: Rozrost obszaru

Lista zastosowanych kryteriów:

- Ziarna:
 - dolny próg dla $\frac{L}{L_{max}}$, gdzie luminacja (jasność) [13] jest zdefiniowana jako $L=R+G+B$ oraz L_{max} jest luminacją najjaśniejszego piksela obrazu
- Punkty dołączane do obszaru:
 - dolny próg dla $\frac{L}{L_{max}}$
 - górny próg dla H w obrazie wejściowym
 - górny próg dla S w obrazie wejściowym
 - dolny próg dla V w obrazie wejściowym

4. Obraz wynikowy



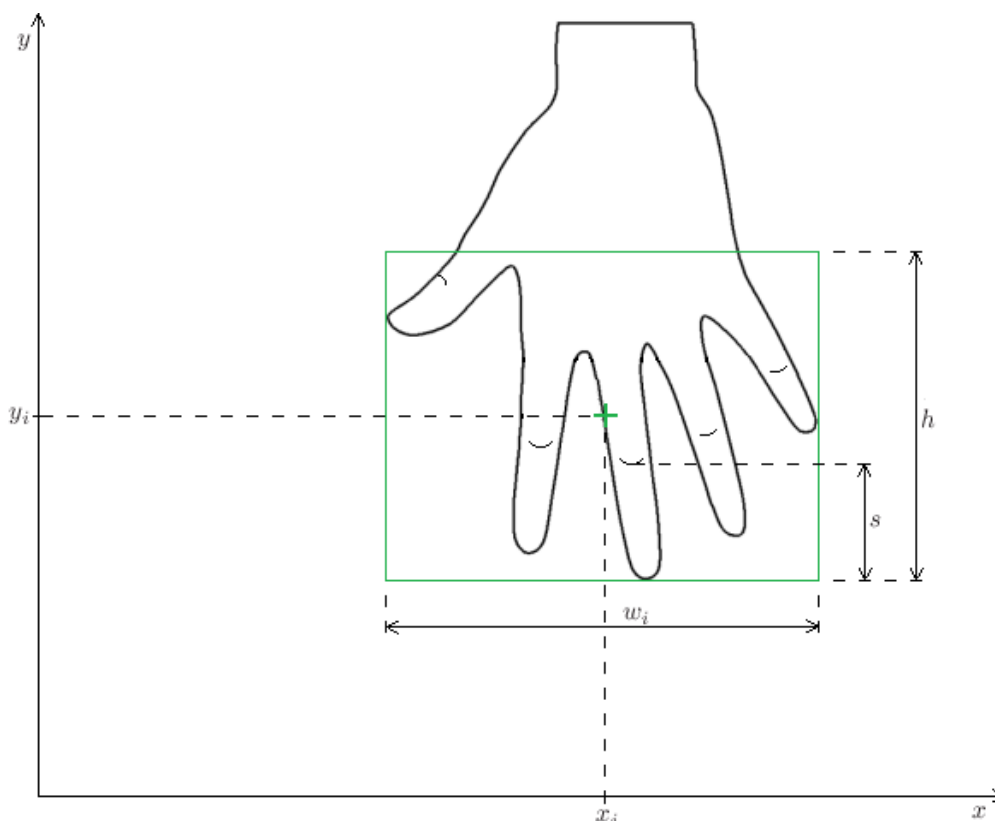
Rysunek 3.13: Rozrost obszaru wraz z rozwiązaniem



Rysunek 3.14: Obraz wejściowy wraz z rozwiązaniem

Analiza

Przetworzony obraz jest następnie poddawany analizie, której celem jest zamiana parametrów ramki na użyteczne wartości, przy pomocy których można sterować systemem oraz efektem.



Rysunek 3.15: Analiza przetworzonego obrazu

Dla uproszczenia wszystkie obliczenia przeprowadzane są na zbiorach ciągłych. W praktycznym zastosowaniu, są to jednak zbiory dyskretne, których wielkość zależna jest od rozdzielczości kamery. Zakładając sytuację w której kamera ma rozdzielczość 640x480, zmienne x , y , w przyjmują wartości:

$$x = 0, 1, \dots, 639 \quad (3.2)$$

$$y = 0, 1, \dots, 479 \quad (3.3)$$

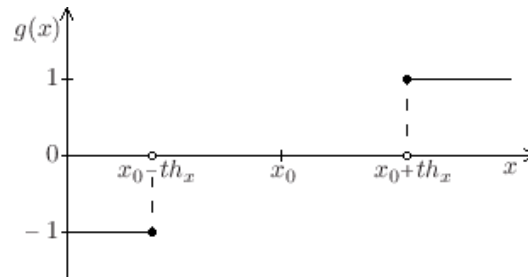
$$w = 0, 1, \dots, 639 \quad (3.4)$$

1. Sterowanie stanem systemu

Ruch ręki na zewnątrz lub zdjęcie ręki ze stołu oznacza zakończenie połączenia z manipulatorem, natomiast ruch ręki do wewnątrz powoduje przejście systemu w tryb sterowania manualnego - tj. przy pomocy pulpitu sterowniczego. Przejście odbywa się w sposób bezuderzeniowy, tj. zachowana zostaje pozycja i orientacja manipulatora oraz wartość sterująca efektem. Podobnie jest przy przechodzeniu z trybu manualnego do trybu sterowania przy użyciu systemu wizyjnego, z tą różnicą, że orientacja nie zostaje zachowana.

W każdej iteracji liczona jest proporcja $\frac{n_o}{n_b}$ gdzie n_o oznacza ilość pikseli obiektu w ramce, natomiast n_b ilość pikseli tła w ramce. Jeśli proporcja ta znajduje się poniżej określonego progu, wówczas połączenie z manipulatorem zostaje zakończone. Sytuacja ta odpowiada zdjęciu ręki ze stołu.

Poniżej przedstawiono sposób sterowania.



Rysunek 3.16: Funkcja $g(x)$

$$g(x) = \begin{cases} 1 & \text{dla } x \geq x_0 + th_x \\ 0 & \text{dla } x \in (x_0 - th_x, x_0 + th_x) \\ -1 & \text{dla } x \leq x_0 - th_x \end{cases} \quad (3.5)$$

th_x - parametr stały (próg)

$g(x_0) = 0$ - warunek początkowy

$g(x_0 - th_x) = -1$ - przejście do trybu sterowania ręcznego, tj. za pomocą pulpitu sterowniczego

$g(x_0 + th_x) = 1$ - zakończenie komunikacji z manipulatorem robota

2. Sterowanie efektozem

Parametry stałe (dobierane pod konkretnego użytkownika):

s - maksymalne przesunięcie wzdłuż osi OY uzyskane dzięki zgięciu samych palców

w_{min} - minimalny rozstaw palców

w_{max} - maksymalny rozstaw palców

Parametry zmienne:

$\Delta y = y - y_0$ - przesunięcie wzdłuż osi OY względem wartości początkowej

$\Delta w = w - w_0$ - zmiana szerokości w względem wartości początkowej

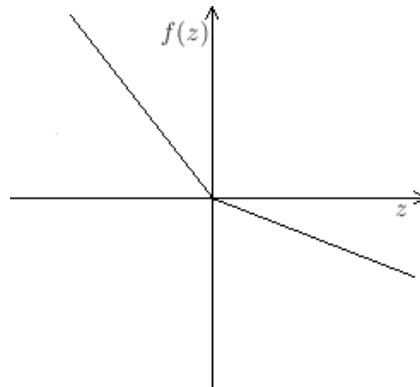
Parametr s odpowiada przesunięciu zielonej ramki kiedy użytkownik zgina palce (np. przy sterowaniu chwytkiem, w celu uchwycenia przedmiotu). Trudno jest dobrać parametry w_{min} i w_{max} tak, żeby zawsze precyzyjnie oddawały rzeczywistość. W praktyce za każdym razem odległość ręki od kamery jest nieco inna, dlatego istotne jest manipulowanie zmienną y poprzez przysuwanie/odsuwanie ręki od/do kamery. Umożliwia to skompensowanie ewentualnych rozbieżności.

Funkcję $f(y, w)$ sprowadzamy do postaci $f(z)$.

$$f(z) = az \quad (3.6)$$

$$z = -\Delta y + \Delta w \quad (3.7)$$

$$a = \begin{cases} -\frac{f_{min}}{w_{min}-w_0-s} & \text{dla } w < w_0 \\ -\frac{f_{max}}{w_{max}-w_0} & \text{dla } w \geq w_0 \end{cases} \quad (3.8)$$



Rysunek 3.17: Funkcja $f(z)$

Współczynnik nachylenia funkcji a może być mniejszy lub większy dla jednej z półpłaszczyzn, w zależności od użytkownika. Aby otworzyć chwytak należy zwiększyć rozstaw palców, natomiast żeby uchwycić przedmiot należy wrócić do naturalnego ułożenia ręki, a następnie zgiąć palce. Istnieją również inne sposoby sterowania efektor, np. chwytakiem sterowanym 2-stanowo można sterować przy pomocy samego kciuka.

Żeby powrócić z trybu manualnego do trybu systemu wizyjnego, należy znaleźć takie y_0 , aby bieżąca wartość funkcji f odpowiadała ostatniej wartości zadanej efektora (tj. zadanej jeszcze przy pomocy pulpitu sterowniczego). Z kolei w_0 wyznaczany jest na podstawie aktualnych pomiarów, dlatego $\Delta w = 0$. Wynik uzyskujemy przekształcając równania (3.6) oraz (3.7)

$$f = -a(y - y_0) \quad (3.9)$$

$$y_0 = y + \frac{f}{a} \quad (3.10)$$

Wyżej opisany sposób dotyczy sterowania efektor, sposobem ciągłym. Do sterowania efektor, 2-stanowo lub 3-stanowo wykorzystuje się ten sam algorytm, z tą różnicą, że o wielkości sterującej stanowi relacja wartości funkcji f względem ustalonych progów.

3.4. Sterowanie za pomocą wskaźnika

3.4.1. Czteropunktowa metoda identyfikacji transformacji stereowizyjnej

W celu obliczenia współrzędnych obiektów wskaźnika w przestrzeni trójwymiarowej zastosowano czteropunktową metodę identyfikacji transformacji stereowizyjnej [10]. Metoda ta wykorzystuje transformacje płaskie, które posiadają minimalną ilość założeń odnośnie jakości sprzętu optycznego oraz precyzji pomiarów. Nie wymaga się również jednakowości skal ani ortogonalności osi w układach współrzędnych. Parametry optyczne kamer nie muszą być znane.

W tej części rozważań będą stosowane wektory \mathbf{p}_R , \mathbf{p}_M zaczepione w początku układu współrzędnych $(0, 0)$ oraz o końcach odpowiednio w punktach P_R , P_M , a także $\mathbf{p}_{R_1}^1$, $\mathbf{p}_{R_1}^2$, $\mathbf{p}_{R_2}^1$, $\mathbf{p}_{R_2}^2$, \mathbf{p}_{M_1} , \mathbf{p}_{M_2} , $\mathbf{p}_{G_1}^1$, $\mathbf{p}_{G_1}^2$, $\mathbf{p}_{G_2}^1$, $\mathbf{p}_{G_2}^2$ zaczepione w początku układu współrzędnych $(0, 0, 0)$ oraz o końcach odpowiednio w punktach $P_{R_1}^1$, $P_{R_1}^2$, $P_{R_2}^1$, $P_{R_2}^2$, P_{M_1} , P_{M_2} , $P_{G_1}^1$, $P_{G_1}^2$, $P_{G_2}^1$, $P_{G_2}^2$

Transformacja płaska

1. Kalibracja

Celem kalibracji układu było wyznaczenie wektora współczynników

$$\mathbf{w}^T = [a_1, b_1, c_1, a_2, b_2, c_2, \alpha, \beta] \quad (3.11)$$

odzwierciedlającego własności geometryczno-optyczne kamery oraz położenie obserwowanej płaszczyzny względem płaszczyzny kamery.

$$\mathbf{p}_M = \begin{bmatrix} x_M \\ y_M \end{bmatrix}, \quad \mathbf{p}_R = \begin{bmatrix} x_R \\ y_R \end{bmatrix} \quad (3.12)$$

Wyżej przedstawiony wektor \mathbf{p}_M odpowiada pozycji piksela na matrycy kamery, natomiast \mathbf{p}_R odpowiada punktowi na badanej płaszczyźnie umiejscowionej w przestrzeni trójwymiarowej.

Wybrano 4 punkty, spośród których żadne 3 nie leżą na tej samej prostej. Ostatecznie rozwiązano układ równań ze względu na wektor współczynników.

$$\begin{bmatrix} x_{R_1} \\ y_{R_1} \\ \vdots \\ x_{R_4} \\ y_{R_4} \end{bmatrix} = \begin{bmatrix} x_{M_1} & y_{M_1} & 1 & 0 & 0 & 0 & -x_{M_1}x_{R_1} & -y_{M_1}x_{R_1} \\ 0 & 0 & 0 & x_{M_1} & y_{M_1} & 1 & -x_{M_1}y_{R_1} & -y_{M_1}y_{R_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M_4} & y_{M_4} & 1 & 0 & 0 & 0 & -x_{M_4}x_{R_4} & -y_{M_4}x_{R_4} \\ 0 & 0 & 0 & x_{M_4} & y_{M_4} & 1 & -x_{M_4}y_{R_4} & -y_{M_4}y_{R_4} \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ \alpha \\ \beta \end{bmatrix} \quad (3.13)$$

Przyjmując oznaczenia

$$\mathbf{h} = \begin{bmatrix} x_{R_1} \\ y_{R_1} \\ \vdots \\ x_{R_4} \\ y_{R_4} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} x_{M_1} & y_{M_1} & \dots & -x_{M_1}x_{R_1} & -y_{M_1}x_{R_1} \\ 0 & 0 & \dots & -x_{M_1}y_{R_1} & -y_{M_1}y_{R_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M_4} & y_{M_4} & \dots & -x_{M_4}x_{R_4} & -y_{M_4}x_{R_4} \\ 0 & 0 & \dots & -x_{M_4}y_{R_4} & -y_{M_4}y_{R_4} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ \alpha \\ \beta \end{bmatrix} \quad (3.14)$$

otrzymujemy

$$\mathbf{h} = \mathbf{C}\mathbf{w} \quad (3.15)$$

$$\mathbf{w} = \mathbf{C}^{-1}\mathbf{h} \quad (3.16)$$

2. Obliczanie pozycji punktu na płaszczyźnie

Znając wektor współczynników, wyznaczony w procesie kalibracji, jesteśmy w stanie określić pozycję dowolnego punktu na badanej płaszczyźnie (wektor \mathbf{p}_R), na podstawie odpowiednika tego punktu na obrazie kamery (wektor \mathbf{p}_M).

Transformacja płaska ma postać:

$$T_P(\mathbf{p}_M) = \frac{1}{\alpha x_M + \beta y_M + 1} \begin{bmatrix} a_1 x_M + b_1 y_M + c_1 \\ a_2 x_M + b_2 y_M + c_2 \end{bmatrix} \quad (3.17)$$

$$\mathbf{p}_R = T_P(\mathbf{p}_M) \quad (3.18)$$

Transformacja typu stereo wykorzystująca transformacje płaskie

1. Kalibracja

W celu skalibrowania układu stereowizyjnego, obydwie kamery obserwują dwie płaszczyzny będące prostopadłe względem siebie oraz odpowiadające powierzchniom frontowym wzorca kalibracyjnego, z którym związany jest globalny układ współrzędnych $X_G Y_G Z_G$.

Przyjmijmy oznaczenia:

- $T_{P_1^1}$ transformacja płaska dla kamery 1, płaszczyzny 1
- $T_{P_1^2}$ transformacja płaska dla kamery 1, płaszczyzny 2
- $T_{P_2^1}$ transformacja płaska dla kamery 2, płaszczyzny 1
- $T_{P_2^2}$ transformacja płaska dla kamery 2, płaszczyzny 2

Ostatecznie należy wykonać 4 kalibracje, tj. dla wszystkich w/w transformacji płaskich.

2. Obliczanie pozycji punktu w przestrzeni trójwymiarowej

Mając dane wektory współczynników, jesteśmy w stanie obliczyć punkt w przestrzeni trójwymiarowej względem układu $X_G Y_G Z_G$. W tym celu należy dokonać transformacji płaskich: $T_{P_1^1}, T_{P_1^2}, T_{P_2^1}, T_{P_2^2}$.

$$\mathbf{PR}_1^1 = T_{P_1^1}(\mathbf{PM}_1) \quad (3.19)$$

$$\mathbf{PR}_1^2 = T_{P_1^2}(\mathbf{PM}_1) \quad (3.20)$$

$$\mathbf{PR}_2^1 = T_{P_2^1}(\mathbf{PM}_2) \quad (3.21)$$

$$\mathbf{PR}_2^2 = T_{P_2^2}(\mathbf{PM}_2) \quad (3.22)$$

Punkty $P_{R_1^1}$ oraz $P_{R_1^2}$ widziane są przez 1. kamerę jako jeden punkt P_{M_1} , podobnie punkty $P_{R_2^1}$ oraz $P_{R_2^2}$ widziane są przez 2. kamerę jako jeden punkt P_{M_2} .

Z powyższego wynika, że punkt P leży na przecięciu prostej przechodzącej przez punkty $P_{R_1^1}$ i $P_{R_1^2}$ oraz prostej przechodzącej przez punkty $P_{R_2^1}$ i $P_{R_2^2}$.

Przy pomocy znanych odwzorowań B_1 oraz B_2 zostały kolejnym punktem, leżącym na płaszczyznach, przyporządkowane współrzędne w globalnym układzie współrzędnych.

$$\mathbf{PG}_1^1 = B_1[\mathbf{PR}_1^1] \quad (3.23)$$

$$\mathbf{PG}_1^2 = B_2[\mathbf{PR}_1^2] \quad (3.24)$$

$$\mathbf{PG}_2^1 = B_1[\mathbf{PR}_2^1] \quad (3.25)$$

$$\mathbf{PG}_2^2 = B_2[\mathbf{PR}_2^2] \quad (3.26)$$

W celu dalszych obliczeń przyjęto oznaczenia

$$\mathbf{PG}_1^1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \quad \mathbf{PG}_1^2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \quad \mathbf{PG}_2^1 = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}, \quad \mathbf{PG}_2^2 = \begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix} \quad (3.27)$$

Najpierw rozwiązywany jest układ równań liniowych ze względu na zmienne x oraz y w celu znalezienia punktu przecięcia się prostych w rzucie na płaszczyznę $X_G Y_G$.

$$\begin{cases} \frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} \\ \frac{x-x_3}{x_4-x_3} = \frac{y-y_3}{y_4-y_3} \end{cases} \quad (3.28)$$

$$x = \frac{x_2 y_1 x_4 - x_2 y_1 x_3 - x_2 y_3 x_4 + x_1 y_3 x_4 - x_1 x_3 y_4 + x_1 y_2 x_3 - x_1 y_2 x_4 + x_2 x_3 y_4}{-y_2 x_4 + y_2 x_3 + y_1 x_4 - y_1 x_3 - y_4 x_1 + y_4 x_2 + y_3 x_1 - y_3 x_2} \quad (3.29)$$

$$y = \frac{y_2 x_3 y_4 + y_3 x_1 y_2 - y_4 x_1 y_2 - y_2 y_3 x_4 - y_1 x_3 y_4 + y_4 y_1 x_2 + y_1 y_3 x_4 - y_3 y_1 x_2}{-y_2 x_4 + y_2 x_3 + y_1 x_4 - y_1 x_3 - y_4 x_1 + y_4 x_2 + y_3 x_1 - y_3 x_2} \quad (3.30)$$

Następnie obliczana jest współrzędna z .

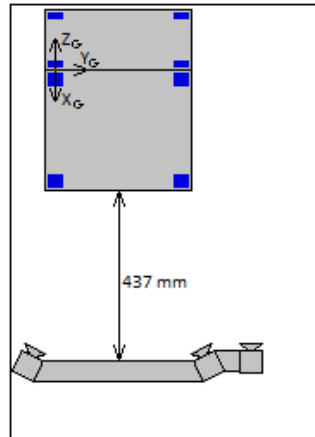
$$\frac{x - x_1}{x_2 - x_1} = \frac{z - z_1}{z_2 - z_1} \quad (3.31)$$

$$z = \frac{-z_1 x_2 - x z_2 + x z_1 + x_1 z_2}{-x_2 + x_1} \quad (3.32)$$

Ostatecznie otrzymano współrzędne punktu $P = (x, y, z)$ w układzie współrzędnych $X_G Y_G Z_G$.

3.4.2. Kalibracja

Aby dostarczyć danych wejściowych, tj. zestawu punktów kalibracyjnych, dla czteropunktowej metody identyfikacji transformacji stereowizyjnej (rozdz. 3.4.1), należało przeprowadzić kalibrację spełniającą jej wymagania.



Rysunek 3.18: Kalibracja

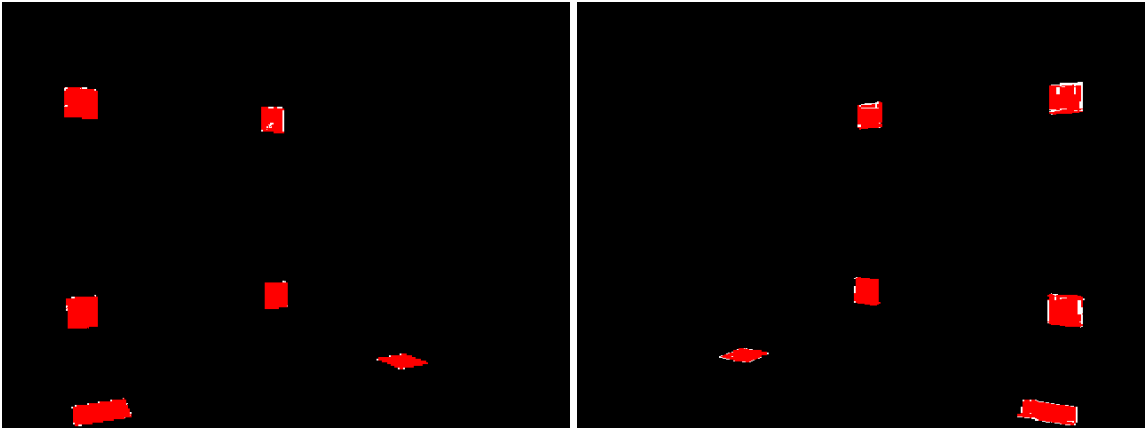
Na powyższym rysunku przedstawione jest optymalne umiejscowienie wzorca kalibracyjnego. Jego odległość od statywu z układem kamer wynosi 437 mm , natomiast oś OX_G jest prostopadła do linii statywu. Globalny układ współrzędnych $X_G Y_G Z_G$ jest punktem wyjściowym do zdefiniowania przestrzeni roboczej wskaźnika - więcej informacji w rozdziale poświęconym skalowaniu.

Przetwarzanie obrazu



Rysunek 3.19: Obrazy wejściowe

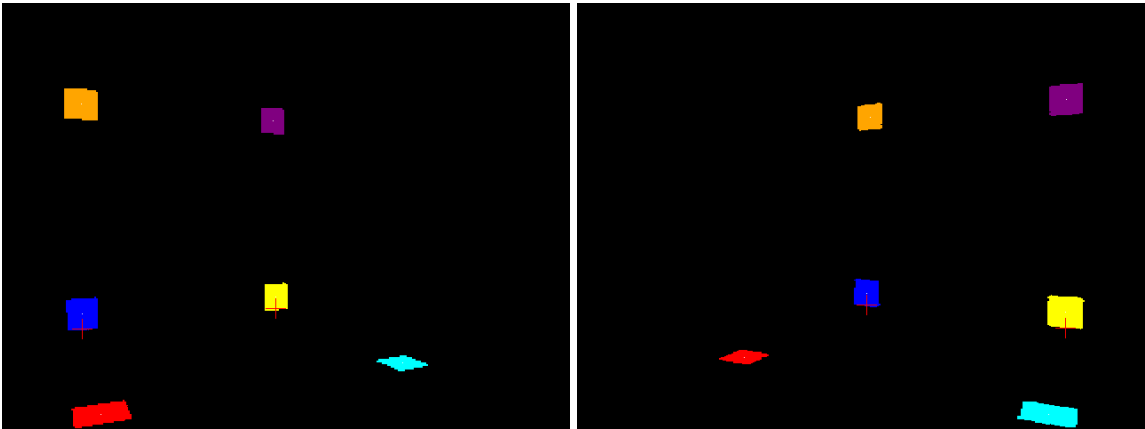
1. Rozrost obszaru



Rysunek 3.20: Rozrost obszaru

Lista zastosowanych kryteriów:

- Ziarna:
 - określona tolerancja dla kanału H
 - dolny próg dla kanału S
 - dolny próg dla kanału V
- Punkty dołączane do obszaru:
 - określona tolerancja dla kanału H
 - dolny próg dla kanału S
 - dolny próg dla kanału V
 - górny próg dla gradientu kanału H
 - górny próg dla gradientu kanału S
 - górny próg dla gradientu kanału V

2. Etykietowanie obiektów metodą *pożar prerii* [2]

Rysunek 3.21: Etykietowanie

Metoda *pożar prerii* pozwala wypełnić dany obszar określonym kolorem. Następnie znajduwany jest środek ciężkości obiektu, by ostatecznie przypisać mu indeks w zależności od położenia na ekranie. Tym sposobem na obrazach z obydwu kamer zostaje wyznaczony zestaw punktów $P_1, P_2, P_3, P_4, P_5, P_6$. Aby uprościć obliczenia identyfikacji transformacji stereowizyjnej, na podstawie w/w punktów zostają wyznaczone punkty leżące na linii przecięcia się obydwu płaszczyzn wzorca kalibracyjnego (prostopadłych względem siebie). W tym celu należało do współrzędnych z punktów P_3 oraz P_4 (środki ciężkości obiektów kolejno koloru niebieskiego oraz żółtego) dodać offsety

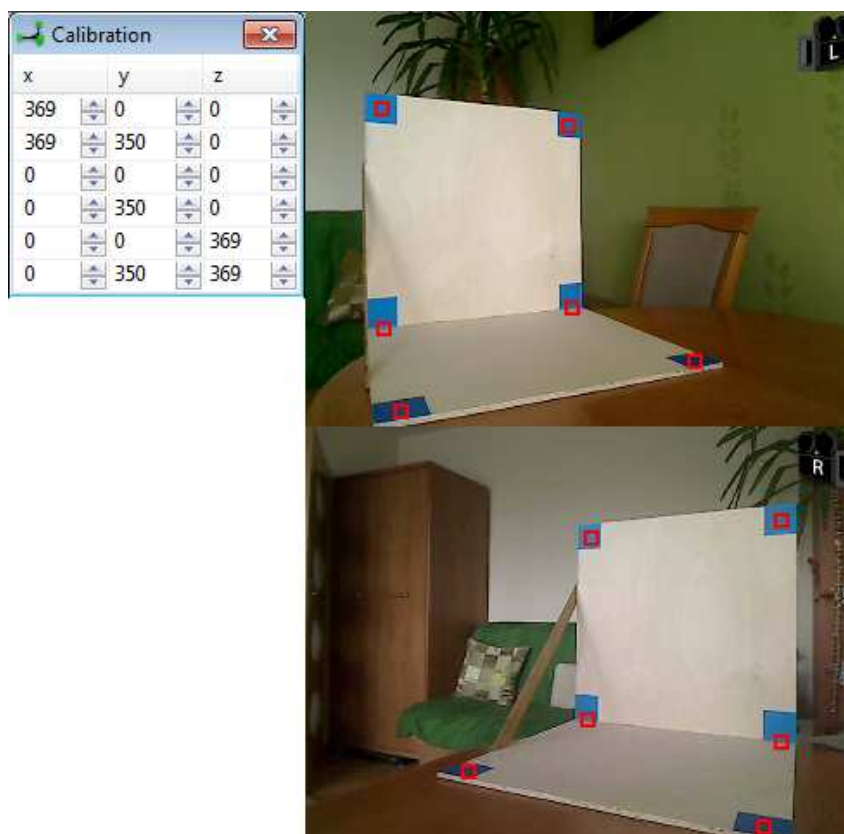
$$o = -\frac{\sqrt{n}}{2} + 1, \quad \text{gdzie } n \text{ jest polem powierzchni danego obiektu} \quad (3.33)$$

Na koniec, obszary, które w poprzednim etapie zostały zbudowane z ziaren oraz punktów dołączonych, muszą teraz spełnić kryteria:

- dolny próg dla $\frac{n_s}{n_a}$, gdzie n_s oznacza ilość ziaren, natomiast n_a ilość punktów dołączonych do obszaru
- dolny próg dla n_s - ilość ziaren w danym obiekcie

W przypadku nie spełnienia któregoś z powyższych warunków, obszar wypełniany jest kolorem szarym oraz zostaje uznany za zakłócenia.

Analiza obrazu



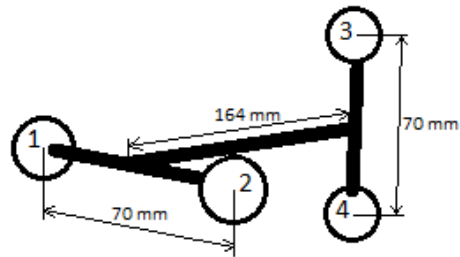
Rysunek 3.22: Współrzędne punktów oraz obrazy wejściowe wraz z rozwiązaniem

Aby skalibrować układ, należy podać położenia punktów kalibracyjnych w globalnym układzie współrzędnych $X_G Y_G Z_G$. Druga część danych jest dostępna po ostatnim etapie przetwarzania jakim jest etykietowanie. Znane są wówczas współrzędne monitorowe poszczególnych punktów kalibracyjnych. Tym sposobem zostają obliczone 4 wektory współczynników dla transformacji płaskich $T_{P_1^1}$, $T_{P_1^2}$, $T_{P_2^1}$ oraz $T_{P_2^2}$, co z kolei umożliwia obliczanie pozycji punktu w przestrzeni trójwymiarowej na podstawie obrazu z kamer układu stereowizyjnego.

3.4.3. Sterowanie manipulatorem robota

Do celów sterowania manipulatorem robota użyto wskaźnik zbudowany z 4 obiektów oraz szkieletu. Aby było możliwe określenie jego pozycji oraz orientacji, w każdej chwili co najmniej trzy obiekty powinny być widziane przez obydwie kamery.

Budowa wskaźnika została przedstawiona poniżej.



Rysunek 3.23: Wskaźnik

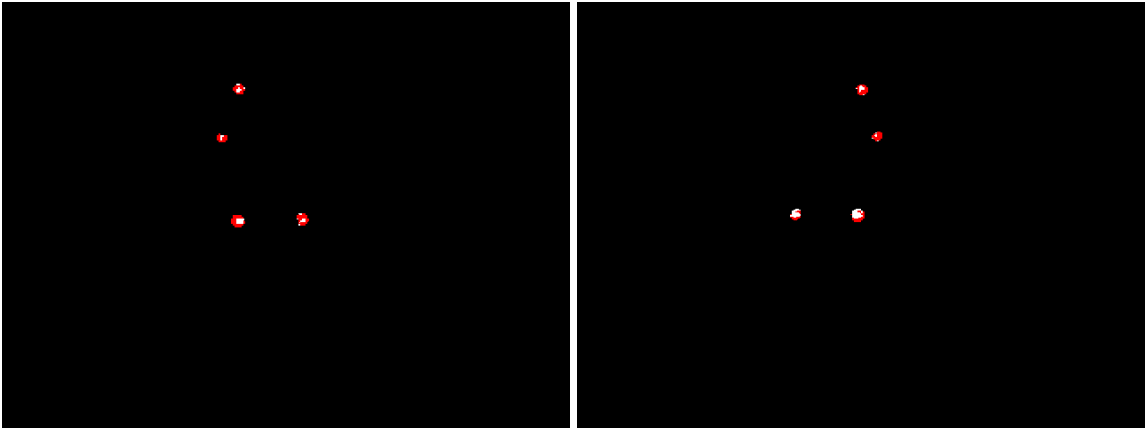
W celu uproszczenia analizy, wszystkie elementy szkieletu wskaźnika, są względem siebie ortogonalne. Taka budowa wskaźnika, spełnia swoje zadanie, ponieważ umożliwia zadawanie orientacji w szerokim zakresie kątów.

Przetwarzanie obrazu



Rysunek 3.24: Obrazy wejściowe

1. Rozrost obszaru

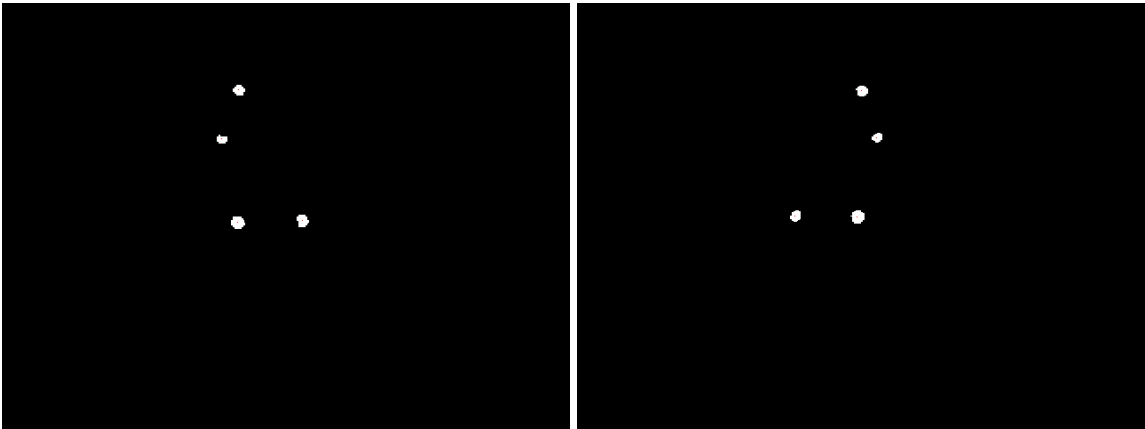


Rysunek 3.25: Rozrost obszaru

Lista zastosowanych kryteriów:

- Ziarna:
 - określona tolerancja dla kanału H
 - dolny próg dla kanału S
 - dolny próg dla kanału V
- Punkty dołączane do obszaru:
 - określona tolerancja dla kanału H
 - dolny próg dla kanału S
 - dolny próg dla kanału V
 - górny próg dla gradientu kanału H
 - górny próg dla gradientu kanału S
 - górny próg dla gradientu kanału V

2. Pożar prerii



Rysunek 3.26: Pożar prerii

Podobnie jak w przypadku kalibracji, obszary, które zostały zbudowane na etapie rozrostu obszaru muszą spełnić kryteria:

- dolny próg dla $\frac{n_s}{n_a}$, gdzie n_s oznacza ilość ziaren, natomiast n_a ilość punktów dołączonych do obszaru
- dolny próg dla n_s - ilość ziaren w danym obiekcie
- dolny oraz górny próg dla $\frac{w}{h}$, gdzie w oznacza szerokość, h wysokość obiektu

W przypadku nie spełnienia któregoś z powyższych warunków, obszar wypełniany jest kolorem szarym oraz zostaje uznany za zakłócenia.

3. Obrazy wynikowe



Rysunek 3.27: Obrazy wejściowe wraz z rozwiązaniem

Na powyższych ilustracjach widoczny jest sposób numeracji obiektów wskaźnika:

- 1. obiekt - kolor czerwony
- 2. obiekt - kolor cyjanowy
- 3. obiekt - kolor niebieski
- 4. obiekt - kolor żółty

Nadawanie obiektom indeksów, w tym przypadku, nie jest już tak trywialne jak podczas kalibracji. Sposób nadawania obiektom indeksów, zostanie omówiony w następnym podrozdziale, poświęconym analizie obrazu.

Analiza obrazu

Dobrze ponumerowane obiekty są warunkiem koniecznym do obliczenia pozycji oraz orientacji dla manipulatora. Aby w każdej iteracji obiektom wskaźnika były nadawane odpowiednie indeksy, należało zaimplementować zestaw algorytmów przeznaczonych do tego celu. W przypadku niekompletnej informacji wejściowej (np. kiedy widoczne są tylko 2 pary obiektów) system przechodzi w tryb *Ograniczone sterowanie*.

1. Algorytm śledzący obiekty wskaźnika

Algorytm śledzi obiekty na obrazach obydwu kamer. Odpowiadające sobie obiekty są następnie wykorzystywane do wyznaczenia położenia danego obiektu wskaźnika w przestrzeni trójwymiarowej - przy pomocy transformacji stereowizyjnej.

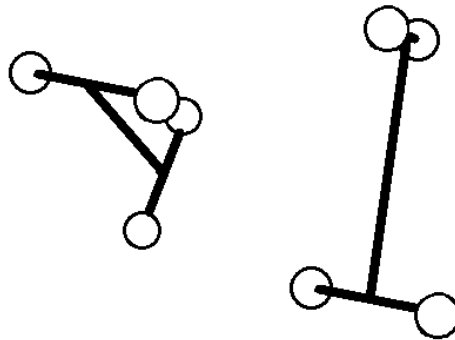
W celu zwiększenia skuteczności śledzenia obiektów, dodatkowo przewiduje się ich pozycje na podstawie poprzednich położzeń (osobny algorytm). Zatem bieżące pozycje obiektów porównywane są z pozycjami przewidywanymi w poprzedniej iteracji.

W przypadku, kiedy pojawi się obiekt, który był wcześniej niewidoczny, zostaje mu nadany, nie wykorzystywany przez żaden inny obiekt, indeks.

Inną częstą sytuacją jest zasłanianie obiektu przez rękę użytkownika. Wówczas indeks tego obiektu zostaje zwolniony. Zazwyczaj ten sam obiekt, kiedy ponownie będzie widoczny dla kamery, otrzymuje swój poprzedni indeks. Nie można jednak zakładać, że taka sytuacja będzie prawidłowością, dlatego w następnych punktach zostanie omówiony algorytm pozwalający rozwiązać ten problem.

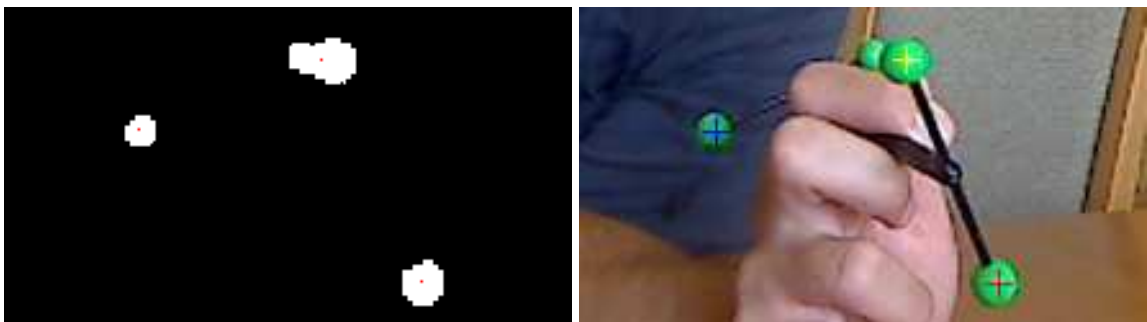
2. Algorytm wydobywający informację o pozycji obiektu zakrywającego drugi obiekt

Kolejnym problemem, który należało rozwiązać, było wzajemne zasłanianie się obiektów w pewnych ustawieniach wskaźnika. Poniżej ukazane są dwa rodzaje scenariuszy.



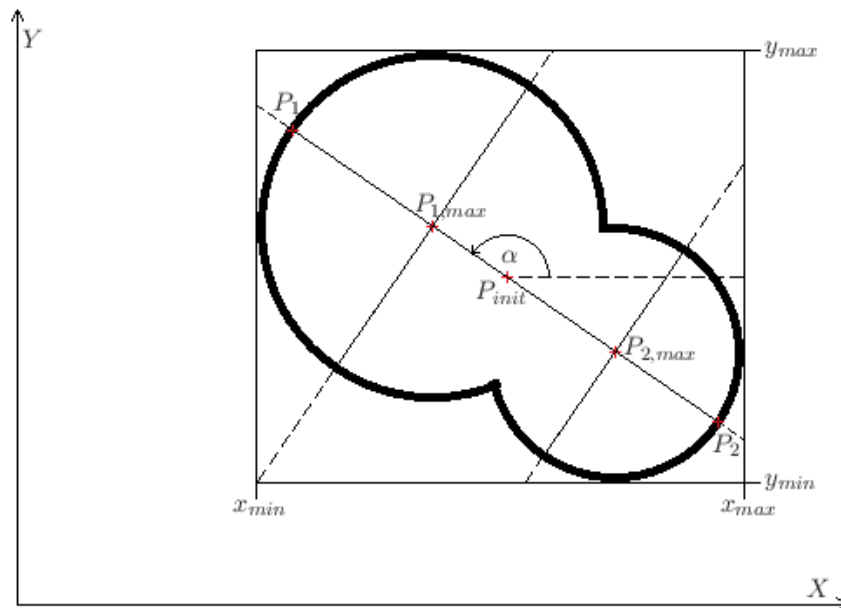
Rysunek 3.28: Dwa rodzaje scenariuszy

Obiekty celowo nie zostały ponumerowane, ponieważ możliwe są 4 sposoby numeracji dla każdego rodzaju scenariusza, zgodnie ze sposobem numeracji wskaźnika.



Rysunek 3.29: Przykład nakładania się obiektów

Celem algorytmu, jest uzyskanie informacji o położeniu obiektu 2D mającego większą średnicę, zatem przedstawiającego obiekt 3D znajdujący się bliżej kamery.



Rysunek 3.30: Analiza obiektu wejściowego

Powyżej został przedstawiony sposób analizy pary nakładających się na siebie obiektów w celu wyłonienia obiektu o większej średnicy. Punktem P_{init} jest punkt odpowiadający czerwonemu punktowi na pierwszej z ilustracji rysunku 3.29. Punktem wynikowym jest punkt zaznaczony na żółto na drugiej ilustracji tego samego rysunku.

Opis algorytmu:

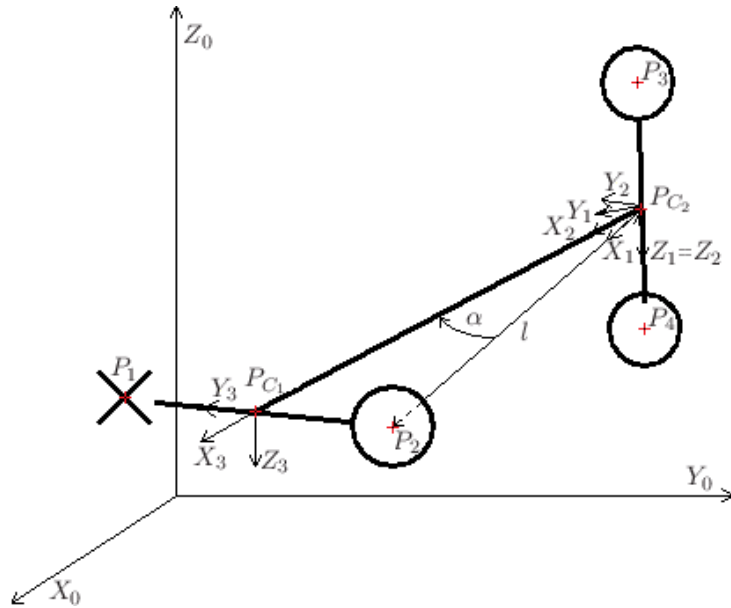
- Znalezienie w przedziale $0 \div 180^\circ$ kąta α będącego medianą kątów nachylenia odcinków o długości powyżej zadanego progu th . Odcinki te przechodzą przez punkt P_{init} oraz łączą przeciwległe krawędzie obiektu wejściowego. Proóg wynosił tutaj $th = 0,7d_{max}$, gdzie d_{max} jest największą znaną długością odcinka.
- Wyznaczenie na długości odcinka P_1P_{init} najdłuższego prostopadłego odcinka oraz na długości odcinka $P_{init}P_2$ wyznaczenie drugiego takiego odcinka.
- Rozwiązaniem przybliżonym jest punkt $P_{1,max}$ lub $P_{2,max}$ (w tym przypadku $P_{1,max}$). Ponieważ w rzeczywistości kształt obiektu wejściowego nie jest tak regularny, pozycja punktu będącego rozwiązaniem przybliżonym jest korygowana. Punkt wyznaczany jest na nowo w odległości $\frac{d_p}{2}$ od punktu P_1 na prostej P_1P_2 , gdzie d_p jest odcinkiem prostopadłym do prostej P_1P_2 , odpowiadającym średnicy większego obiektu.

Algorytm dostarcza informacji o pozycji tylko jednego, większego, obiektu, ponieważ precyzja z jaką określa się pozycję mniejszego obiektu jest niewystarczająca. Czasami zdarza się sytuacja, w której rozmiary obiektów są do siebie bardzo zbliżone (np. kiedy wskaźnik jest w dużej odległości od układu kamer). Wówczas w przypadku uzyskania rozwiązania niespełniającego kryteriów algorytmu selekcyjnego (jeszcze nie omawiany), w pierwszej kolejności sprawdzana jest konfiguracja z drugim, skorygowanym, obiektem z pary obiektów przysłaniający się - w tym przypadku byłby to obiekt o środku w punkcie $P_{2,max}$.

3. Obliczanie pozycji oraz orientacji wskaźnika

W tej części rozważań będą stosowane wektory $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_{c1}, \mathbf{p}_{c2}$ zaczepione w początku układu współrzędnych $(0, 0, 0)$ oraz o końcach odpowiednio w punktach $P_1, P_2, P_3, P_4, P_{C1}, P_{C2}$. Poniżej przedstawiono cztery przypadki, w których algorytm jest zdolny wyliczyć pozycję oraz orientację wskaźnika:

(a) Niewidoczny 1. obiekt.



Rysunek 3.31: Wskaźnik z zasłoniętym 1. obiektem

Na początku obliczany jest punkt znajdujący się w połowie odcinka P_3P_4 .

$$\mathbf{p}_{c2} = \frac{\mathbf{p}_3 + \mathbf{p}_4}{2} \quad (3.34)$$

Następnie wyznaczane są wersory pierwszego z układów współrzędnych $X_1Y_1Z_1$ względem układu bazowego $X_0Y_0Z_0$.

$$\mathbf{x}_1 = \mathbf{p}_2 - \mathbf{p}_{c2}, \quad \mathbf{z}_1 = \mathbf{p}_4 - \mathbf{p}_{c2}, \quad \mathbf{y}_1 = \mathbf{z}_1 \times \mathbf{x}_1 \quad (3.35)$$

$$\mathbf{x}_1 := \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}, \quad \mathbf{z}_1 := \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|}, \quad \mathbf{y}_1 := \frac{\mathbf{y}_1}{\|\mathbf{y}_1\|} \quad (3.36)$$

gdzie

$$\|\mathbf{x}_1\| = \sqrt{x_{1x}^2 + x_{1y}^2 + x_{1z}^2}, \quad \|\mathbf{z}_1\| = \sqrt{z_{1x}^2 + z_{1y}^2 + z_{1z}^2}, \quad \|\mathbf{y}_1\| = \sqrt{y_{1x}^2 + y_{1y}^2 + y_{1z}^2} \quad (3.37)$$

Kolejnym etapem są transformacje układów współrzędnych wzdłuż szkieletu wskaźnika, w celu wyznaczenia jego pozycji oraz orientacji, a także wyliczenia pozycji czwartego, brakującego punktu.

$$\mathbf{T}_0^1 = \begin{bmatrix} x_{1x} & y_{1x} & z_{1x} & p_{c2x} \\ x_{1y} & y_{1y} & z_{1y} & p_{c2y} \\ x_{1z} & y_{1z} & z_{1z} & p_{c2z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

$$\mathbf{T}_1^2 = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.39)$$

$$\mathbf{T}_2^3 = \begin{bmatrix} 1 & 0 & 0 & \cos(\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.40)$$

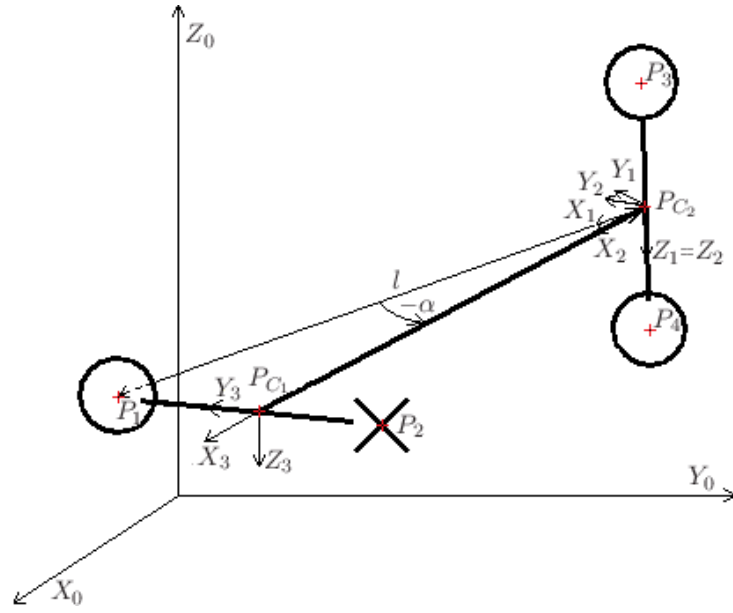
$$\mathbf{T}_0^2 = \mathbf{T}_0^1 \mathbf{T}_1^2 \quad (3.41)$$

$$\mathbf{T}_0^3 = \mathbf{T}_0^2 \mathbf{T}_2^3 \quad (3.42)$$

Na koniec obliczany jest brakujący punkt:

$$\mathbf{p}_1 = 2\mathbf{p}_{c_1} - \mathbf{p}_2, \quad \text{gdzie } \mathbf{p}_{c_1} \text{ jest 4. kolumną macierzy } \mathbf{T}_0^3 \quad (3.43)$$

(b) Niewidoczny 2. obiekt.



Rysunek 3.32: Wskaźnik z zasłoniętym 2. obiektem

$$\mathbf{p}_{c2} = \frac{\mathbf{p}_3 + \mathbf{p}_4}{2} \quad (3.44)$$

$$\mathbf{x}_1 = \mathbf{p}_1 - \mathbf{p}_{c2}, \quad \mathbf{z}_1 = \mathbf{p}_4 - \mathbf{p}_{c2}, \quad \mathbf{y}_1 = \mathbf{z}_1 \times \mathbf{x}_1 \quad (3.45)$$

$$\mathbf{x}_1 := \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}, \quad \mathbf{z}_1 := \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|}, \quad \mathbf{y}_1 := \frac{\mathbf{y}_1}{\|\mathbf{y}_1\|} \quad (3.46)$$

$$\mathbf{T}_0^1 = \begin{bmatrix} x_{1x} & y_{1x} & z_{1x} & p_{c2x} \\ x_{1y} & y_{1y} & z_{1y} & p_{c2y} \\ x_{1z} & y_{1z} & z_{1z} & p_{c2z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.47)$$

$$\mathbf{T}_1^2 = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) & 0 & 0 \\ \sin(-\alpha) & \cos(-\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

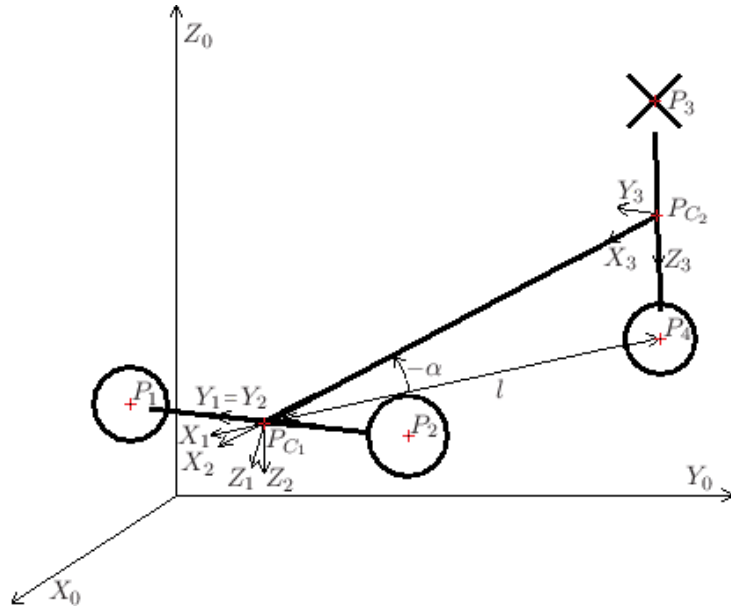
$$\mathbf{T}_2^3 = \begin{bmatrix} 1 & 0 & 0 & \cos(-\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cos(\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.49)$$

$$\mathbf{T}_0^2 = \mathbf{T}_0^1 \mathbf{T}_1^2 \quad (3.50)$$

$$\mathbf{T}_0^3 = \mathbf{T}_0^2 \mathbf{T}_2^3 \quad (3.51)$$

$$\mathbf{p}_2 = 2\mathbf{p}_{c1} - \mathbf{p}_1, \quad \text{gdzie } \mathbf{p}_{c1} \text{ jest 4. kolumną macierzy } \mathbf{T}_0^3 \quad (3.52)$$

(c) Niewidoczny 3. obiekt.



Rysunek 3.33: Wskaźnik z zasłoniętym 3. obiektem

$$\mathbf{p}_{c1} = \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \quad (3.53)$$

$$\mathbf{x}_1 = \mathbf{p}_{c1} - \mathbf{p}_4, \quad \mathbf{y}_1 = \mathbf{p}_1 - \mathbf{p}_{c1}, \quad \mathbf{z}_1 = \mathbf{x}_1 \times \mathbf{y}_1 \quad (3.54)$$

$$\mathbf{x}_1 := \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}, \quad \mathbf{y}_1 := \frac{\mathbf{y}_1}{\|\mathbf{y}_1\|}, \quad \mathbf{z}_1 := \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|} \quad (3.55)$$

$$\mathbf{T}_0^1 = \begin{bmatrix} x_{1x} & y_{1x} & z_{1x} & p_{c1x} \\ x_{1y} & y_{1y} & z_{1y} & p_{c1y} \\ x_{1z} & y_{1z} & z_{1z} & p_{c1z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.56)$$

$$\mathbf{T}_1^2 = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.57)$$

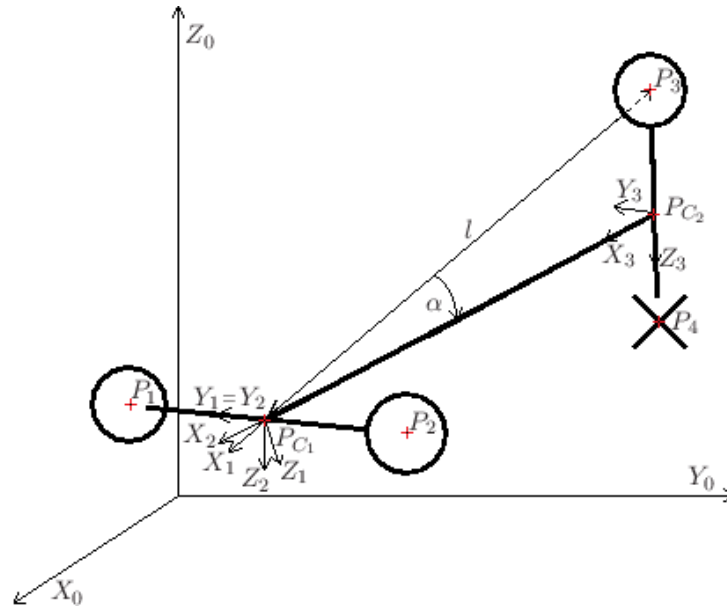
$$\mathbf{T}_2^3 = \begin{bmatrix} 1 & 0 & 0 & -\cos(-\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -\cos(\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.58)$$

$$\mathbf{T}_0^2 = \mathbf{T}_0^1 \mathbf{T}_1^2 \quad (3.59)$$

$$\mathbf{T}_0^3 = \mathbf{T}_0^2 \mathbf{T}_2^3 \quad (3.60)$$

$$\mathbf{p}_3 = 2\mathbf{p}_{c2} - \mathbf{p}_4, \quad \text{gdzie } \mathbf{p}_{c2} \text{ jest 4. kolumną macierzy } \mathbf{T}_0^3 \quad (3.61)$$

(d) Niewidoczny 4. obiekt.



Rysunek 3.34: Wskaźnik z zasłoniętym 4. obiektem

$$\mathbf{p}_{c_1} = \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \quad (3.62)$$

$$\mathbf{x}_1 = \mathbf{p}_{c_1} - \mathbf{p}_3, \quad \mathbf{y}_1 = \mathbf{p}_1 - \mathbf{p}_{c_1}, \quad \mathbf{z}_1 = \mathbf{x}_1 \times \mathbf{y}_1 \quad (3.63)$$

$$\mathbf{x}_1 := \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}, \quad \mathbf{y}_1 := \frac{\mathbf{y}_1}{\|\mathbf{y}_1\|}, \quad \mathbf{z}_1 := \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|} \quad (3.64)$$

$$\mathbf{T}_0^1 = \begin{bmatrix} x_{1x} & y_{1x} & z_{1x} & p_{c_1x} \\ x_{1y} & y_{1y} & z_{1y} & p_{c_1y} \\ x_{1z} & y_{1z} & z_{1z} & p_{c_1z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.65)$$

$$\mathbf{T}_1^2 = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.66)$$

$$\mathbf{T}_2^3 = \begin{bmatrix} 1 & 0 & 0 & -\cos(\alpha)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.67)$$

$$\mathbf{T}_0^2 = \mathbf{T}_0^1 \mathbf{T}_1^2 \quad (3.68)$$

$$\mathbf{T}_0^3 = \mathbf{T}_0^2 \mathbf{T}_2^3 \quad (3.69)$$

$$\mathbf{p}_4 = 2\mathbf{p}_{c_2} - \mathbf{p}_3, \quad \text{gdzie } \mathbf{p}_{c_2} \text{ jest 4. kolumną macierzy } \mathbf{T}_0^3 \quad (3.70)$$

Do wyznaczenia orientacji ponownie zostanie wykorzystana kinematyka odwrotna dla kątów RPY (opisana w rozdz. 2.2.2 poświęconym wirtualnym manipulatorom). Wcześniej jednak, należy znaleźć wersory określające orientację wskaźnika - \mathbf{n} , \mathbf{o} , \mathbf{a} .

W przypadku 3a i 3b orientację oraz pozycję wskaźnika można uzyskać z macierzy \mathbf{T}_0^3 :

- 1. kolumna macierzy \mathbf{T}_0^3 - \mathbf{n}
- 2. kolumna macierzy \mathbf{T}_0^3 - \mathbf{o}
- 3. kolumna macierzy \mathbf{T}_0^3 - \mathbf{a}
- 4. kolumna macierzy \mathbf{T}_0^3 - pozycja

W pozostałych przypadkach, tj. 3c i 3d orientację oraz pozycję wskaźnika można uzyskać z macierzy \mathbf{T}_0^2 :

- 1. kolumna macierzy \mathbf{T}_0^2 - \mathbf{n}
- 2. kolumna macierzy \mathbf{T}_0^2 - \mathbf{o}
- 3. kolumna macierzy \mathbf{T}_0^2 - \mathbf{a}
- 4. kolumna macierzy \mathbf{T}_0^2 - pozycja

Na koniec obliczane są kąty RPY:

$$\alpha_Y = \operatorname{atan2}\left(\frac{n_y}{n_x}\right) \quad (3.71)$$

$$\alpha_P = \operatorname{atan2}\left(\frac{-n_z}{n_x C_Y + n_y S_Y}\right) \quad (3.72)$$

$$\alpha_R = \operatorname{atan2}\left(\frac{a_x S_Y - a_y C_Y}{o_y C_Y - o_x S_Y}\right) \quad (3.73)$$

W przypadku wykrycia 4 obiektów, do obliczeń brane są po uwagę 3 pierwsze. Wówczas pozycja 4. obiektu powinna być wystarczająco bliska pozycji wyliczonej przez algorytm. W przeciwnym wypadku system przejdzie w stan *Ograniczone sterowanie*. Taka sytuacja może wystąpić przy zbyt dużej odległości wskaźnika od układu kamer.

4. Algorytm selekcyjny rodziny rozwiązań

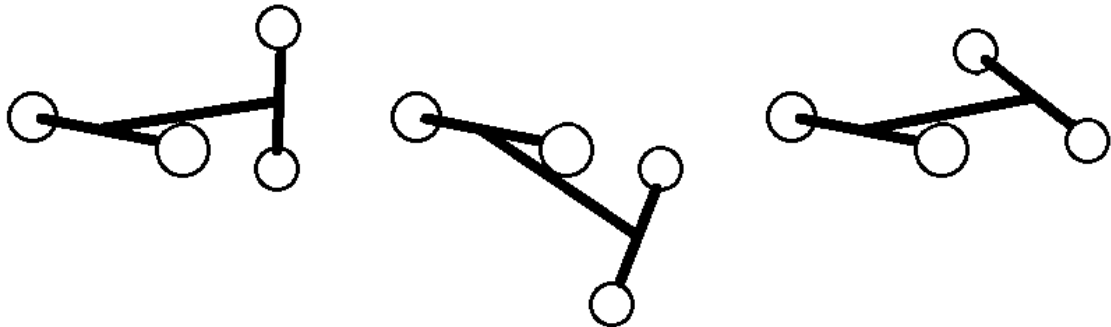
Algorytm ten selekcyjnie wybiera rodzinę rozwiązań, ostatecznie dochodząc do unikalnego rozwiązania (jeśli istnieje). W praktyce, zdarza się co jakiś czas, że obiekty zostaną źle ponumerowane. Wówczas sprawdzane są wszystkie 24 permutacje dla możliwych połączeń obiektów w pary oraz wszystkie 24 permutacje dla możliwych sposobów indeksacji obiektów wskaźnika. Ostatecznie sprawdzanych jest $24^2 = 576$ rozwiązań, spośród których wybierane jest to, które spełnia najlepiej kryteria oraz spełnia pewne kryteria minimalne (ponieważ zdarza się, że rozwiązanie nie istnieje z powodu niepełnej informacji wejściowej). Mimo tak dużego zbioru, który jest przeszukiwany, jest to operacja praktycznie nieodczuwalna dla użytkownika, ponieważ wykonywana jest tylko w jednej iteracji. W kolejnych iteracjach, algorytm śledzący kontynuuje swoje zadanie (tj. nadaje indeksy obiektom).

Zastosowano tutaj kolejne wyspecjalizowane algorytmy, w celu stopniowego ograniczania rodziny rozwiązań, aż do momentu otrzymania unikalnego rozwiązania.

(a) *Algorytm kontrolujący wymiary wskaźnika*

Algorytm porównuje wymiary wirtualnego wskaźnika (wskaźnik widziany przez system) z wymiarami rzeczywistego wskaźnika. Badane są długości odcinków P_1P_2 , P_3P_4 oraz $P_{C_1}P_{C_2}$ (zgodnie z oznaczeniami z punktu 3). Dopuszczalna tolerancja wynosi $\frac{d}{10}$, gdzie d jest długością badanego odcinka odpowiadającego odcinkowi rzeczywistego wskaźnika.

Podobnie jak w punkcie 2 obiekty celowo nie zostały ponumerowane, ponieważ możliwe są 4 sposoby numeracji dla każdego scenariusza, zgodnie ze sposobem numeracji wskaźnika.

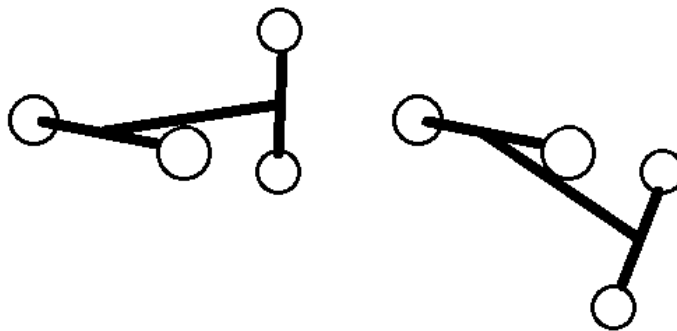


Rysunek 3.35: Rodzina rozwiązań

(b) *Algorytm kontrolujący ortogonalność odcinków P_1P_2 oraz P_3P_4*

Algorytm odrzuca rozwiązania nie spełniające warunku ortogonalności [12][11], przy tolerancji tol

$$|P_1\vec{P}_2 \cdot P_3\vec{P}_4| < tol \quad (3.74)$$



Rysunek 3.36: Rodzina rozwiązań

(c) *Algorytm kontrolujący orientację wskaźnika*

Ostatnim etapem selekcji jest odrzucenie rozwiązań, w których orientacja wskaźnika odbiega za bardzo od orientacji przewidywanej w poprzedniej iteracji. Warunek ten ma postać

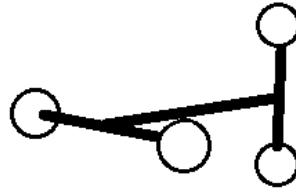
$$\max(|R_{curr} - R_{pred}|, |P_{curr} - P_{pred}|, |Y_{curr} - Y_{pred}|) < tol \quad (3.75)$$

gdzie

$(R_{curr}, P_{curr}, Y_{curr})$ - orientacja badanego rozwiązania

$(R_{pred}, P_{pred}, Y_{pred})$ - orientacja przewidywana w poprzedniej iteracji

tol - tolerancja

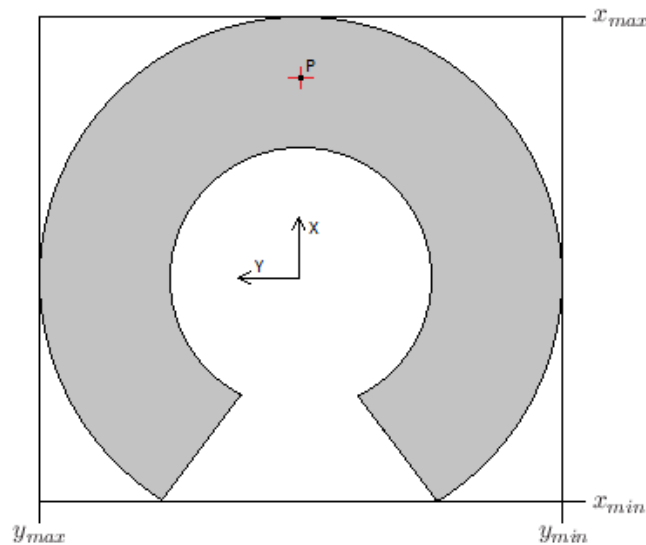


Rysunek 3.37: Rozwiązanie

Jeśli na pewnym etapie analizy okaże się, że wszystkie rozwiązania zostały odrzucone, wówczas system przechodzi do stanu *Ograniczone sterowanie*.

5. Skalowanie

Końcowym etapem obliczeń w systemie wizyjnym jest skalowanie, w celu zamiany pozycji wskaźnika na pozycję zadaną manipulatorowi.



Rysunek 3.38: Rzut prostopadły na osiągalną toroidalną przestrzeń roboczą manipulatora

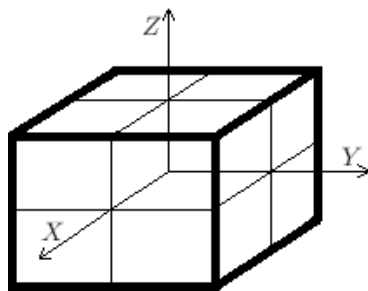
Powyżej przedstawiono osiągalną przestrzeń roboczą manipulatora, oraz przykładowy punkt P , będący pozycją końcówki roboczej manipulatora.

	Minimum	Maximum		Minimum	Maximum
x	-185	246	x	-1500	2000
y	-175	175	y	-1500	1500
z	-185	185	z	-1500	1500

Rysunek 3.39: Ograniczenia przestrzeni roboczych wskaźnika oraz manipulatora - [mm]

Aby dokonać skalowania, należało wyznaczyć arbitralnie taki prostopadłościan, aby przestrzeń robocza całkowicie się w nim zawierała. Innymi słowy, należało wyznaczyć ograniczenia $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$.

Do celów testowych ustawiono ograniczenia z pewną rezerwą, w celu obserwacji zachowania się manipulatora w sytuacji gdy zadana pozycja wykracza poza jego przestrzeń roboczą. Wynika z tego, że rozdzielczość układu może być jeszcze nieco lepsza niż obecnie (dalej przy tych samych rozdzielczościach kamer - 640x480). Przy obecnych ustawieniach rozdzielczość szacowana jest na 5 mm (rozdzielczości dla współrzędnych x, y, z różnią się nieznacznie między sobą).



Rysunek 3.40: Przestrzeń robocza

Powyższy rysunek przedstawia kształt przestrzeni roboczej wskaźnika oraz manipulatora (pamiętając równocześnie o wartościach ograniczeń - rys. 3.39)

W systemie występują trzy rodzaje offsetów:

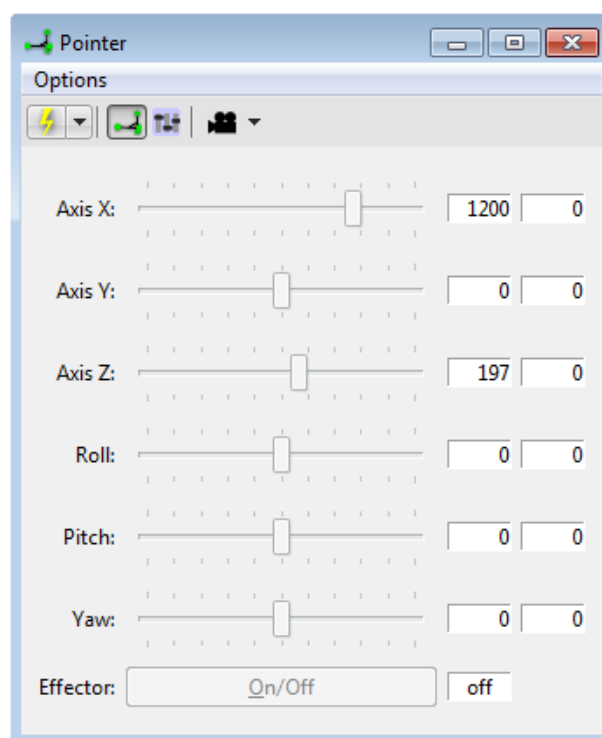
- (a) offset punktu początkowego w przestrzeni roboczej wskaźnika - P_{C_1}
- (b) offset punktu początkowego w przestrzeni roboczej manipulatora - P_{init}
- (c) offset punktu centralnego wskaźnika wirtualnego (P_C), dzięki któremu, wirtualny wskaźnik cały czas znajduje się w tym samym miejscu, a widoczna jest jedynie zmiana jego orientacji. Offset ten jest wykorzystywany jedynie do celu wizualizacji.

Układy współrzędnych wraz z offsetami 5a oraz 5b definiują przestrzenie robocze wskaźnika oraz manipulatora. W obydwu przestrzeniach roboczych współrzędne punktów początkowych są w takich samych proporcjach w stosunku do ograniczeń. Umożliwia to optymalne wykorzystanie przestrzeni roboczych wskaźnika oraz manipulatora, a także pozwala na bezuderzeniowe przejście ze sterowania manualnego do sterowania za pośrednictwem systemu wizyjnego.

4. Opis programu

4.1. Graficzny interfejs użytkownika (GUI)

Podstawowym narzędziem służącym do komunikowania się z systemem jest graficzny interfejs użytkownika - GUI (*Graphical User Interface*). W skład głównego okna wchodzi: menu, pasek narzędzi oraz panel główny.



Rysunek 4.1: Graficzny interfejs użytkownika (GUI) - okno główne

Poniżej szczegółowy opis okna głównego GUI.

- Menu - *Opcje*:
 - *Preferencje* - otwiera okno umożliwiające ustawianie parametrów systemu
 - *Kalibracja* - uruchamia kalibrację systemu wizyjnego
- Pasek narzędzi:
 - *Połącz/Rozłącz* - zarządza połączeniem z manipulatorem, daje również możliwość wyboru między rzeczywistym a wirtualnym manipulatorem (rozwijane menu)
 - *System wizyjny* - przełącza system w tryb *Lokalizacja* (jeśli połączenie jest aktywne) lub uruchamia system wizyjny przy kolejnym wciśnięciu przycisku *Połącz/Rozłącz*
 - *Pulpit sterowniczy* - uruchamia sterowania ręczne
 - *Widok z kamery* - umożliwia zarządzanie oknami z podglądem obrazu z kamer

Pasek narzędzi umożliwia wybór w podglądaniu przetwarzanych obrazów a także na dostosowanie wyglądu ramki (np. wł./wył. warstwy z widokiem wirtualnego wskaźnika lub/i informacji o szybkości przetwarzania - FPS). Istnieje także możliwość nagrywania materiału wideo w formacie MOV oraz zapisywania pojedynczych klatek w formacie PNG.

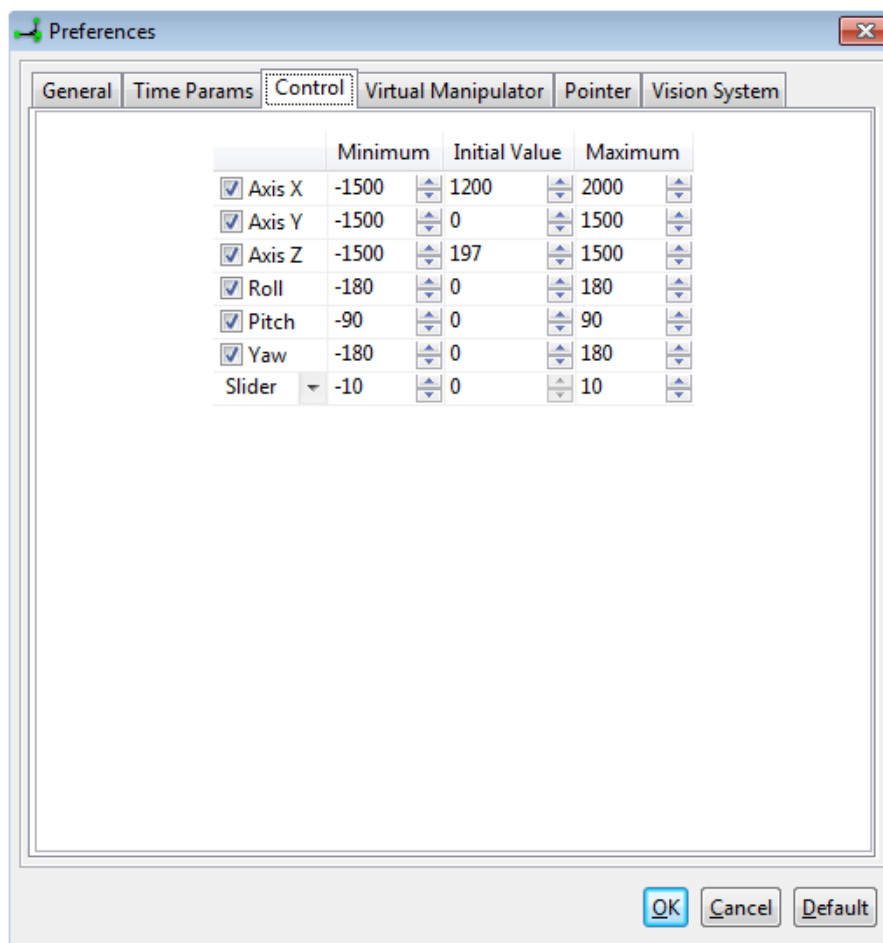
- *Panel główny* - osadzony jest w nim pulpit sterowniczy, który umożliwia zadawanie wartości manipulatorowi, oraz odczytywanie wartości procesowych. W razie wystąpienia niesprzyjających warunków oświetleniowych lub innej nieprzewidzianej sytuacji, jest to podstawowe narzędzie pozwalające komunikować się z manipulatorem.

Warto również wspomnieć, że program posiada pamięć, zatem zapamiętywane są takie zmienne jak np. poprzednio używany tryb komunikacji z robotem, poprzednia pozycja okna, lub też poprzednia zakładka w oknie *Preferencje*.

Kolejnym elementem GUI jest okno *Preferencje* w którym ustawiane są poszczególne parametry systemu. Poniżej opis zakładek:

- *Główne* - umożliwia ustawienie takich parametrów jak m.in. rozdzielczości kamer, numery kamer w systemie operacyjnym, łączenie z manipulatorem na starcie systemu.
- *Parametry czasowe* - m.in. częstości próbkowania w różnych elementach systemu, czas trwania stanu *Bezczynność*

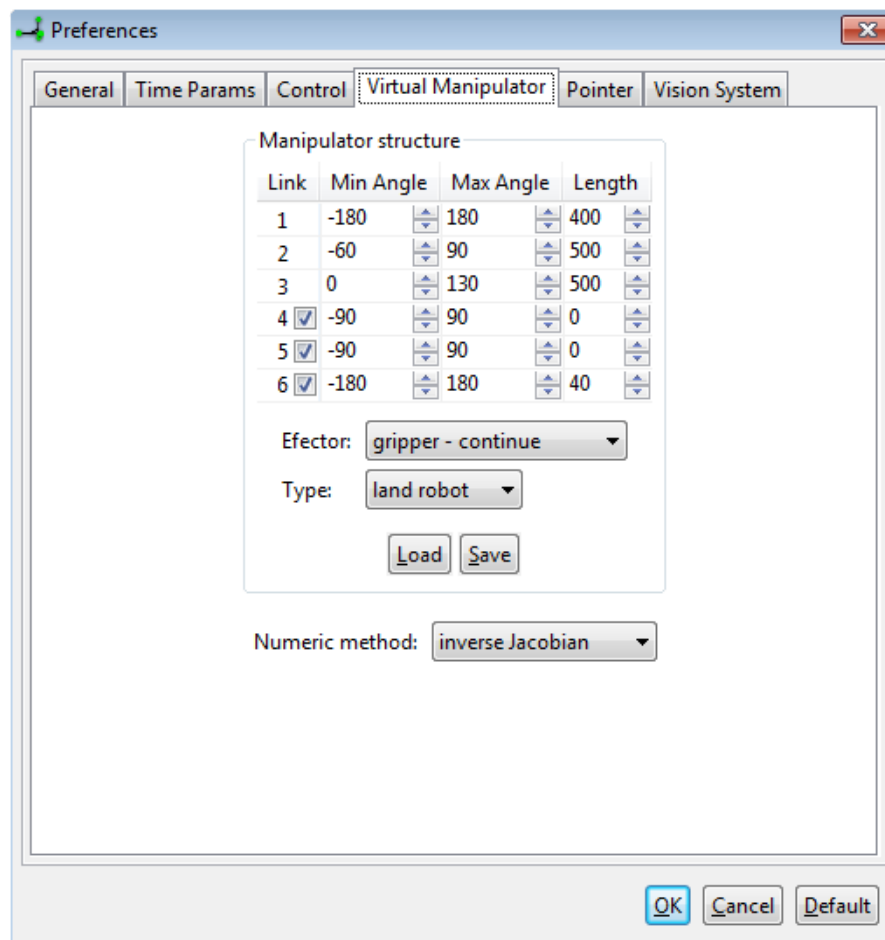
– Sterowanie



Rysunek 4.2: Zakładka *Sterowanie*

Pulpit sterowniczy jest customizowalny, zatem jest możliwość dostosowania go do danego typu manipulatora. Przykładowo, jeśli zachodzi potrzeba sterowania manipulatorem o 5 stopniach swobody - np manipulatorem bez możliwości sterowania odchyleniem - jest możliwość ograniczenia zestawu zmiennych. Podobnie, jeśli efektor jest sterowany 2-stanowo, wówczas na pulpicie sterowniczym powinien pojawić się toggle button, jeśli natomiast efektor jest sterowany 3-stanowo, lub w sposób ciągły, wtedy warto wybrać do tego celu suwak.

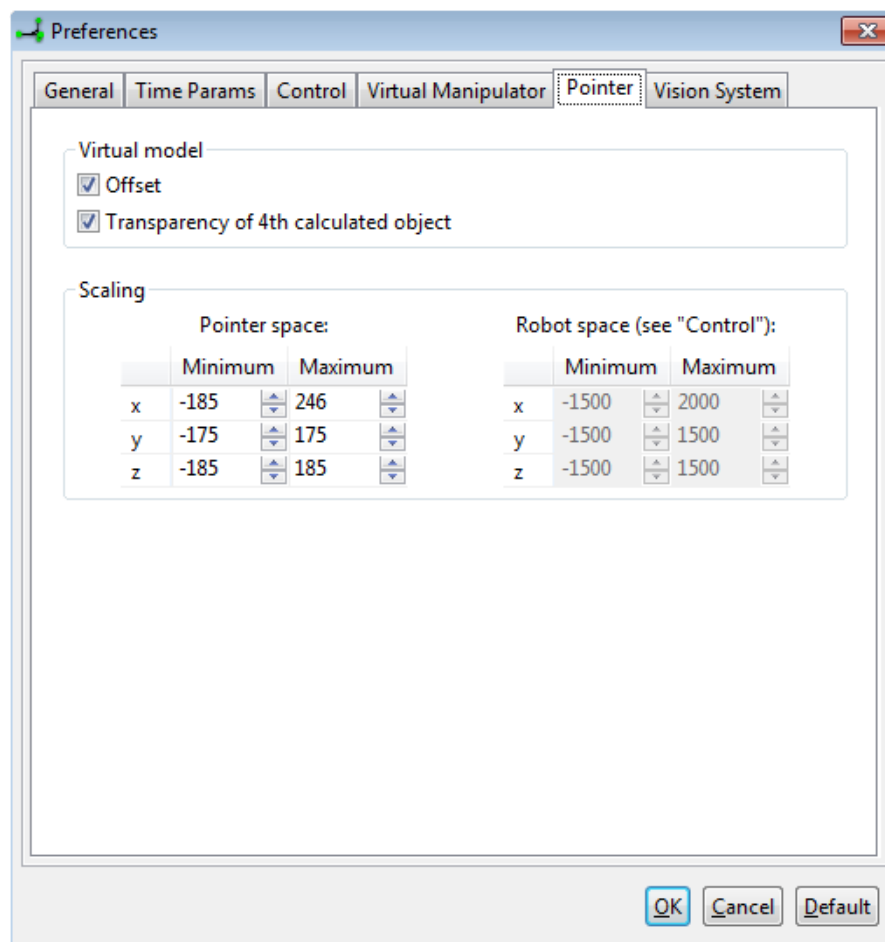
– Wirtualny manipulator robota

Rysunek 4.3: Zakładka *Wirtualny manipulator*

W tej zakładce jest możliwość utworzenia wirtualnego manipulatora robota terenowego lub podwodnego o dowolnych rozmiarach nieprzekraczających 10000 jednostek (odpowiadających milimetrom). Każdy manipulator można odpowiednio skonfigurować wybierając konkretny rodzaj efektora. Utworzony model można zapisać na dysku w pliku tekstowym INI. Tak zapisany model można w późniejszym czasie wczytać do systemu.

Dodatkowo jest możliwość wybrania metody numerycznej realizującej odwrotne zadanie kinematyki.

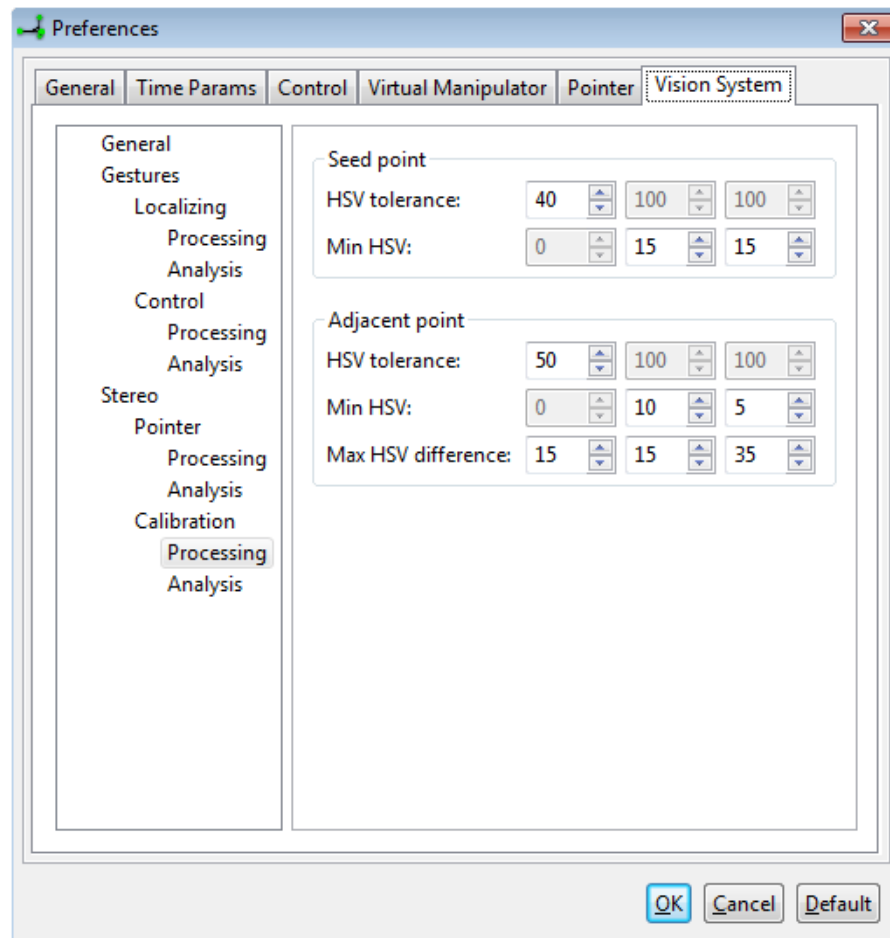
– Wskaźnik



Rysunek 4.4: Zakładka *Wskaźnik*

Powyższa zakładka umożliwia wyskalowanie podsystemu obliczającego pozycję dla manipulatora. Jest też możliwość ustawiania właściwości wirtualnego wskaźnika - np. dynamiczne ustawianie offsetu pozwalające na obserwowanie wirtualnego wskaźnika nawet przy dużych zmianach pozycji względem wartości początkowej.

– System wizyjny



Rysunek 4.5: Zakładka *System wizyjny*

Wszystkie parametry systemu wizyjnego, można ustawić przy pomocy powyższej zakładki. Istnieje również możliwość strojenia parametrów w czasie rzeczywistym, tj. w czasie pracy systemu wizyjnego.

Należy na koniec wspomnieć, że wszystkie ustawienia w oknie *Preferencje* mogą być przywrócone do ustawień domyślnych, poprzez użycie przycisku *Default*.

4.2. Skróty klawiaturowe

1. Podstawowe skróty:

[Alt+Enter] Ustawienia

[Ctrl+C] Kalibracja

[Shift+R] Rzeczywisty robot

[Shift+V] Wirtualny robot

[Alt+Up; Alt+X; Ctrl+Up] Połącz/Rozłącz

- [**Alt+Left; Ctrl+Left**] System wizyjny
- [**Alt+Right; Ctrl+Right**] Tryb manualny
- [**Alt+Down; Ctrl+Down**] Podgląd z kamer (menu)
- [**Ctrl+1**] Kamery stereo
- [**Ctrl+2**] Kamera rejestrująca gesty
- [**Ctrl+3**] Kamera robota
- [**Ctrl+P**] Podgląd wskaźnika
- [**Ctrl+F**] Podgląd FPS
- [**Ctrl+R**] Nagrywanie
- [**Ctrl+A**] Stop klatka

2. Podgląd obrazu z kamer:

- [**Shift+(0 ÷ 9)**] Obraz/obrazy nr (0 ÷ 9) - jeśli np. jest żądanie 6. obrazu a dostępnych jest tylko 5, to wyświetlany jest ten ostatni obraz.
- [**, / .**] Obraz z lewej/prawej kamery stereo

3. Wirtualny manipulator robota:

- [**Alt+Shift+Left**] Obrót w lewo
- [**Alt+Shift+Right**] Obrót w prawo
- [**Alt+Shift+Up**] Obrót w górę
- [**Alt+Shift+Down**] Obrót w dół
- [**=**] Przybliżenie (1x)
- [**-**] Oddalenie (1x)
- [**Ctrl+=**] Przybliżenie (100x)
- [**Ctrl+-**] Oddalenie (100x)

4. Wskaźnik:

- [**PgUp**] Przybliżenie (1x)
- [**PgDown**] Oddalenie (1x)
- [**Ctrl+PgUp**] Przybliżenie (100x)
- [**Ctrl+PgDown**] Oddalenie (100x)

5. Podsumowanie

Do podstawowych założeń pracy magisterskiej należało: zaprojektowanie, wykonanie, przetestowanie oraz zaprezentowanie systemu wizyjnego, przy użyciu którego można sterować manipulatorem robota. Zastosowanie wirtualnych manipulatorów znacznie uprościło proces testowania systemu wizyjnego. Dodatkowo, napisano moduł umożliwiający komunikację z rzeczywistym manipulatorem, a następnie przetestowano go przy udziale prostych symulatorów manipulatorów napisanych w C++ oraz w MATLABie.

Podstawowym czynnikiem określającym jakość systemu była szybkość przetwarzania obrazu z 3 kamer równocześnie. Aby była jak największa, zastosowano przetwarzanie współbieżne. Czynnikiem ograniczającym była moc dwóch rdzeni CPU (2GHz). Mimo to, udało się uzyskać satysfakcjonującą szybkość przetwarzania 9-10 FPS przy rozdzielczościach kamer 640x480, w trybie nie przeznaczonym do prezentacji (tj. bez skalowania obrazów oraz nagrywania materiału wideo).

Bibliografia

- [1] C. Bołdak. *Cyfrowe Przetwarzanie Obrazów*. Białystok University of Technology, Institute of Computer Science, 2008. <http://aragorn.pb.bialystok.pl/~boldak/DIP/Wyklady.html>.
- [2] Z. Bublński. *Systemy wizyjne i głosowe*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków, 2009. wykłady.
- [3] S. R. Buss. *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods*. Department of Mathematics University of California, San Diego La Jolla, CA 92093-0112, 2009. <http://math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>.
- [4] J. J. Craig. *Wprowadzenie do robotyki, mechanika i sterowanie*. Wydawnictwa Naukowo-Techniczne, Warszawa 1993, 1995. tłumaczenie.
- [5] B. Eckel. *Thinking in Java*. Pearson Education, Inc. Publishing as Prentice Hall PTR Upper Saddle River, NJ 07458, 2003.
- [6] J. Grębosz. *Symfonia C++ Standard. Tom 1-2*. Wydawnictwo Edition 2000, Kraków, 2008.
- [7] M. Kelly. *Unified Modelling Language (UML) and Use Case Diagrams (UCDs)*. McKinnon Secondary College. <http://www.vceit.com/SD/usecase/uml.htm>.
- [8] A. Maldonado, A. Fedrizzi. *Controlling Robot Manipulators*. Technische Universitat Munchen, 2008.
- [9] M. Dajda, G. Łojek. *Zastosowanie stereowizji do odtwarzania trajektorii obiektów w przestrzeni 3D*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, 2010. praca magisterska.
- [10] S. Fuksa, W. Byrski. *Czteropunktowa metoda identyfikacji transformacji stereowizyjnej*. Automatyka, Tom 9, Zeszyt 3., Krakow, 2005.
- [11] T. Jurlewicz, Z. Skoczylas. *Algebra liniowa 2*. Oficyna Wydawnicza GIS, Wrocław, 2005. przykłady i zadania.

- [12] T. Jurlewicz, Z. Skoczylas. *Algebra liniowa 1*. Oficyna Wydawnicza GIS, Wrocław, 2006. przykłady i zadania.
- [13] S. Myśliński. *Rozpoznawanie obrazów dłoni za pomocą gramatyk klasy ETPL(k) w systemach wizyjnych analizy języka migowego*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków, 2009. praca doktorska.
- [14] Oracle. *The Java Tutorials - Concurrency*. <http://docs.oracle.com/javase/tutorial/essential/concurrency/procthread.html>.
- [15] A. Ratnaweera. *Robot Kinematics for Orientation*. Department of Mechanical Engineering, University of Peradeniya, 2007. <http://www.pdn.ac.lk/eng/old/mechanical/menu/class/downloads/notes/Kinematics2.pdf>.
- [16] P. Rotter. *Transformacje kinematyczne i wizyjne w robotyce*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków, 2009. wykłady.
- [17] Sparx Systems. *UML Tutorial - UML 2 Use Case Diagram*. http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_usecasediagram.html.
- [18] Wortal Robotyka.com. *Notacja Denavita-Hartenberga*. <http://www.robotyka.com/teoria.php/teoria.47>.
- [19] M. Zaczyk. *Podstawy robotyki*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków, 2008. wykłady.
- [20] M. Zaczyk. *Konstrukcja i programowanie robotów*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków, 2009. wykłady.