

PROJEKT

PROJEKT ZESPOŁOWY

Doręczenie II

Krystian Cyga, 259247
Paweł Łyszczaż, 259258
Mateusz Strembicki, 259263

Prowadzący:

**dr hab. inż. Elżbieta
Roszkowska**

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki, Fotoniki i
Mikrosystemów

Politechniki Wrocławskiej



7 października 2023

Spis treści

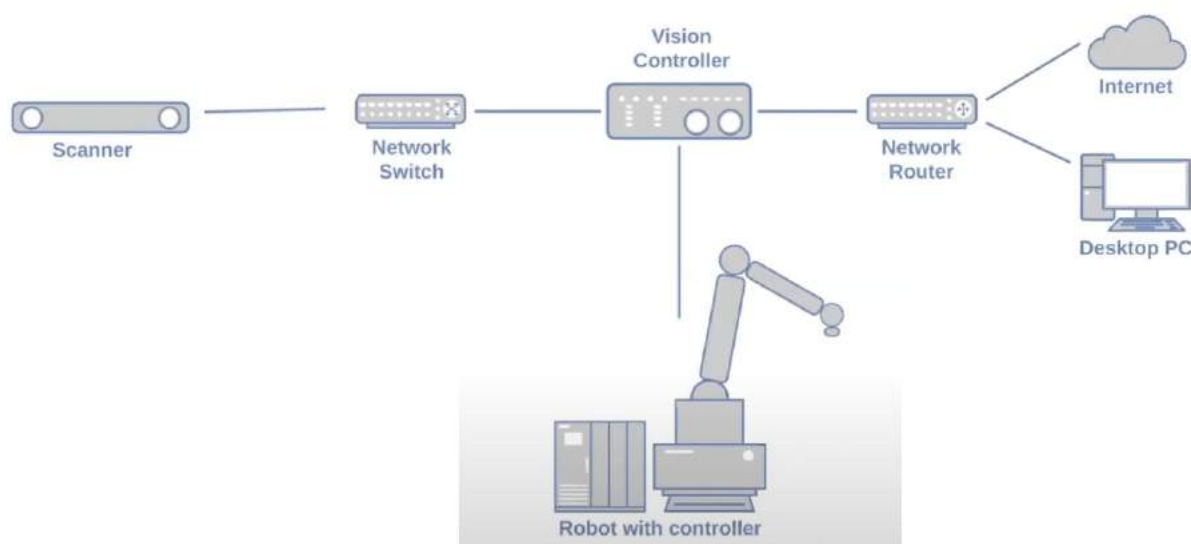
1	Zadania realizowane z firmą	1
1.1	Podłączenie oraz konfiguracja systemu wizyjnego	1
1.2	Połączenie robota do systemu wizyjnego, jego kalibracja oraz strefy roboczej robota	1
1.3	Podłączenie robota do sterownika	1
1.4	Dodanie komponentów robota do aplikacji	2
1.4.1	Kalibracja	5
1.4.2	Strefy pracy robota	7
1.5	Symulacja	8
2	Zadania realizowane w lab. 010	11
2.1	Realizowane zadania	11
2.2	Kalibracja systemu wizyjnego <i>Wrist Camera</i>	11
2.3	Montaż kamery	11
2.3.1	Oprogramowanie	11
2.4	Problem bin-picking'u	15
2.4.1	Opis	15
2.4.1.1	Założenia	15
2.4.1.2	Realizacja	15
2.4.1.3	Algorytm programu	16
2.4.2	Podsumowanie	17
2.5	Rozpoznawanie barw	18
2.5.1	Opis	18
2.5.2	Realizacja	18
2.5.2.1	Założenia	18
2.5.2.2	Dodanie możliwości rozpoznawania barw	19
2.5.2.3	Dodanie systemu paletyzacji w oparciu o znaleziony kształt	19
2.5.2.4	Przeprowadzenie testów	20
2.5.3	Podsumowanie	20
2.6	Czujnik siły	21
2.6.1	Opis	21
2.6.2	Realizacja	21
2.6.2.1	Włączenie czujnika	21
2.6.2.2	Odkładanie elementów	21
2.6.3	Podsumowanie	21
3	Porównanie	22
3.1	KPZ	23
3.1.1	Plusy	23
3.1.1.1	Głębia	23
3.1.1.2	Osie	23
3.1.2	Minusy	23
3.1.2.1	Barwy	23
3.1.2.2	Kalibracja	23
3.1.2.3	Redundantność	23

3.2	UR3	24
3.2.1	Plusy	24
3.2.1.1	Nauka	24
3.2.1.2	Różnorodność	24
3.2.2	Minusy	24
3.2.2.1	Głębia	24
4	Podsumowanie	24
5	Diagram UML	25

1 Zadania realizowane z firmą

1.1 Podłączenie oraz konfiguracja systemu wizyjnego

W celu poprawnego skonfigurowania pierwszego testowanego systemu do realizacji rozwiązania bin-picking należało odpowiednio połączyć i skomunikować kolejne elementy schematu.



Zdjęcie 1: Schemat połączeń systemu

W celu poprawnego działania całego systemu należało podłączyć do głównej jednostki kontrolnej całego systemu wizyjnego i operacyjnego monitor oraz peryferia pozwalające na kontrolę pulpitu oraz skaner wizyjny, wybierając odpowiedni rodzaj i model już na tym etapie możliwy był podgląd działania skanera. Oto przykład:

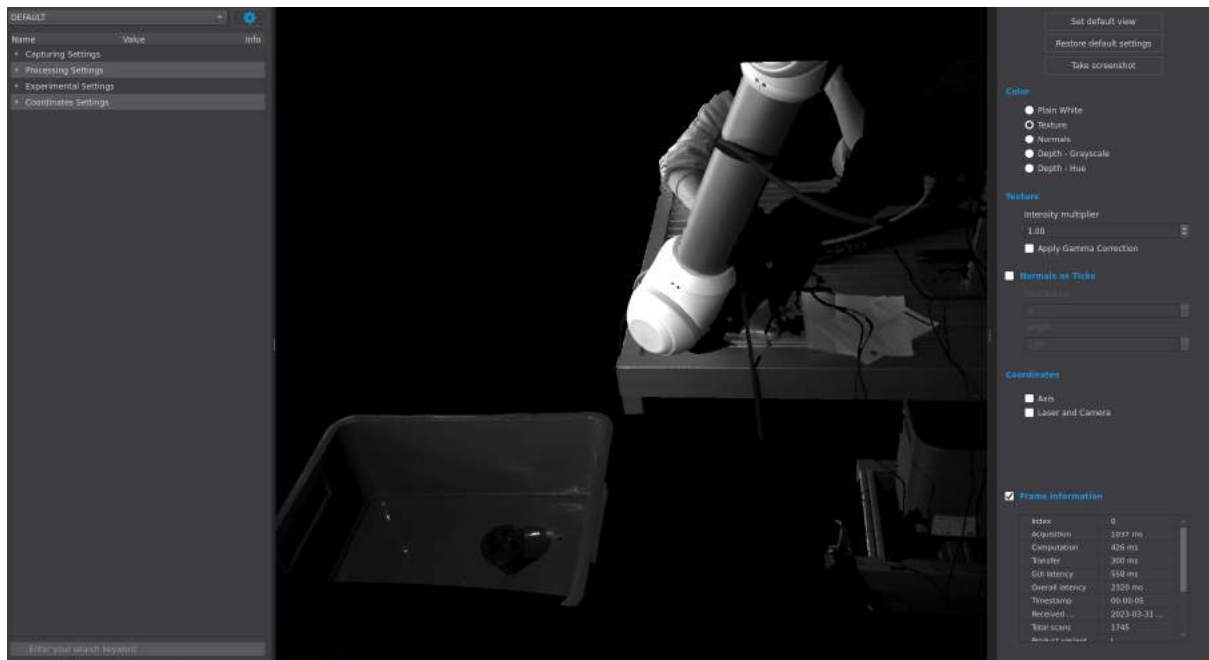
Następnym krokiem było podłączenie kontrolera robota to jednostki centralnej. W skutek tego, program jest w stanie w oparciu o symulacje przestrzenne poruszać robotem wedle zaplanowanych ścieżek do podnoszenia przedmiotów co zostanie przedstawione w późniejszych rozdziałach. Wszystkie z tych połączeń zostały zrealizowane poprzez kable typu RJ-45.

Po pomyślnym połączeniu wszystkich peryferiów, należało w aplikacji kontrolującej cały system ustawić odpowiednie adresy IP skanera oraz robota, a następnie w aplikacji dedykowanej konkretnej marce skanerów wizyjnych skonfigurować urządzenie i upewnić się że odpowiedni profil utworzony w aplikacji wizyjnej został wczytany w aplikacji odpowiedzialnej za cały system bin-picking, tak jak zostało ukazane poniżej.

1.2 Połączenie robota do systemu wizyjnego, jego kalibracja oraz strefy roboczej robota

1.3 Podłączenie robota do sterownika

Zgodnie z rysunkiem 1 robota oraz jego jednostkę centralną do kontrolera systemu wizyjnego poprzez przewód Ethernet. Jednostkę centralną robota należy również podłączyć do

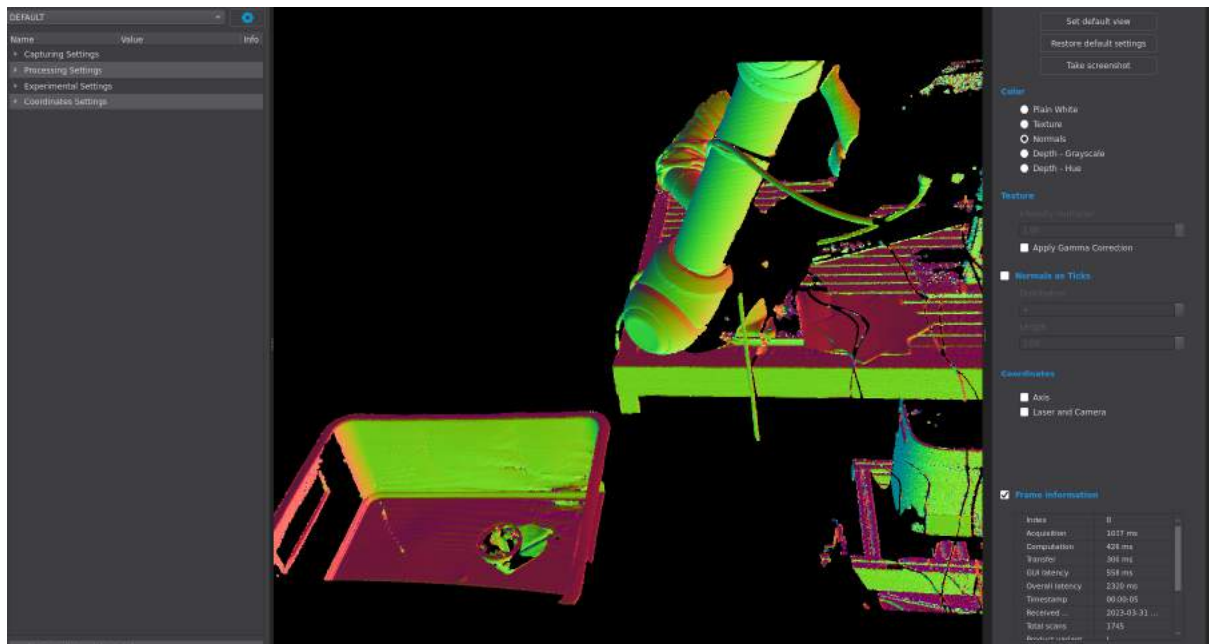


Zdjęcie 2: Przykład wizji skanera

zasilania sieciowego, a chwytak robota, o którym będzie mowa niżej, również należy podłączyć ze sterownikiem efektora, a ten natomiast do zasilania oraz jednostki centralnej robota.

1.4 Dodanie komponentów robota do aplikacji

Pierwszym krokiem jest dodanie modelu robota, z którego będziemy korzystać. W naszym wypadku będzie to robot firmy *Doosan*.




Zdjęcie 3: Przykład wizji skanera

Kolejnym krokiem jest wczytanie modelu 3D efektor. Ze względu na brak wiedzy o elemencie chwytak nie został zaprojektowany przez nas. Skorzystaliśmy z gotowego modelu chwytaka.

Na tym etapie ustawiamy TCP (Tool Center Point) narzędzia. Ustawianie TCP jest stosunkowo łatwe gdyż opiera się na odpowiednim przeciągnięciu układu współrzędnych odpowiedzialnych za ten parametr efektor.

Network

[Save](#)

 Robot interface

IPv4 address* ⓘ

Subnet mask <1; 32>* ⓘ


Action Request Server port <1024; 65535> ⓘ

Robot controller

IPv4 address* ⓘ

Robot State Server port <1024; 65535> ⓘ


[Test connection](#)

 Scanner interface

☐ Static IP

IPv4 address* ⓘ

Subnet mask <1; 32>* ⓘ

 Network interface

☒ DHCP client

IPv4 address ⓘ

Subnet mask <1; 32> ⓘ

Gateway ⓘ

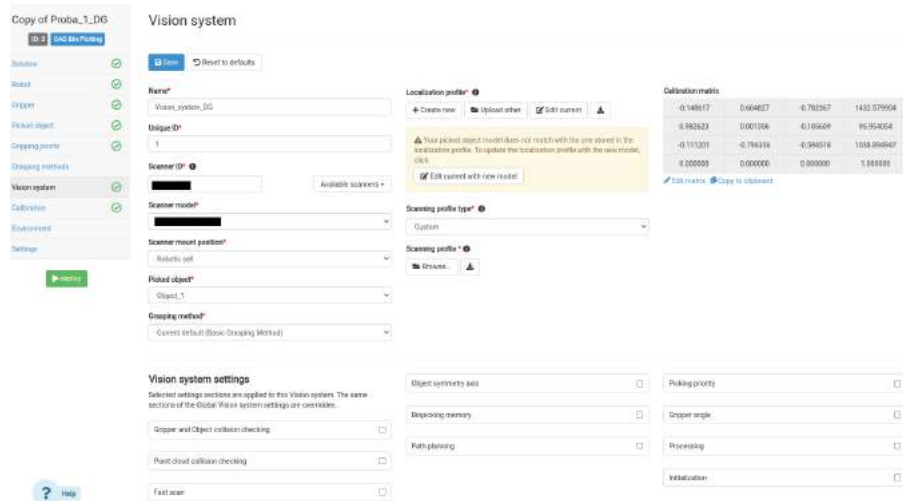
Primary DNS server ⓘ

Secondary DNS server ⓘ

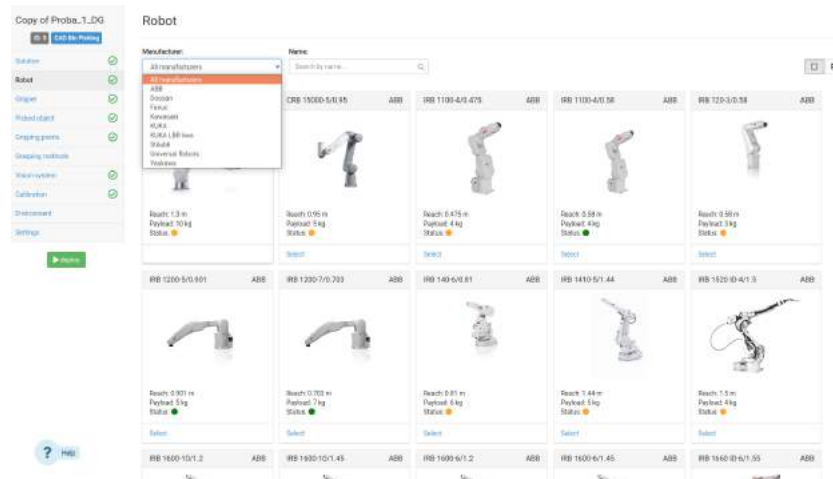
Zdjęcie 4: Ustawienia Vision system

Niestety ten system wizyjny wymaga również rysunku modelu pobieranego elementu. W tym przypadku również skorzystaliśmy z gotowych rysunków przykładowych elementów dostarczonych przez firmę.

Kolejnym krokiem jest skonfigurowanie sposobu pobierania elementu.



Zdjęcie 5: Ustawienia Vision system

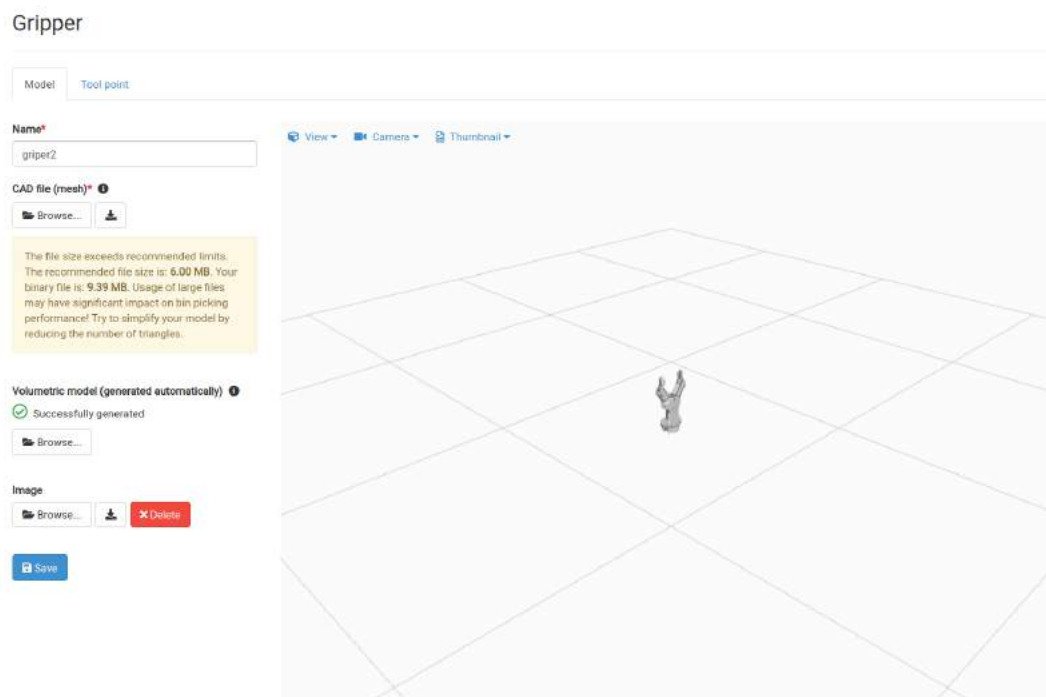


Zdjęcie 6: Wybór modelu robota

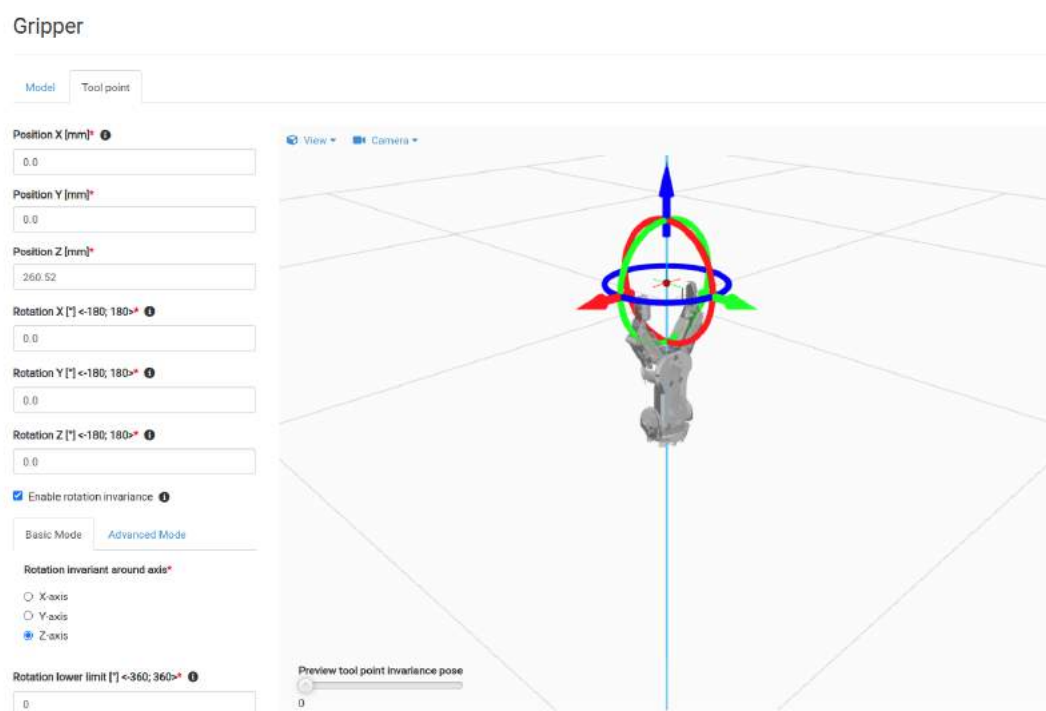
Jak widać na zdjęciu powyżej, możemy ustawić w jakiej pozycji ma być pobierany element, oraz możemy wybrać orientację samego chwytaka. Ze względu na rozwiązywanie problemu bin-picking efektor musi pobierać element z różnych pozycji, z różnych stron. Poniżej przedstawiono opisaną procedurę pobierania.

1.4.1 Kalibracja

W tym etapie należy przeprowadzić kalibrację kamery przy użyciu robota oraz piłki kalibracyjnej. Na początku należy wybrać model systemu wizyjnego, a następnie włączyć kalibrację. Później należy włączyć wcześniej napisany program na robocie z dziewięcioma różnymi pozycjami do kalibracji.



Zdjęcie 7: Wybór modelu efektora



Zdjęcie 8: Edycja TCP efektora

Po tym etapie macierz kalibracji ze zdjęcia 5 wypełni się wartościami dla systemu wizyjnego.

Picked object

Name*

Description

CAD file (mesh)* ⓘ

Volumetric model (generated automatically) ⓘ
☒ Successfully generated

Image

View Camera Thumbnail

Zdjęcie 9: Model elementu 3D

Gripping points

Editing selected gripping point for picked object - profil_pelny

Unique ID*

Priority*
☒ Normal

Position X [mm]*

Position Y [mm]*

Position Z [mm]*

Rotation X [°] <-180; 180>* ⓘ

Rotation Y [°] <-180; 180>* ⓘ

Rotation Z [°] <-180; 180>* ⓘ

☒ Enable rotation invariance ⓘ

Rotation invariant around axis*
☒ X-axis
☐ Y-axis
☐ Z-axis

View Camera Thumbnail Visibility

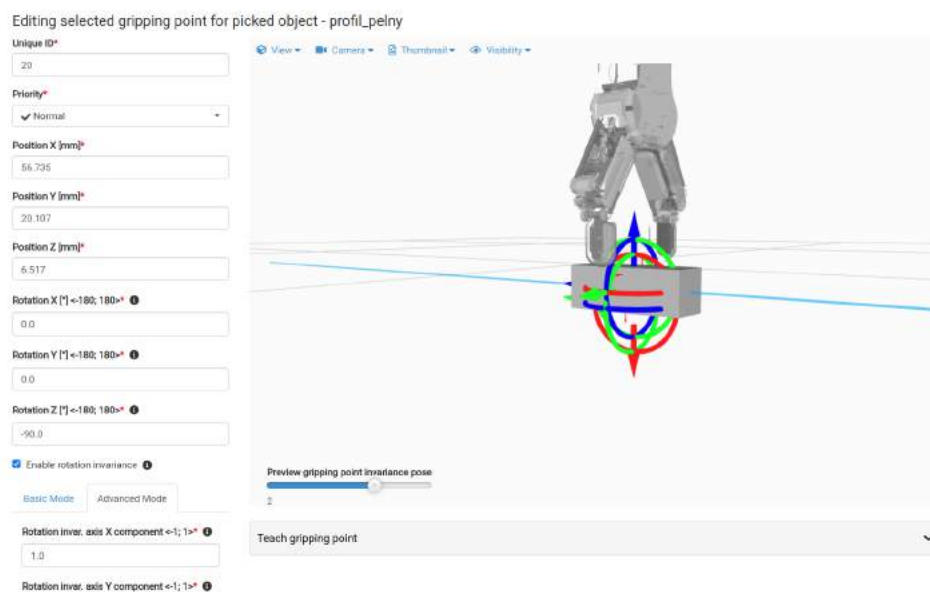
Preview gripping point invariance pose

Zdjęcie 10: Ustawienie pobierania elementu przez robota

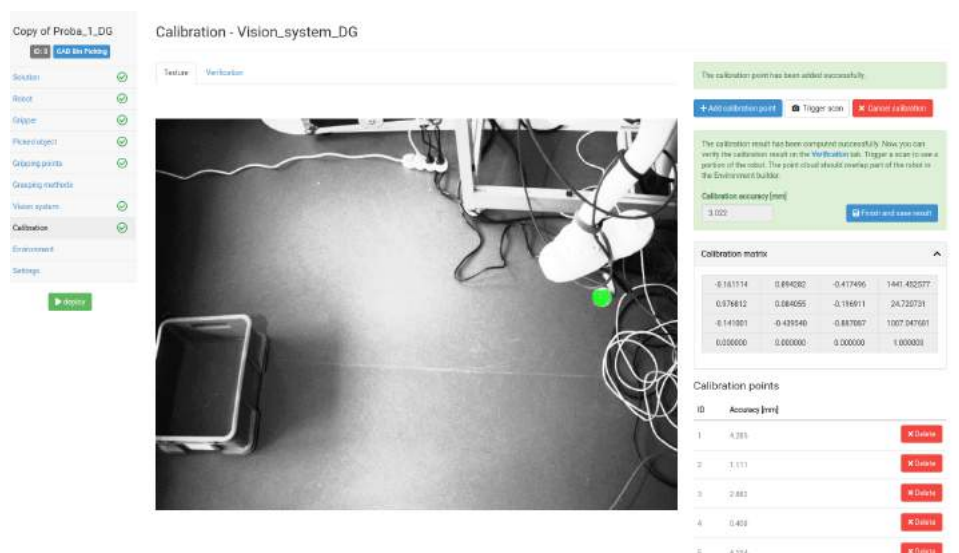
1.4.2 Strefy pracy robota

W tej zakładce zdefiniowany został obszar roboczy robota, jego podstawa, widok systemu wizyjnego oraz po zeskanowaniu modele obiektu. Można tutaj również dodać modele obiektów takich jak pojemniki, po wcześniejszym zaimplementowaniu ich rysunków 3D.

Gripping points



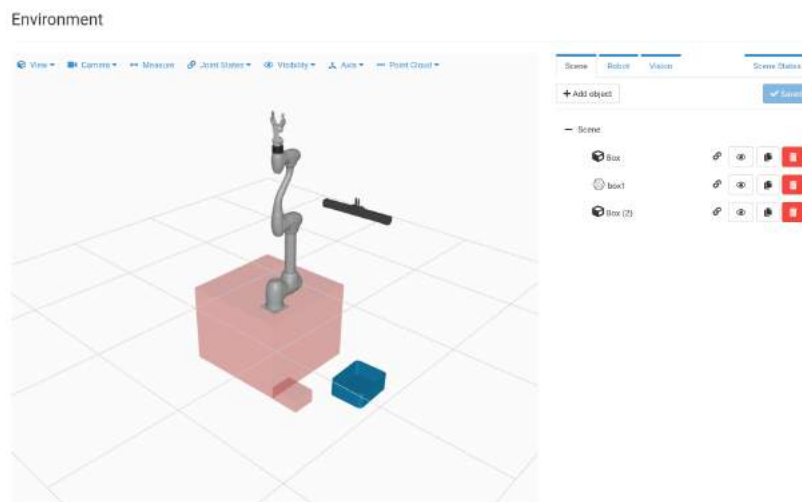
Zdjęcie 11: Ustawienie pobierania elementu przez robota



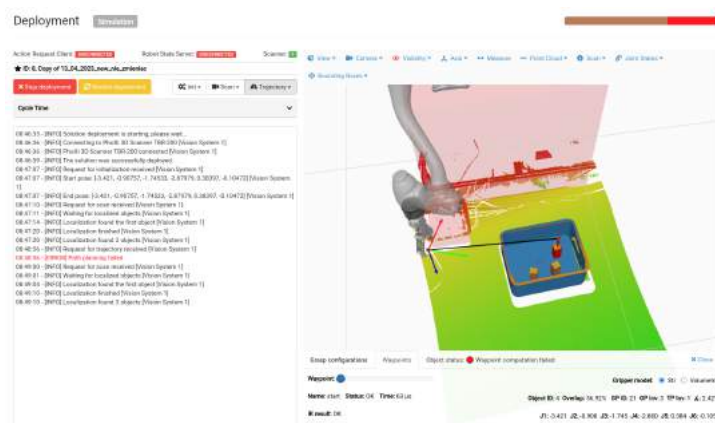
Zdjęcie 12: Podglądowe zdjęcie kalibracji

1.5 Symulacja

W tym etapie następuje planowanie ruchu robota i sposobu pobierania i odkładania elementów. Na początku możemy wykonać skan na którym zostaną wykryte elementy zdefiniowane we wcześniejszym etapie. Proces ten pomoże nam w prawidłowym ustaleniu ruchu robota.

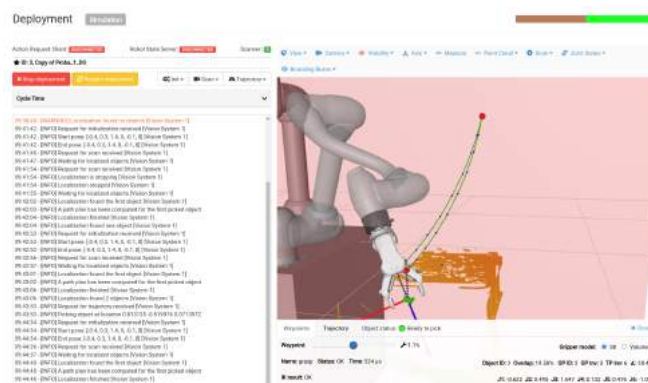


Zdjęcie 13: Obszary robocze robota



Zdjęcie 14: Skan poglądowy środowiska pracy

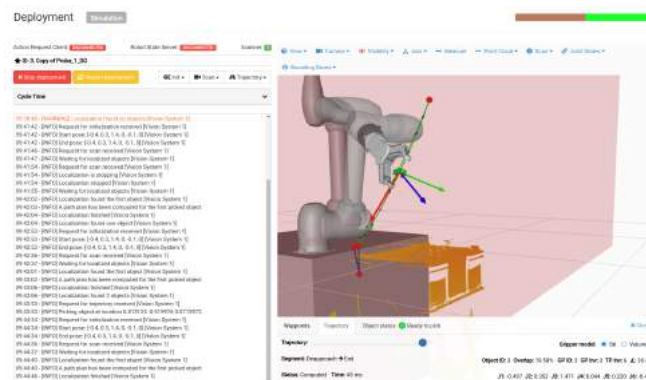
W następnym kroku dzięki podglądowi z kamery i symulacji należy ustalić trajektorię ruchu robota od pozycji startowej poprzez obszar w którym znajdują się elementy do pobrania, aż do wypuszczenia elementu z uchwytu.



Zdjęcie 15: Tworzenie trajektorii ruchu

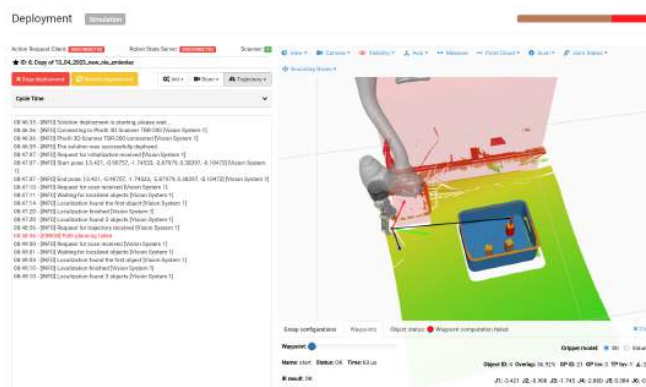
Na koniec gdy cały ruch został zaplanowany możemy uruchomić symulację która pozwoli

na wyeliminowanie ewentualnych błędów i kolizji.



Zdjęcie 16: Symulacja ruchu robota

Podjęto również udaną próbę zmiany pozycji początkowej oraz końcowej trajektorii tak, aby obiekty odkładane zostały w miejsce oddalone od wizji kamery.



Zdjęcie 17: Zmiana punktu początkowego oraz końcowego trajektorii

2 Zadania realizowane w lab. 010

2.1 Realizowane zadania

Na robocie Universal Robot 3, z wbudowanym systemem wizyjnym Wrist Camera firmy ROBOTIQ, postanowiono wykorzystać system wizyjny do wykonania następujących zadań:

1. Realizacja zadania bin-pickingu.
2. Rozszerzenie poprzedniego zadania o rozpoznawanie barw oraz o system paletyzacji.
3. Wykorzystanie czujnika siły.

2.2 Kalibracja systemu wizyjnego *Wrist Camera*

Aby zapewnić poprawne działanie układ niezbędnym był dobrze skalibrowany system wizyjny robota, w tym przypadku *Wrist Camera* firmy ROBOTIQ, który był w zasadzie mechaniczną częścią chwytaka.

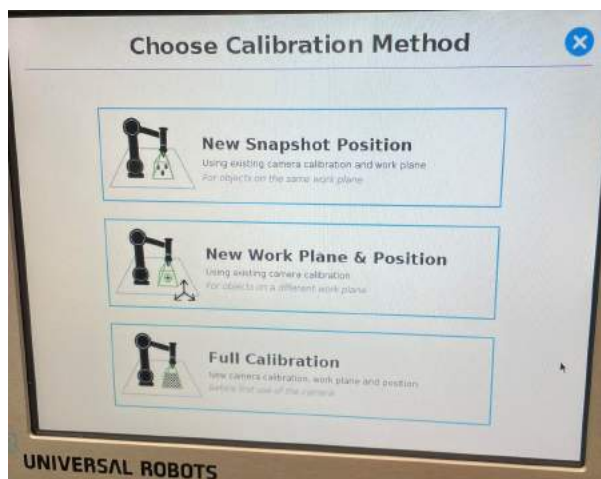
2.3 Montaż kamery

Z racji na to że cały robot został już wcześniej poprawnie zainstalowany a kamera zamontowana i podłączona do komputera, ominął nas etap instalacji mechanicznej.

2.3.1 Oprogramowanie

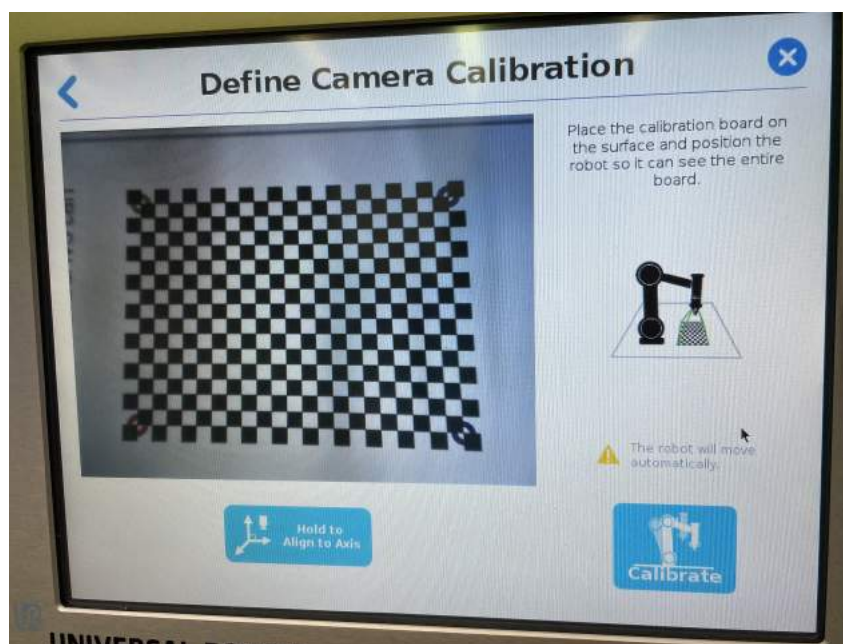
Korzystając z odpowiedniego oprogramowania dla robotów marki *Universal Robots* dostarczonego przez firmę ROBOTIQ należało przed samą implementacją rozwiązania problemu wykonać następującą kalibrację

Zaczynając kalibrację oprogramowanie prosi nas o wykonanie tzw. Snapshot'a 18, przez co można rozumieć ogólny widok przestrzeni roboczej robota w celu potwierdzenia informacji dotyczących pozycji, położenia i orientacji kamery. Samo wykonanie Snapshot'a polega na wybraniu jednej opcji w zakładce **Installation**.

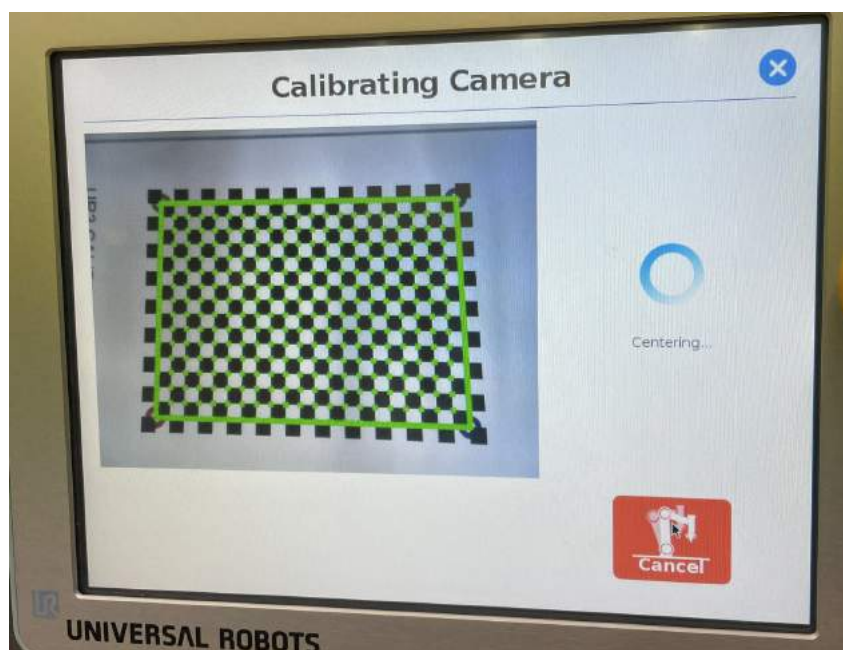


Zdjęcie 18: Wybór nowego Snapshot'a

Po ustawieniu robota w pozycji w specjalny panel kalibracyjny z szachownicą będzie w zasięgu widoku *Wrist Camera* robota należy wtedy wyprostować robota względem osi - przycisk **Hold to Align to Axis** 19. A następnie kliknąć przycisk **Calibrate** aby uruchomić proces automatycznej kalibracji 20.



Zdjęcie 19: Przygotowania do kalibracji



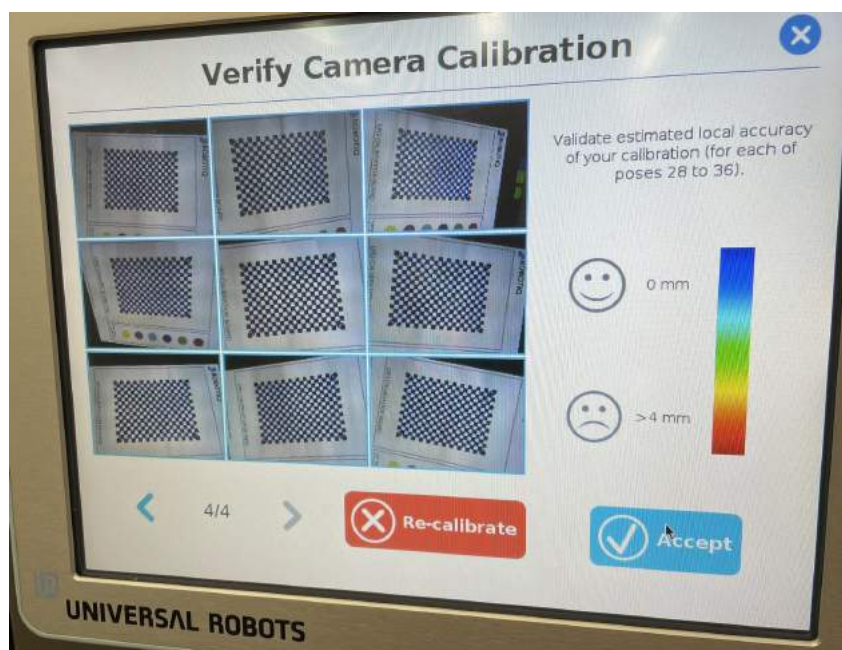
Zdjęcie 20: Początek kalibracji

Po rozpoczęciu kalibracji, robot automatycznie ustawia się w 36 różnych pozycji aby na każdej z nich zrobić zdjęcie szachownicy kalibracyjnej z innej perspektywy 21.



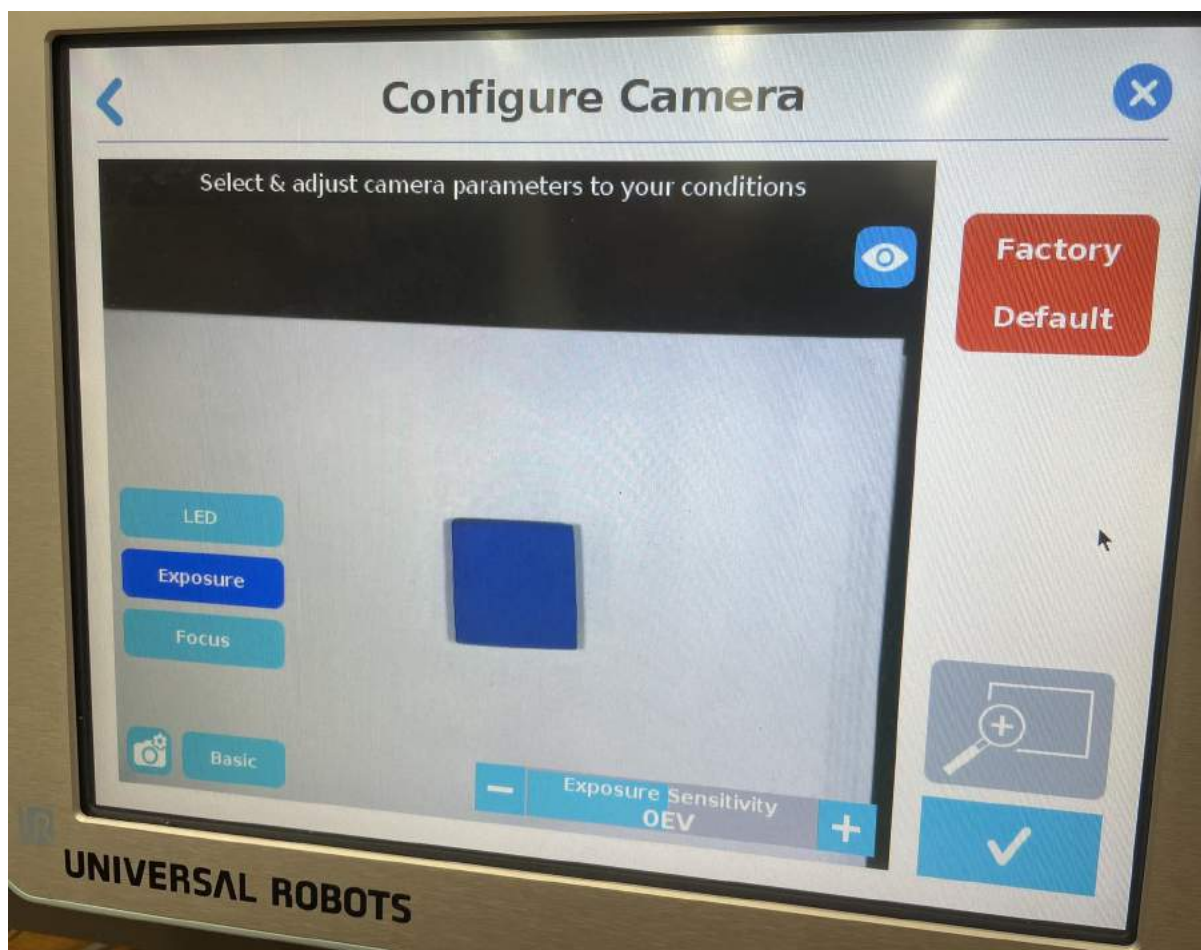
Zdjęcie 21: Tok wykonywania zdjęć z różnych perspektyw

Po wykonaniu zdjęć system okazuje nam ekran kontroli jakości kalibracji, wskazuje ilość wystarczająco celnie wykonanych fotografii 22.



Zdjęcie 22: Kontrola jakości kalibracji

Po poprawnej kalibracji, oprogramowanie dostarcza również opcję dostosowania ustawień samej kamery takie jak np. natężenie kolorów, ekspozycja świetlna, czułość czy też skupienie kamery, wszystkie te ustawienia są dostępne w oknie interfejsu 23. Oprócz dopasowywania ustawień samej kamery, warto pomyśleć o jednolitym tle nad którym robot będzie wykonywał operacje. W naszym przypadku sprawdził się biały brystol, na którym robot nie miał żadnego problemu z identyfikacją przedmiotów.



Zdjęcie 23: Ustawienia kamery

2.4 Problem bin-picking'u

2.4.1 Opis

Bin-picking realizowany na tym robocie realizowany był na sześciennych kostkach, gdzie rozpoznawany jest tylko i wyłącznie kształt obiektu. Pobierane obiekty odkładane były w linii obok wyimaginowanego pojemnika z elementami.

2.4.1.1 Założenia

- Kostki: Sześcienne kostki mają wymiary 3x3x3 cm. Mogą mieć różne kolory, ale rozpoznawany jest tylko kształt.
- Pojemnik z elementami: Kostki są przechowywane w wyimaginowanym pojemniku lub obszarze. Nie ma znaczenia kolejność lub układ kostek w pojemniku.
- Pobieranie i odkładanie obiektów: Robot UR3 korzysta z chwytaka, który jest odpowiednio dostosowany do pobierania sześciennych kostek o wymiarach 3x3x3 cm. Na podstawie informacji dostarczanych przez system wizyjny Wrist Camera, robot identyfikuje i pobiera żadaną kostkę z pojemnika, a następnie przenosi ją w odpowiednie miejsce.
- Linia odkładania: Kostki są odkładane w linii obok pojemnika. Można ustawić stałą odległość między kolejnymi kostkami, na przykład 5 cm, lub dostosować odległość w zależności od preferencji.
- Powtarzalność: Zadanie bin pickingu może być wykonywane w sposób powtarzalny. Robot UR3 może kontynuować pobieranie i odkładanie kostek w linii, dopóki wszystkie żądane elementy nie zostaną wydobyte z pojemnika.

2.4.1.2 Realizacja

Realizacja zadania bin pickingu sześciennych kostek przy użyciu robota UR3 i systemu wizyjnego Wrist Camera przebiega według następującego opisu:

Robot UR3 wykonuje ruchy, aby dotrzeć do ustalonego punktu "snapshot". Jest to pozycja, w której chwytak robota znajduje się nad obszarem, który zapewnia odpowiednie pole widzenia dla kamery Wrist Camera. Pozycja ta pozwala na uruchomienie procesu wizyjnego.

Po osiągnięciu punktu "snapshot" robot rozpoczyna proces wizyjny. Kamera Wrist Camera analizuje obraz i wyszukuje elementy o pierwszym zadanym kształcie (np. sześciany o wymiarach 3x3x3 cm). Algorytmy wizyjne lub sieci neuronowe mogą być wykorzystane do rozpoznawania kształtów kostek na podstawie cech charakterystycznych.

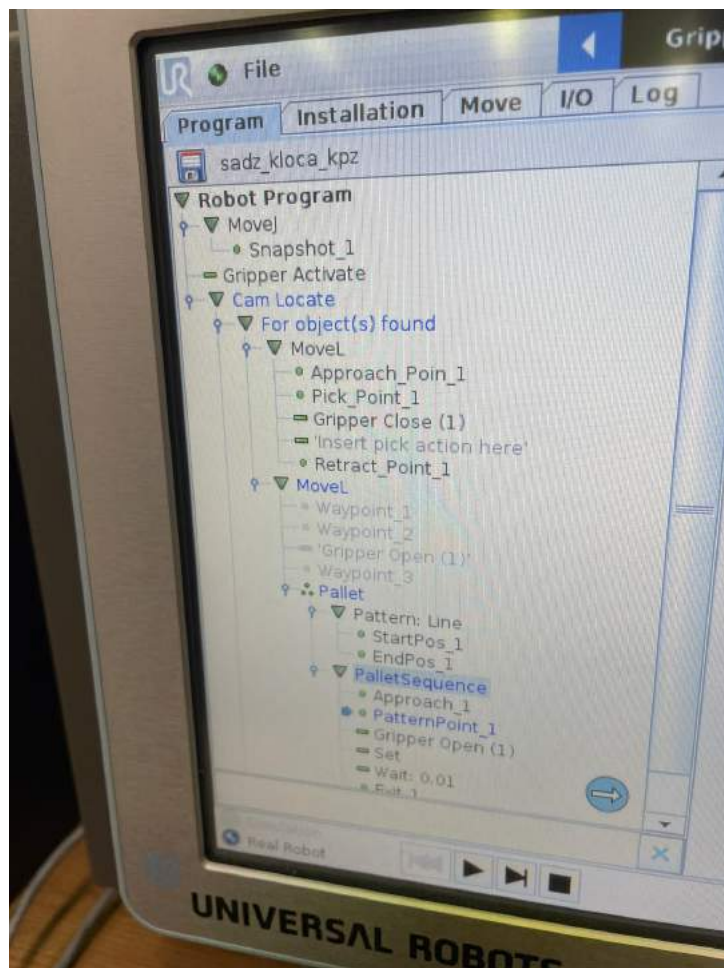
Po zidentyfikowaniu kostki o zadanym kształcie, robot UR3 wykonuje ruchy, aby odpowiednio ją pobrać i przenieść do docelowego miejsca. Mogą być wykorzystane chwytaki dostosowane do manipulacji sześciennymi kostkami o wymiarach 3x3x3 cm. Po pobraniu kostki robot przemieszcza się i odkłada ją w linii obok pojemnika.

Robot UR3 kontynuuje proces wizyjny, przesuając się ponownie do pozycji "snapshot". Wyszukuje pozostałe obiekty o zadanym kształcie i powtarza czynności pobierania i odkładania, dopóki wszystkie wykryte przedmioty nie zostaną odłożone.

Powtarzanie procesu może odbywać się aż do wyczerpania wszystkich kostek o zadanym kształcie w pojemniku. W przypadku braku dalszych kostek, robot może zakończyć zadanie bin pickingu.

2.4.1.3 Algorytm programu

Na poniższym zdjęciu ukazany został układ programu. Składa się on z kilku głównych części.



Zdjęcie 24: Układ Programu

Główny program - "Robot Program": Jest to obszar, w którym programista tworzy logikę działania aplikacji. Pierwszym krokiem jest dotarcie do pozycji "snapshot", która w tym przypadku została nazwana jako "Snapshot4". Dotarcie do tej pozycji odbywa się poprzez ruch w przestrzeni przegubowej robota.

Wykrywanie obiektów: Funkcja "Cam Locate" została wykorzystana w programie. każde jej użycie odpowiada za proces wykrywania sześciennych kostek. Proces wykrywania obiektów może być wykonany przy użyciu metody automatycznej lub parametrycznej. Zakłada się, że na obszarze roboczym może pojawić się maksymalnie dziesięć obiektów danego typu.

Pobieranie obiektów: Kolejnym etapem programu jest stworzenie instrukcji odpowiedzialnych za pobieranie wykrytych obiektów. Proces ten jest analogiczny dla obu sekcji "Cam

Locate”. Polega na dotarciu do pozycji nad polem roboczym (np. ”Waypoint1” i ”Waypoint2”) i zdefiniowaniu punktu, który zapewnia neutralną pozycję przed etapem pobierania. Pobieranie odbywa się względem punktu ”snapshot”. Wykorzystywane są instrukcje ruchowe generujące odpowiednią ścieżkę, a także instrukcje sterujące chwytakiem robota. Odkładanie obiektów: Ostatnim etapem programu jest odkładanie pobranych obiektów do odpowiedniego pojemnika. Proces ten obejmuje ruch liniowy do punktu bezpiecznego oddalonego od obszaru roboczego (np. ”Waypoint3” i ”Waypoint4”) oraz dojazd do pudełka w konfiguracji przegubowej. Następnie następuje zwolnienie chwytaka.

2.4.2 Podsumowanie

Podsumowując, program bin pickingu realizuje zadanie sortowania obiektów według kształtu i odkładania ich do pojemnika. Opiera się na wykrywaniu obiektów przy użyciu kamery oraz sterowaniu robotem UR3 w celu pobrania i odkładania obiektów.

2.5 Rozpoznawanie barw

Program realizuje proste sortowanie takich samych elementów i odkładanie ich w konkretne, z góry ustalone, segmenty.

2.5.1 Opis

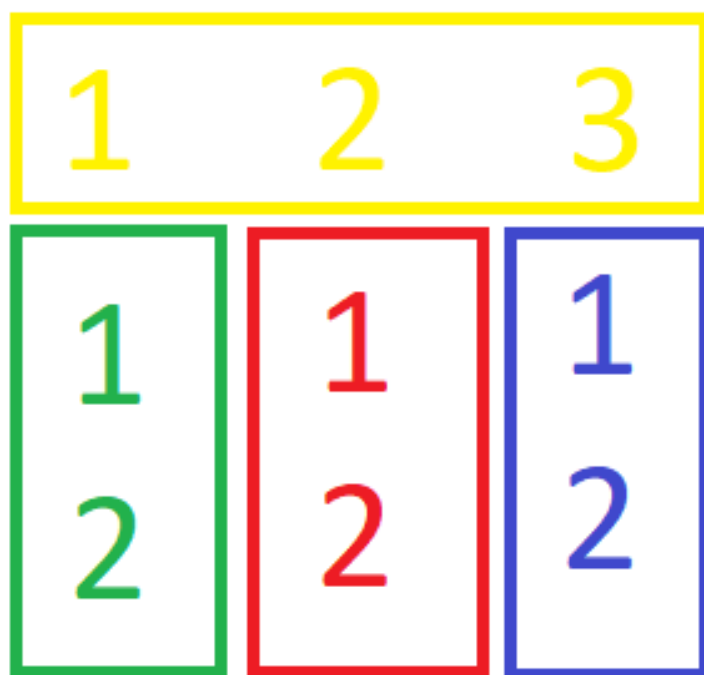
Zadanie to opierało się o poprzednie, jednak postanowiono, że robot nauczy się rozpoznawać barwy i w zależności od pobranego elementu danego koloru odłoży go na odpowiedni segment wyznaczony tylko i wyłącznie dla tych barw. Wszystkie elementy to były sześciiany jak poprzednio.

2.5.2 Realizacja

Do realizacji zadania wykorzystano skalibrowaną kamerę, której widok przedstawiony został na rysunku 25. Do samego zadania potrzebny również był chwytak. Wykorzystano efektor 2F-85.

2.5.2.1 Założenia

Przyjęto, że elementy umieszcza będą według poniższego rysunku (rys. 25).



Zdjęcie 25: Zobrazowanie obszarów odkładania kolorów

Jak widać idealnie sprawdzi się tutaj system paletyzacji, który dbać będzie o to, aby elementy były równo odkładane i nie dochodziło do kolizji. Numery oznaczają kolejność odkładania obiektów. Jedyńka jest pierwszym miejsce odkładczym, dwójka – drugim, trójka – trzecim.



Zdjęcie 26: Nauka rozpoznawania barw obiektu

2.5.2.2 Dodanie możliwości rozpoznawania barw

W celu realizacji postawionego zadania należało nauczyć robota rozpoznawania barw. Zrobiono poprzez funkcję wbudowaną w *Cam Locate*. Można w niej ustawić rozpoznawaną barwę, oraz jak duży obszar ma być brany pod uwagę (rys. 26). Następnie w oknie głównym należy wybrać prób, po którym system rozpozna dany kolor elementu. Kolejno należało pobierać rozpoznane elementy, jednak tutaj posłużono się rozwiązaniem zastosowanym w problemie bin-picking'u, czyli ustawiono jak blisko siebie mogą być ustawione elementy, aby można było je pobrać bez kolizji efektora z pobieranym obiektem.

2.5.2.3 Dodanie systemu paletyzacji w oparciu o znaleziony kształt

Zastosowany system paletyzacji jest dokładnie taki sam jak poprzednim rozdziale. Należało tylko wykorzystać układanie tych elementów w sposób taki jak na rysunku 25, czyli odpowiednio dodać system paletyzacji dla obiektów o różnych kształtach.

2.5.2.4 Przeprowadzenie testów

Przeprowadzono kilka testów mających na celu dobranie najlepszych parametrów tj. procentowe otwarcie chwytaka oraz siła i prędkość z jaką ma pobierać elementy. Zdarzało się, że elementy, pomimo tego, że były w zasięgu kamery, nie były wykrywane, więc należało odpowiednio zmienić próg wykrywania elementu lub ewentualnie próg rozpoznawania barw, czyli momentu kiedy kamera stwierdzi, że dany kolor jest tym kolorem. Po kilku poprawkach udało się uzyskać efekt taki jak na rys 28.



Zdjęcie 27: Wynik badań – sortowanie barw

2.5.3 Podsumowanie

Jak widać kamera oraz robot bardzo dobrze poradziły sobie z postawionym wyżej problemem. Możliwa jest rozbudowa zadania o rozpoznawanie innych kształtów i odkładanie ich w inne, bądź to samo, dedykowane miejsce. Czasami system wizyjny przekazywał informację, że robot nie może pobrać danego elementu danego koloru, więc przechodził do kolejnego. Było to spowodowane, prawdopodobnie, przez ułożenie elementów do poprania blisko siebie i system wizyjny stwierdził, że nie będzie to element wygodny do pobrania na tą chwilę. Widać więc, że kamerka podejmuje decyzję czy podjechać pod dany obiekt, czy też nie. nie wykonuje zadań „na ślepo”.

2.6 Czujnik siły

2.6.1 Opis

Założeniem wykorzystania czujnika siły było pobieranie poprzez system wizyjny elementów i odkładanie ich w jedno miejsce tak, aby ułożyć z nich wieżę. Model czujnika siły to *FT 300*, który zamontowany jest na manipulatorze UR3.

2.6.2 Realizacja

Do realizacji tego zadania wykorzystano system wizyjny (opisany we wcześniejszych zadaniach) oraz czujnik siły, którego należało odpowiednio obsłużyć.

2.6.2.1 Włączenie czujnika

Aby móc wykorzystywać czujnik siły, należało go uruchomić. Udało się to dzięki dostarczonej funkcji *Force Control*, w której ustawiono, żeby czujnik siły reagował z siłą 5N na osi Z. Oznacza to, że gdy czujnik wykryje opór rozpocznie jakieś działania związane z tym zdarzeniem.

2.6.2.2 Odkładanie elementów

Miejsce odkładcze ustawiono na sztywno. Proces ten można nazwać paletyzacją, ale tylko w osi pionowej. Mianowicie, gdy robot pobrał element, korzystając z systemu wizyjnego, podjechał do wyznaczonego punktu w przestrzeni rozpoczęło się działanie funkcji związanej z obsługą czujnika siły. Robot powoli opadał wraz z elementem przez ustawiony okres czasu. Jeżeli napotkał przeszkodę, to znaczy podłoże lub odstawiony poprzednio element, chwytak rozwierał się, a robot odjeżdżał do ustalonej pozycji, która znajdowała się ponad wieżą. Następnie proces był powtarzany.

2.6.3 Podsumowanie

Czujnik siły można wykorzystać do różnych celów. Na przykład do znajdowania otworów na elementy. W takim przypadku stosowana jest technika spirali gdzie czujnik szuka miejsca, w którym umieści pobrany obiekt. Jak widać czujnik działa świetnie z połączeniem z systemem wizyjnym, który pomaga pobierać elementy. Obsługa czujnika jest prosta oraz skuteczna. Postawione zadanie zostało w pełni zrealizowane.



Zdjęcie 28: Wynik badań – układanie wieży

3 Porównanie

Tabela 1: Tabela porównawcza

Porównanie wykorzystanych systemów wizyjnych			
KPZ		UR3	
+	-	+	-
Kamera tego systemu posiada głębie (wie w jakiej odległości od niej leżą przedmioty)	Rozmiar kamery definiuje rozmiary wykrywanych obiektów	Przyjazne środowisko konfiguracyjne	Kamera tego systemu nie posiada głębi (nie wie jak dalego zjechać efektem aby pobrać element)
Przyjazne środowisko konfiguracyjne	Nauka kształtów obiektów odbywa się poprzez wgranie modelu 3D pobieranego obiektu	Nauka kształtów odbywa się poprzez zapamiętanie rozmiarów obiektów oraz robienie im zdjęć w jakim są położeniu	Brak możliwości zaimplementowania prawdziwego bin-picking'u
Losowo zorientowane elementy mogą być pobierane z różnych pozycji chwytaka	Brak możliwości rozpoznawania barw	System wizyjny pozwala na rozpoznawanie barw obiektów	Obszar działania systemu wizyjnego jest dość mały
Kamera posiada wysoką jakość obrazu monochromatycznego	Kalibracja robota odbywa się poprzez napisanie programu do dojazdu robota w określone miejsca	W jednym programie istnieje możliwość wykrywania różnych kształtów oraz barw obiektów	Prostszy system wizyjny niż stosowany w firmie
Kamera posiada opcję rozpoznającą ułożenie osi układów współrzędnych dla obiektów	Jednocześnie wykrywany może być tylko jeden kształt obiektu	Możliwość ustawienia odległości obiektów od siebie, aby np. chwytak mógł podjechać po element	
Duży obszar działania systemu wizyjnego	System stworzony głównie pod rozwiązania bin-picking'u	Ten system wizyjny, poprzez jego różnorodność, może świetnie sprawdzić się przy różnorodnych zadaniach	
Znacznie bardziej zaawansowany system wizyjny niż stosowany na robocie UR3	Redundantność robota przeszkadzała przy rozwiązaniu bin-pickingu		
Możliwość ustawienia stref, gdzie robot nie może się poruszać.			

Jednym z lepszych sposobów na zaprezentowanie pozytywów oraz negatywów danych rzeczy, czynności jest tabela. Zdecydowano, więc, że w takiej tabeli umieszczone zostaną plusy oraz minusy obu systemów wizyjnych

3.1 KPZ

W pierwszej kolejności opisany zostanie system wizyjny wypożyczony przez firmę w ramach współpracy z Konferencją Projektów Zespołowych.

3.1.1 Plusy

Pozytywne aspekty systemu wizyjnego.

3.1.1.1 Głębia

Kamera ta posiada głębie (obraz 3D) co oznacza, że robot, po otrzymaniu danych od systemu wizyjnego, wie na jaką głębokość ma zjechać, aby móc pobrać element. Bez tej opcji rozwiązanie bin-picking'u byłoby niemożliwe.

3.1.1.2 Osie

Kamera, w oparciu o modele 3D wgrywane do środowiska programistycznego, pozwala rozpoznanie osi elementu. Oznacza to, że jeżeli element będzie ustawiony pod pewnym kątem to robot odczytując taką informację, będzie wiedział, że do obiektu musi podejść tak, aby jego oś Z na efektorze zgadzała się z osią Z elementu. Ta opcja bezpośrednio wpływa na działanie bin-picking'u. Bez niej zadanie nie byłoby wykonalne.

3.1.2 Minusy

Negatywne aspekty systemu wizyjnego.

3.1.2.1 Barwy

System nie potrafi rozpoznać barw obiektów, przez co nadaje się on tylko do przekazywania robotowi, aby ten przenosił elementy z jednego miejsca w drugie, czy też pobierał i odkładał. Gdyby elementy miały ten sam model, ale byłyby innych kolorów, to robot robiłby to samo, ale niektóre kolory mogłyby nie zostać pobrane. Tracimy również możliwość rozbudowy takiego systemu o sortowanie elementów.

3.1.2.2 Kalibracja

Trudny i żmudny proces kalibracji, która nie jest prosta do ustawienia, powodowała liczne błędy. Należało dbać, aby kamera odpowiedzialna za pobieranie obrazu stała cały czas w tym samym miejscu. Inaczej należy powtórzyć kalibrację.

3.1.2.3 Redundantność

Robot **Doosan** jest robotem redundantnym, co oznacza, że może podejść do elementu na różne sposoby. Wielokrotnie przeszkadzało to w przeprowadzeniu symulacji, gdyż system przekazywał informację że w jednej konfiguracji będzie lepiej pobrać element, ale zapominał, że jest ustawiona strefa kolizji i robot nie może się tam znaleźć. Firma nie przewidziała możliwości wyboru konfiguracji robota, co skutecznie utrudniało pracę.

3.2 UR3

Cobot z laboratorium 010 w budynku C-3, który zamontowany ma na efektorze zamontowany system wizyjny.

3.2.1 Plusy

Pozytywne aspekty systemu wizyjnego.

3.2.1.1 Nauka

System nauki kamery nowych kształtów odbywa się poprzez podłożenie tego elementu, wykonanie zdjęcia, nadania rozmiarów, a następnie progę wykrywania krawędzi. Sposób ten jest szybszy i lepszy od systemu z KPZ. Ponadto interfejs obsługi tej kamery jest znacznie bardziej przyjaźniejszy od interfejsu kamery wypożyczonej przez firmę.

3.2.1.2 Różnorodność

Kamera ta pozwala prosto i szybko uczyć się nowych kształtów, a nawet ich kolorów i obsłużyć każdy z przypadków w jednym programie, czego nie potrafiła zrobić kamera zastosowana na robocie Doosan. Co więcej rozpoznawanie barw pozwala na poszerzenie możliwości robota o np. sortowanie elementów w różne kosze. System wizyjny dobrze współpracuje również z czujnikiem siły, który został wykorzystany przy realizacji jednego z zadań.

3.2.2 Minusy

Negatywne aspekty systemu wizyjnego.

3.2.2.1 Głębia

Kamera ta nie posiada głębi (obraz jest w postaci 2D) przez co przekazywała informację, że obiekt znajduje się w pewnej odległości, ale nie brała pod uwagę, że efektor musi jeszcze ten obiekt podnieść.

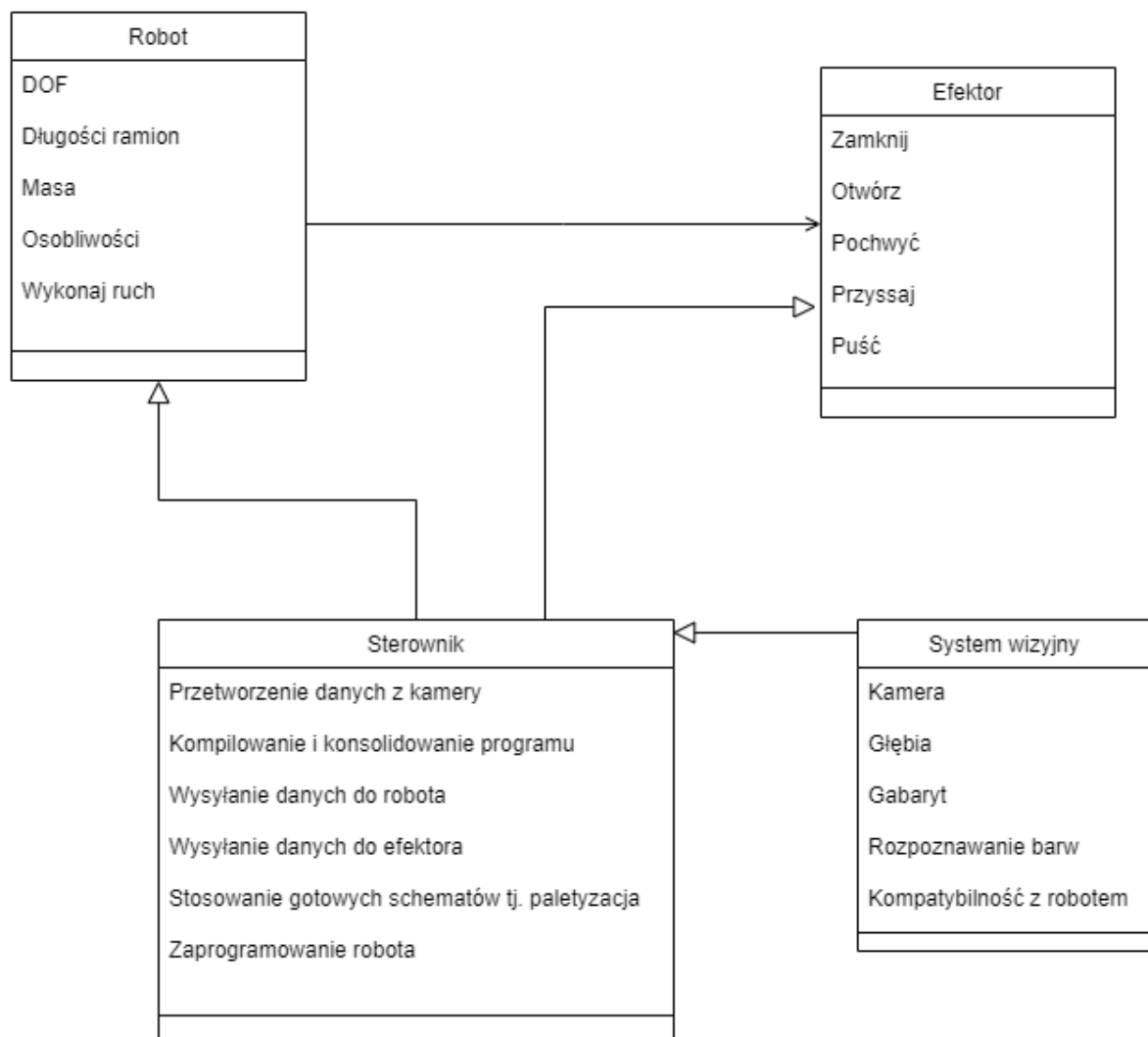
4 Podsumowanie

Podsumowując współpraca z firmą zewnętrzną świetnie sprawdziła się na początku. Później trzeba było samemu wymyślać zadania na robocie z laboratorium. Jak się okazało robot UR3 i jego system wizyjny jest znacznie przyjemniejszy w obsłudze jak i jest bardziej skuteczny niż wypożyczony system wizyjny w ramach KPZ. Widać również, że dobór kamery jest zależny od zadania jakie należy wykonać. Rozpoznawanie barw przydaje się przy sortowaniu elementów (system wizyjny robota UR3), natomiast głębia obrazu (3D) lepiej sprawdzi się przy zadaniach typu bin-picking, gdzie należy rozpoznawać orientację elementów.

Każde zadania były dokumentowane oraz opisywane, a zwieńczeniem tego jest powstanie strony internetowej do publikacji wyników oraz efektów prac: <https://manipulanci.mobirisesite.com/>.

5 Diagram UML

Poniżej przedstawiono prosty diagram UML, który pokazuje zasady działania i powiązania między poszczególnymi blokami. Widać tutaj, że zebrano ważniejsze wykorzystywane aspekty związane z problemem bin-picking'u oraz innych realizowanych zadań. Schemat jest prosty, gdyż w zadaniu najwięcej aspektów związanych było z konfiguracją sprzętu ze sobą.



Zdjęcie 29: Uproszczony diagram UML