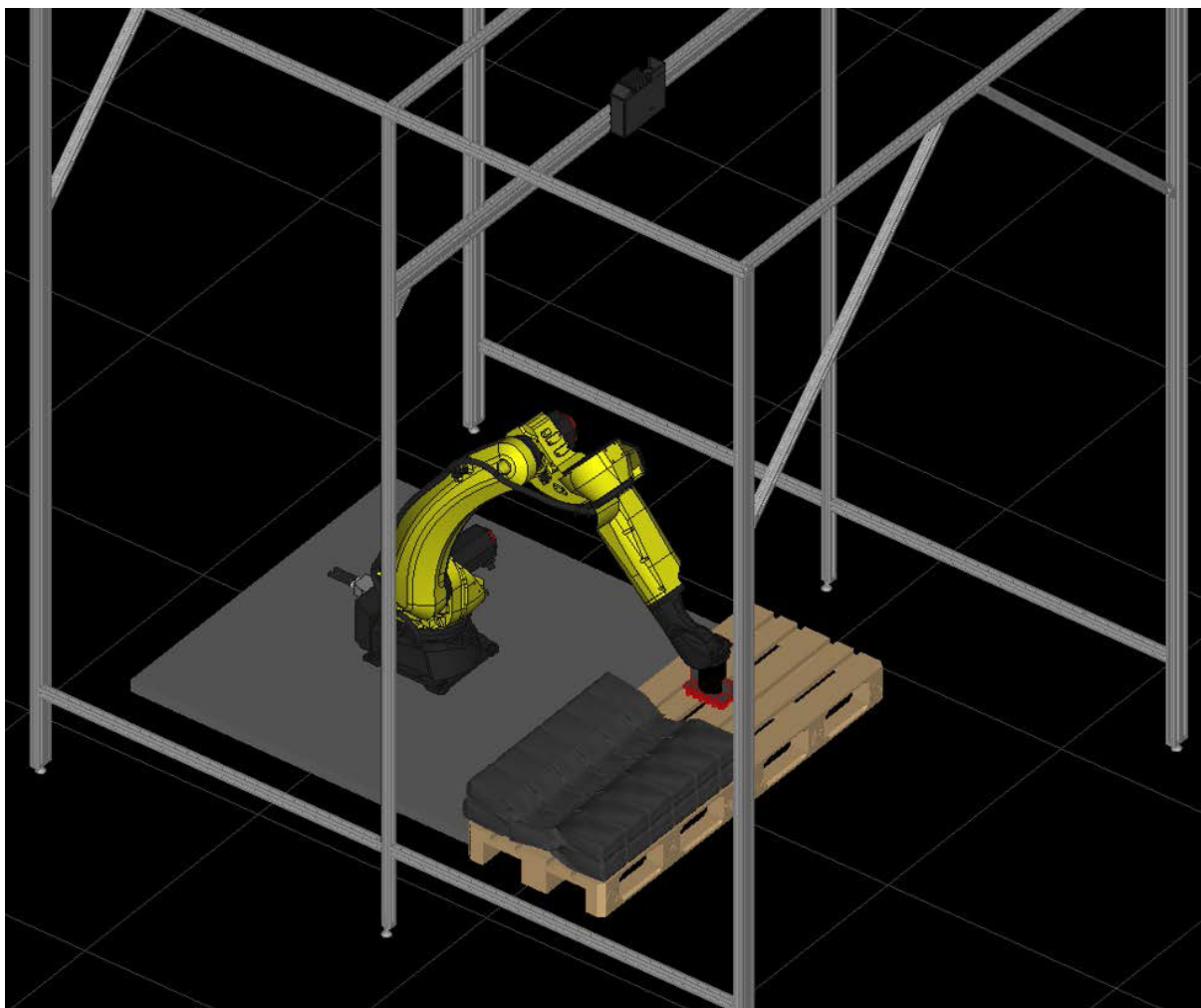


Depaletyzacja z wykorzystaniem systemu wizyjnego 3D Vision Sensor



Spis treści:

1. Opis sytuacji
2. Opis zadania
3. Zadania do wykonania
4. Elementy stanowiska
5. System oceniania

1. Opis sytuacji

Pewna firma produkcyjna posiada stanowisko z robotem przemysłowym. Zadaniem systemu zrobotyzowanego w tym zakładzie jest wspomaganie człowieka w procesie depaletyzacji artykułów spożywczych. Chcąc zwiększyć elastyczność produkcji oraz w celu optymalizacji procesu postanowiono wykorzystać możliwości systemu wizyjnego 3D. Realizacja założonych celów będzie możliwa poprzez odpowiednie skonfigurowanie systemu wizyjnego oraz prawidłowe zaprogramowanie robota. Elastyczność stanowiska wesprą funkcje systemu 3D Vision Sensor pozwalające na automatyczny odczyt położenia i orientacji detali oraz modyfikacji trajektorii ruchu narzędzia robota. Wdrożeniem nowego rozwiązania zajmuje się integrator systemów, firma, której jesteś pracownikiem. Zajmij się tym, powodzenia!

2. Opis zadania

W zadaniu oceniana będzie odpowiednia konfiguracja funkcji wykrywania położenia detali, w tym celu należy przeprowadzić proces końcowej konfiguracji przygotowanego wcześniej procesu wizyjnego. Dodatkowo należy napisać program TP realizujący proces szukania oraz depaletyzacji detali tj. przenoszenia ich z jednej palety na drugą, wykorzystując niezbędne instrukcje ruchowe oraz wizyjne. Dla przyśpieszenia pracy można skorzystać z przygotowanego wcześniej podprogramu CALC_PLACE.

Do tej pory zrealizowane zostały:

- fizyczny montaż stanowiska
- proces wyznaczenia punktu TCP
- Zdefiniowanie obciążenia flaszki robota
- podłączenie oraz komunikacja robota z systemem 3D Vision Sensor.
- dobór parametrów oświetlenia i pracy sensora wizyjnego

Ocenie podlega wykonanie każdego z podpunktów opisanych poniżej oraz czas wykonania zadania.

Za zrealizowanie celu zadania otrzymuje się dodatkowe punkty. Za udzielone przez opiekuna stanowiska podpowiedzi punkty są potrącane.

Cel główny:

Celem głównym jest końcowa konfiguracja procesu wizyjnego oraz stworzenie i uruchomienie programu TP umożliwiającego realizację procesu depaletyzacji czarnych worków z kawą, zlokalizowanych na półpalecie wewnątrz stanowiska zrobotyzowanego. W tym celu można wykorzystać program CALC_PLACE

```
CALC_PLACE 1/16
1: R[12:div]=R[10:current_place]
: DIV 12
2: R[11:mod]=R[10:current_place]
: MOD 12
3: R[7:layer_num]=R[12:div]
4: R[12:div]=R[11:mod] DIV 4
5: R[11:mod]=R[11:mod] MOD 4
6: R[8:row_num]=R[12:div]
7: R[9:column_num]=R[11:mod]
8: PR[14:current_position]=LPOS-
: LPOS
9: PR[14,1:current_position]=155*
: R[9:column_num]
10: PR[14,2:current_position]=230*
: R[8:row_num]
11: PR[14,3:current_position]=100*
: R[7:layer_num]
12:
13:
14:
```

Rysunek 1 Zrzut ekranu z TP przedstawiający strukturę programu CALC_PLACE

Opis procesu:

Program napisany przez użytkownika powinien umożliwić pobranie, a następnie spaletyzowanie kolejno wszystkich zlokalizowanych przez system wizyjny detali. Należy pamiętać o tym, że w tym celu **można użyć maksymalnie trzech indeksów punktów dojazdowych, np. P[1], P[2], P[3]**. Program powinien zaczynać się i kończyć w pozycji domowej. W pierwszej kolejności niezbędne będzie wykonanie zdjęcia palety, by móc pobrać dane z procesu. Następnie robot powinien realizować dojazd do detalu po trajektorii pionowej linii, a po jego pobraniu powrócić do punktu nad detalem. W kolejnej fazie należy zaprogramować sam proces paletyzacji, mając na uwadze zasady rządzące takimi procesami. Robot z detalem powinien dojeżdżać do punktu nad miejscem odłożenia, przesuniętego o połowę wymiarów paczki w płaszczyźnie palety. Umożliwi to dociśnięcie do siebie spaletyzowanych już wcześniej worków na palecie. Po spaletyzowaniu całej warstwy worków robot powraca do pozycji domowej i wykonuje kolejne zdjęcie, wymienione kroki powtarzają się. Po depaletyzacji dwóch warstw program powinien się zakończyć.

3. Zadania do wykonania

Aplikacja została podzielona na cztery, następujące po sobie etapy. Każdy z nich jest zadaniem do wykonania, aby przejść do kolejnego zadania należy w pełni ukończyć etap poprzedni.

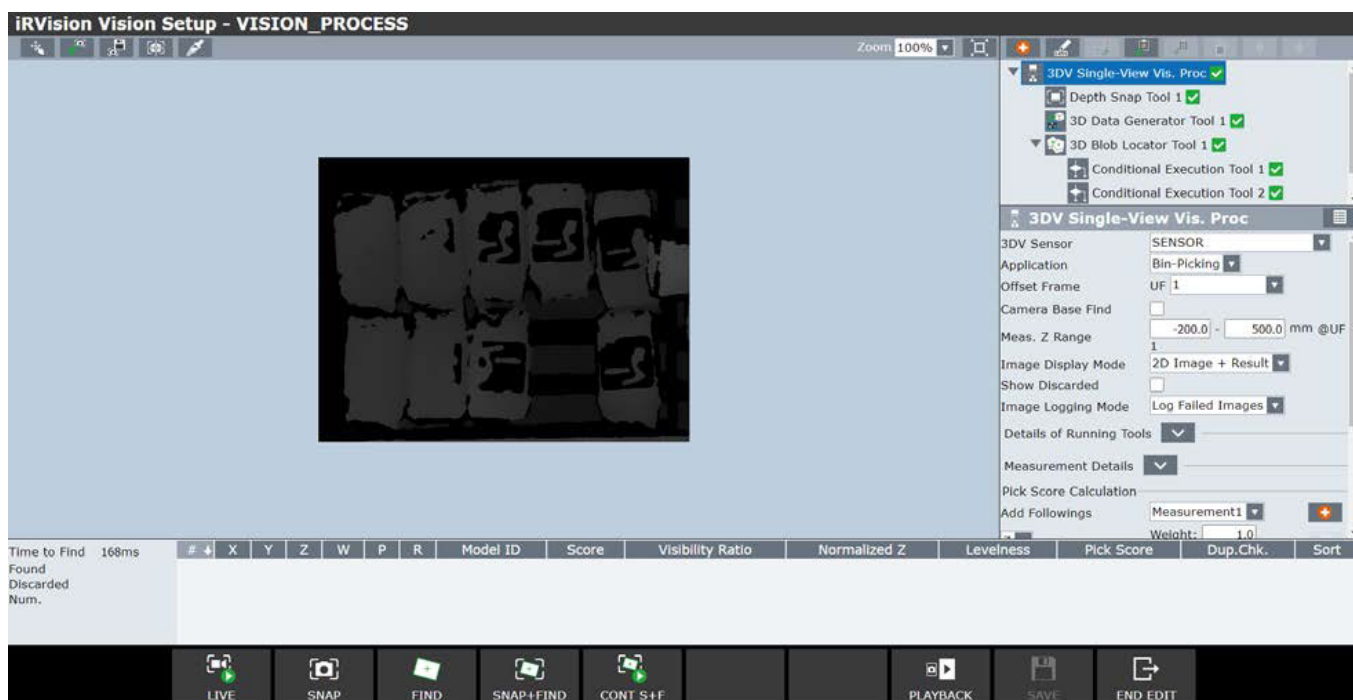
3.1. Końcowa konfiguracja parametrów systemu wizyjnego.

Procesy wizyjne działające w czasie rzeczywistym są zaawansowanymi procesami pomiarowymi wymagającymi dobrania odpowiednich parametrów, których ilość i jakość zależy w dużej mierze od charakteru samego procesu, właściwości detalu, charakterystyki oświetlenia itd.

W każdym przypadku proces wizyjny FANUC podzielony jest na dwa etapy. Etap pierwszy dotyczy konfiguracji sprzętowej samego sensora oraz jego kalibracji, natomiast etap drugi polega na dobraniu odpowiedniej procedury, obróbce danych wizyjnych, oraz doborze wykorzystywanych w procedurze narzędzi i odpowiedniej ich kalibracji.

Wykorzystywany w zadaniu proces został już skonfigurowany, a do uczestnika należy jedynie końcowa jego edycja, tak by można było wykorzystać go w kolejnym etapie pisania programu TP. Do procesu dostać można się zarówno z poziomu komputera PC jak i panelu programisty robota TeachPendant, przechodząc kolejno:

MENU > 8 iRVision > 1 VisionSetup



Rysunek 2 Zrzut ekranu z PC przedstawiający okno konfiguracyjne Vision Process Tool



Rysunek 3 Zrzut z ekranu TP przedstawiający okno konfiguracyjne CameraData

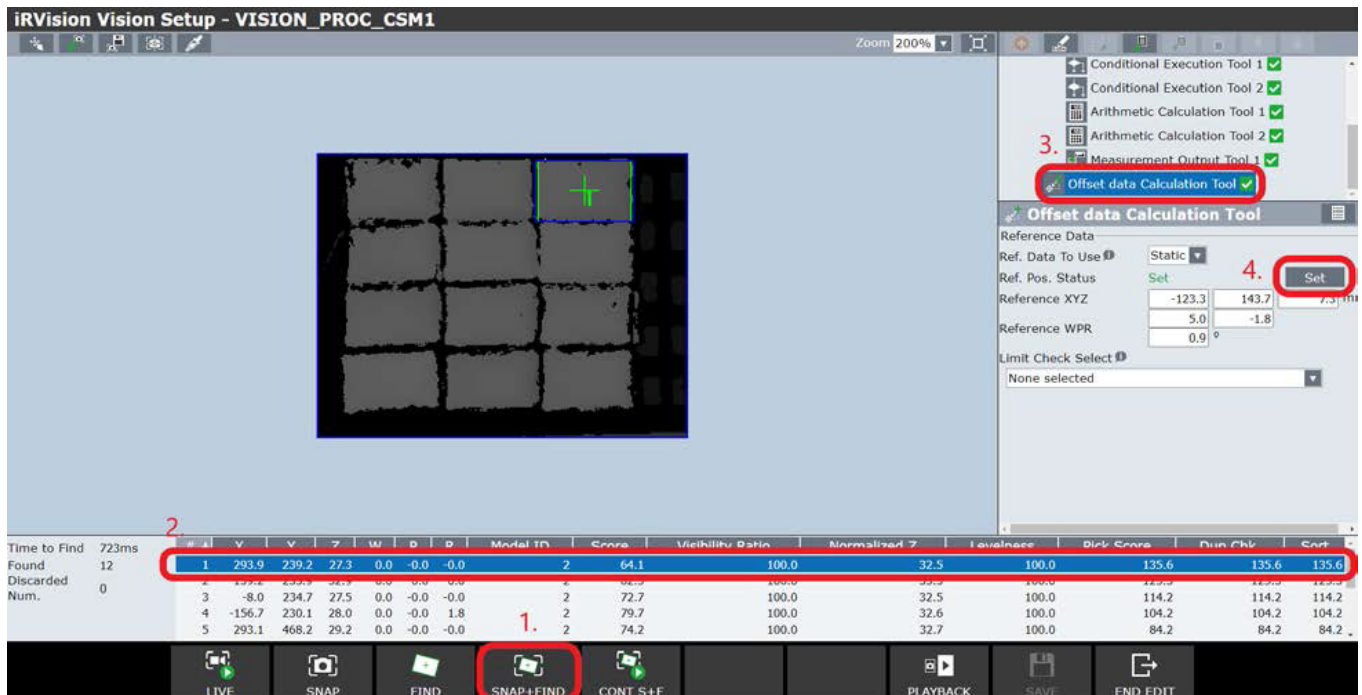
Treść zadania:

W zadaniu tym należy ustawić pozycję referencyjną procesu **VISION_PROC_CSM1** i tym samym umożliwić poprawne działanie aplikacji i przejście do kolejnego etapu.

Pozycja referencyjna jest to miejsce odniesienia w przestrzeni, względem, którego proces wizyjny oblicza wszelkie przesunięcia znalezionych detali.

Aby ustawić pozycję referencyjną:

W oknie procesu **VISION_PROC_CSM1** należy wykonać zdjęcie i przeprowadzić proces szukania. Pozycja referencyjna, ustawiana w zakładce **Offset data Calculation Tool** dotyczyć będzie jednego detalu, dlatego też z listy znalezionych elementów należy wybrać ten detal, od którego rozpoczniemy proces paletyzacji, co ułatwi pisanie programu TP.



Rysunek 4 Zrzut ekranu z PC ilustrujący kolejność poszczególnych operacji ustawiania pozycji referencyjnej

3.2. Stworzenie głównego programu TP i pobranie danych z procesu wizyjnego

Aby aplikacja działała zgodnie z założeniami należy wykorzystać dane zgromadzone przez proces wizyjny w etapie poprzednim.

Dostęp do danych w programie TP uzyskujemy poprzez rodzinę instrukcji VISION. Każda instrukcja wizyjna wymaga podania jako pierwszego argumentu nazwy procesu, do którego chcemy się odwołać.

Należy skorzystać z procesu **VISION_PROC_CSM1**

RUN_FIND – Instrukcja wywołuje wykonanie pomiaru procesu wizyjnego. Wykonane zostaje zdjęcie, które następnie poddawane jest analizie. Jako argument podajemy nazwę procesu, który chcemy wykonać.

```
VISION RUN_FIND  
'VISION_PROC_CSM1'
```

Rysunek 5 Zrzut ekranu TP z wykorzystaniem instrukcji RUN_FIND

GET_NFOUND – Komenda zwraca ilość rozpoznanych obiektów. Wynik zapisywany jest w podanym przez programistę rejestrze.

```
VISION GET_NFOUND  
'VISION_PROC_CSM1' R[2]
```

Rysunek 6 Zrzut ekranu TP z wykorzystaniem instrukcji GET_NFOUND

GET_OFFSET – Zapisuje wartość przesunięcia znalezionej części względem pozycji referencyjnej, w podanym przez programistę rejestrze wizyjnym.

```
VISION GET_OFFSET  
'VISION_PROC_CSM1' VR[R[1]]
```

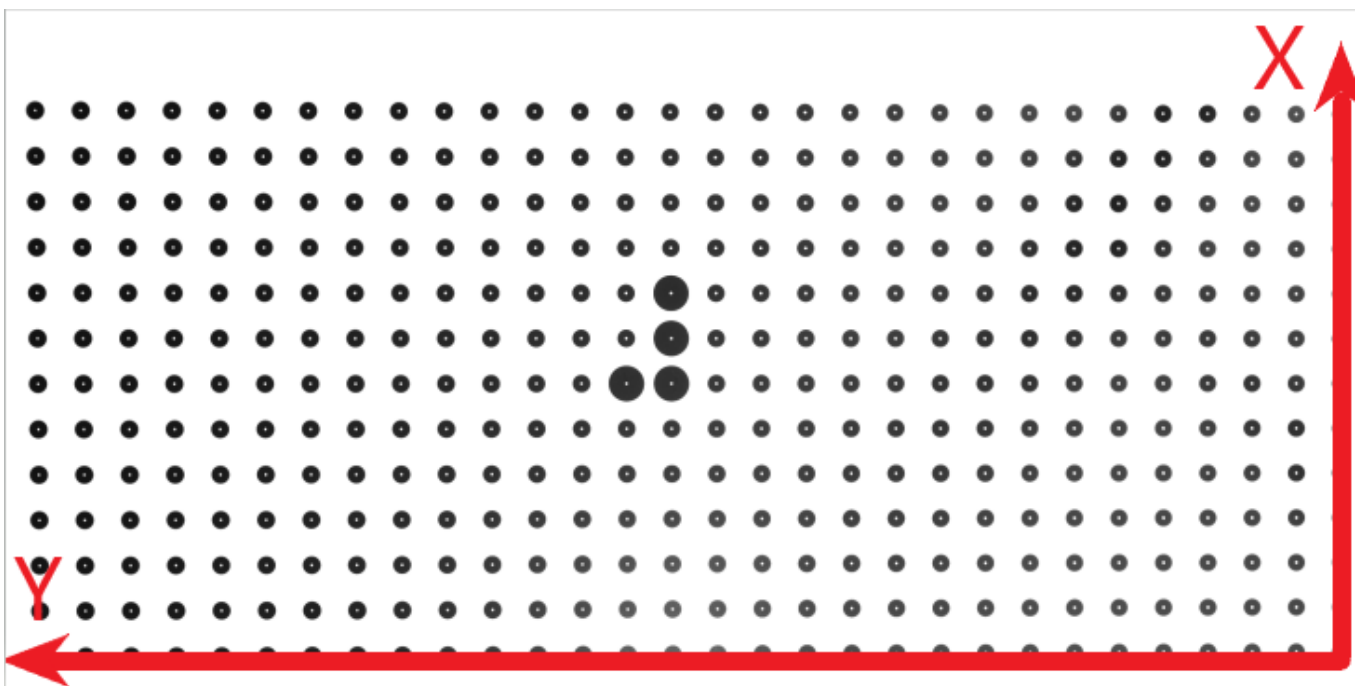
Rysunek 7 Zrzut ekranu TP z wykorzystaniem instrukcji GET_OFFSET

VOFFSET – komendę można wykorzystać przy instrukcji ruchowej. Powoduje ona przesunięcie punktu dojazdu o wartość zapisaną w rejestrze.

```
L P[2] 50mm/sec FINE  
VOFFSET,VR[R[1]]
```

Rysunek 8 Zrzut ekranu TP z wykorzystaniem instrukcji VOFFSET

Program powinien rozpoczynać się w punkcie domowym, bezpiecznym. Punkty do programu należy dodawać poruszając się w układzie użytkownika **UF1**, który został wyznaczony na siatce kalibracyjnej procesu wizyjnego



Rysunek 9 Zdjęcie siatki kalibracyjnej wykonane kamerą 3DVisionSensor

3.3. Zaprogramowanie poboru zlokalizowanych detali

W tym etapie należy rozwinąć napisany wcześniej program ucząc robota dojazdu do detalu i jego odkładania na sąsiedniej palecie, pamiętając o uwagach z punktu drugiego, opisanych na stronie trzeciej. Celem w tym etapie jest również stworzenie pętli głównej wykonującej się tyle razy, ile detali zostało zlokalizowanych w procesie wizyjnym.

Należy pamiętać o pionowym dojeździe do detalu.

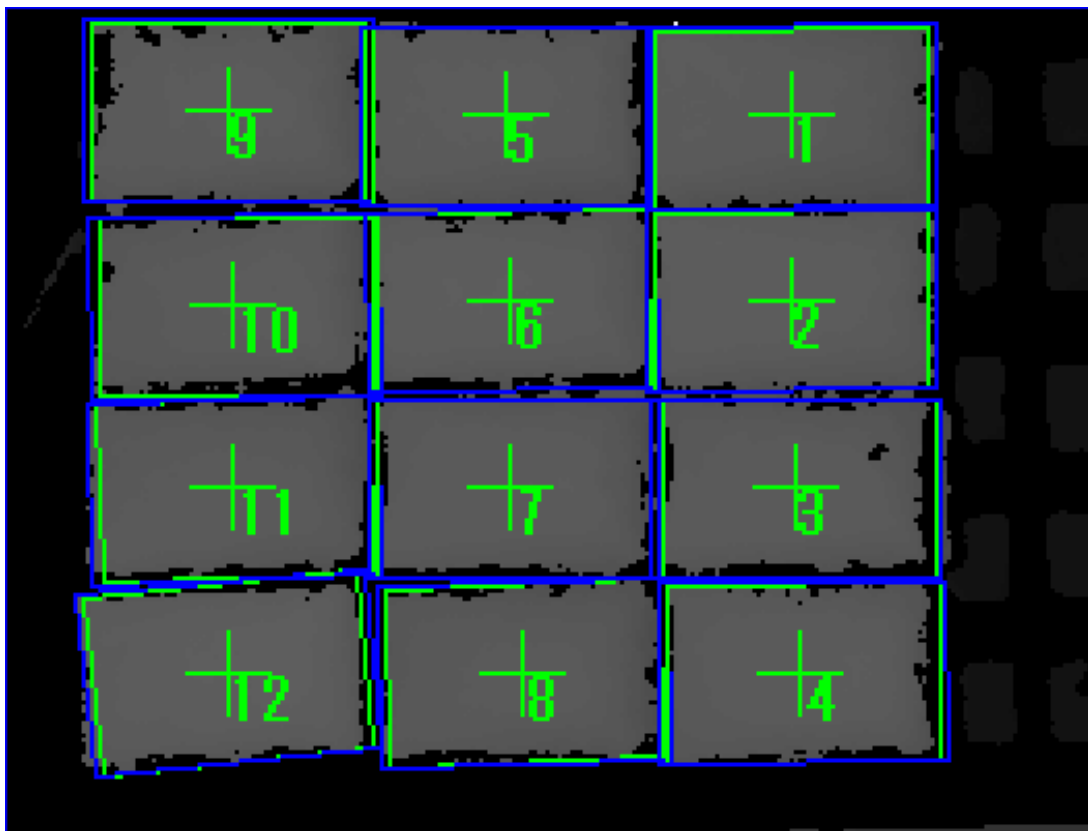
Na flanszy robota zainstalowany jest chwytak pneumatyczny dedykowany do detali wykorzystywanych w zadaniu. Do jego zamykania i otwierania służą programy macro:

VACUUM_ON - SU1

VACUUM_OFF - SU2

Dodatkowo, trzeba mieć na uwadze zdefiniowaną w parametrach procesu wizyjnego kolejność numerowania detali, jak również ich layout na palecie.

W przypadku niniejszego zadania numerowanie i lokalizacja detali w każdym przypadku będzie wyglądała następująco:



Rysunek 10 Zrzut ekranu PC z okna procesu wizyjnego przedstawiający kolejność numerowania elementów

Do realizacji ruchu robota bez detalu należy wykorzystać instrukcję **PAYLOAD[1]**

Do realizacji ruchu robota z detalem należy wykorzystać instrukcję **PAYLOAD[2]**

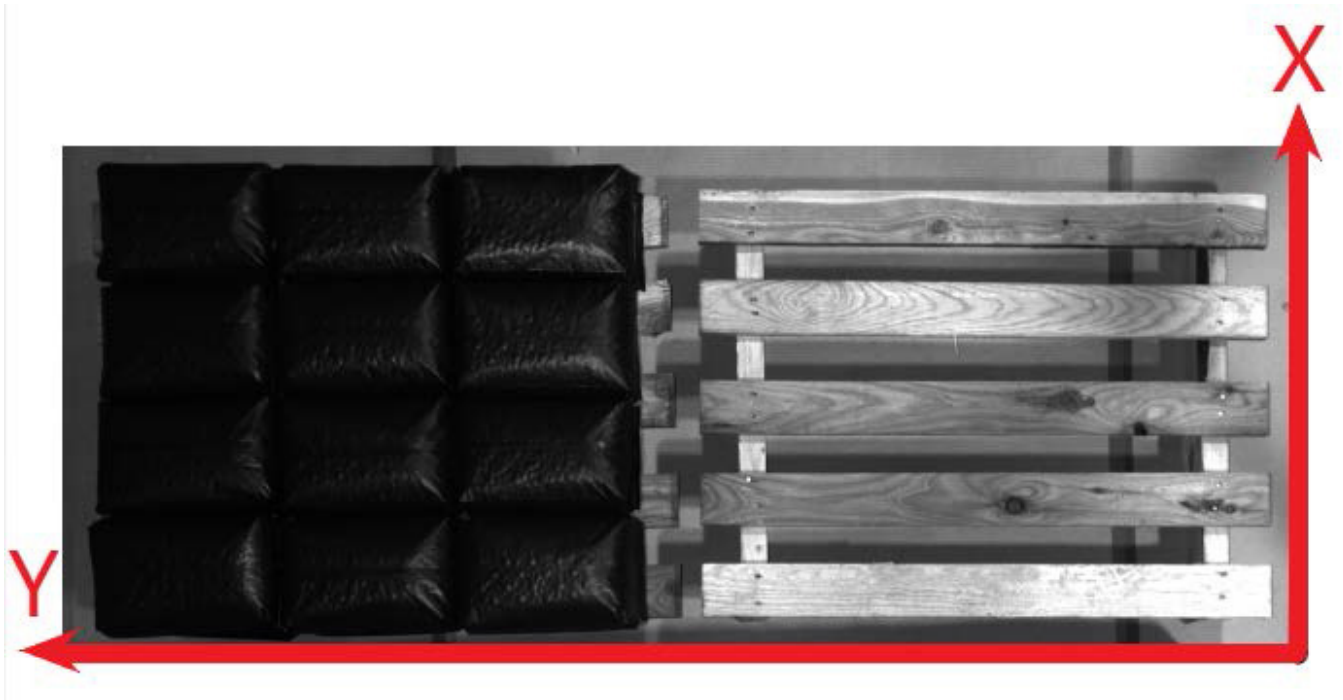
3.4. Zaprogramowanie paletyzacji

Ostatnim zadaniem jest nauczenie robota punktu odkładania detalu.

Pozycja tego punktu w przestrzeni powinna zmieniać się wraz z inkrementacją licznika pętli głównej. Do wyliczenia pozycji offsetu można skorzystać z przygotowanej funkcji **CALC_PLA-CE1**, której wyjściem jest rejestr pozycyjny **PR[14:current_position]**.

Program CALC_PLACE1 wylicza pozycję bazując na wartości przechowywanej w rejestrze R[10:current_place] i fizycznych gabarytach elementu.

Aby program działał prawidłowo w rejestrze R[10:current_place] należy przechowywać aktualny numer odkładanego detalu, należy więc każdorazowo inkrementować jego wartość. Ponieważ program CALC_PLACE1 zwiększa wartość pozycyjną rejestru PR[14:current_position] w osi X oraz osi Y, należy pamiętać o dodaniu punktu odkładania detali na skrajnej prawej stronie palety (patrzac od strony robota).



Rysunek 11 Zdjęcie palet wykonane kamerą 3DV

UWAGI

- Nie należy modyfikować nazw oraz wartości rejestrów: R[] – R[]
- Nie należy modyfikować nazw oraz wartości rejestrów pozycyjnych: PR[] – PR[]
- Nie należy modyfikować struktury programów: VACCUM_ON, VACUUM_OFF, CALC_PLACE1
- Nie należy modyfikować struktury procesów wizyjnych (z wyjątkiem instrukcji przedstawionej w punkcie 3.1.)
- Nie należy modyfikować TCP robota.
- Nie należy modyfikować Payload'u robota
- Testowanie programu należy przeprowadzać na prędkości OVERRIDE[100%] w trybie T1

4. Elementy stanowiska

Elementy składowe stanowiska pracują pod wysokim napięciem, niebezpiecznym dla zdrowia człowieka, podczas pracy należy zachować szczególną ostrożność

4.1. Kontroler Robota R-30iB Plus + iPendant

Kontroler odpowiada za sterowanie robota. Wykorzystany model dedykowany jest do robota M-20iD/25. W aplikacji kontroler steruje również procesami wizyjnymi i przetwarza sygnały

otrzymywane z czujnika podciśnienia zintegrowanego w chwytaku. Za pomocą panelu realizowane są wszystkie funkcje programowania i sterowania robotem. Kontroler zasilany jest z sieci trójfazowej 400V



Rysunek 12 Kontroler R-30iB Plus

4.2. Robot M-20iD/25

W tej aplikacji robot realizuje proces depaletyzacji.



Rysunek 14 Robot FANUC M-20iD/25

4.3. System wizyjny 3D VisionSensor 3DV/1600

System wizyjny 3D wyposażony w dwie kamery, projektor oraz oświetlacz LED. System dokonuje pomiarów geometrii trójwymiarowej i przedstawia użytkownikowi wyniki w postaci sześciu współrzędnych – X, Y, Z, W, P, R.



Rysunek 15 3D Vision Sensor 3DV/1600

4.4. Kawa

Detale wykorzystywane w aplikacji - aromatyczne i zamknięte w czarnych workach.



Rysunek 16 Detal

UWAGA!

Należy zachować szczególną ostrożność. Nie należy pić kawy!

5. Wykonanie poszczególnych etapów:

- Końcowa konfiguracja procesu wizyjnego – **1 punkt**,
- Stworzenie programu głównego i pobranie danych z procesu wizyjnego – **1 punkt**,
- Zaprogramowanie poboru zlokalizowanych detali - **1 punkt**,
- Zaprogramowanie paletyzacji – 1 punkt,

Czas wykonania zadania:

- Wykonanie zadanie w czasie poniżej 40 minut – **1 punkt**,
- Wykonanie zadania w czasie poniżej 35 minut – **2 punkty**,
- Wykonanie zadania w czasie poniżej 30 minut – **3 punkty**.

Realizacja celu głównego – 1 punkt.