

Systemy wizyjne w robotach przemysłowych na przykładzie FANUC iRVision*

Arkadiusz Mielczarek, Robert Muszyński[†]

Laboratorium Robotyki
Wydział Elektroniki
Politechnika Wrocławska

Spis treści

1	Cel ćwiczenia	1
2	Wymagania wstępne	2
3	System wizyjny iRVision	2
3.1	Połączenie	2
3.2	Procesy wizyjne	3
3.3	Nauka rozpoznawania detalu	4
3.4	Użycie systemu wizyjnego programie	5
3.4.1	VISION RUN_FIND '...'	5
3.4.2	VISION GET_OFFSET '...' VR[...] JMP LBL[...]	5
3.4.3	VOFFSET, VR[...]	6
4	Uwagi odnośnie wykonywanych zadań	6
5	Zadania do wykonania	6
6	Sprawozdanie	7

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie z możliwościami programowania robotów przemysłowych w oparciu o wizyjne sprzężenie zwrotne. System wizyjny używany w ćwiczeniu to FANUC iRVision dla robotów przemysłowych FANUC LR Mate 200iD.

W przypadku realizacji zadania w formie zdalnej należy je wykonać w środowisku ROBOGUIDE wykorzystując komórkę CELL_Viz umieszczoną na stronie kursu. Elementy instrukcji różniące się w przypadku stacjonarnego i zdalnego trybu zostały oznaczone [STAC] i [ZDAL].

*Ćwiczenie laboratoryjne przeznaczone do realizacji w ramach kursu Robotyka (3) – data ostatniej modyfikacji: 9 października 2020.

[†]Katedra Cybernetyki i Robotyki

2 Wymagania wstępne

Przed przystąpieniem do realizacji ćwiczenia należy:

- zapoznać się z treścią instrukcji *Instrukcja bezpieczeństwa i higieny pracy w Laboratorium Robotyki* [2],
- (FANUC) zapoznać się z treścią instrukcji *Obsługa i programowanie robota FANUC LR Mate 200iC/200iD* [3],
- przypomnieć podstawowe pojęcia związane z robotami manipulacyjnymi, w tym:
 - układ współrzędnych,
 - przestrzeń konfiguracyjna,
 - przestrzeń robocza,
 - łańcuch kinematyczny,
 - kinematyka prosta,
 - kinematyka odwrotna,
- zapoznać się z niniejszą instrukcją.
- [ZDAL] zapoznać się z instrukcją *Programowanie robota FANUC w środowisku symulacyjnym Roboguide* [1].

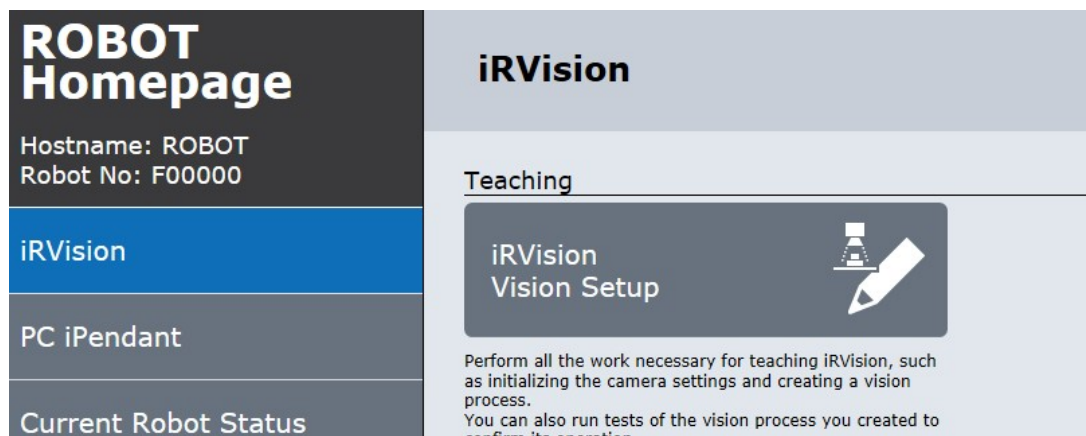
3 System wizyjny iRVision

iRVision jest komponentem robota pozwalającym na ekstrakcję i wykorzystanie informacji zawartych w obrazie z kamery. Interfejsem systemu iRVision jest przeglądarka internetowa.

3.1 Połączenie

Aby połączyć się z systemem iRVision należy:

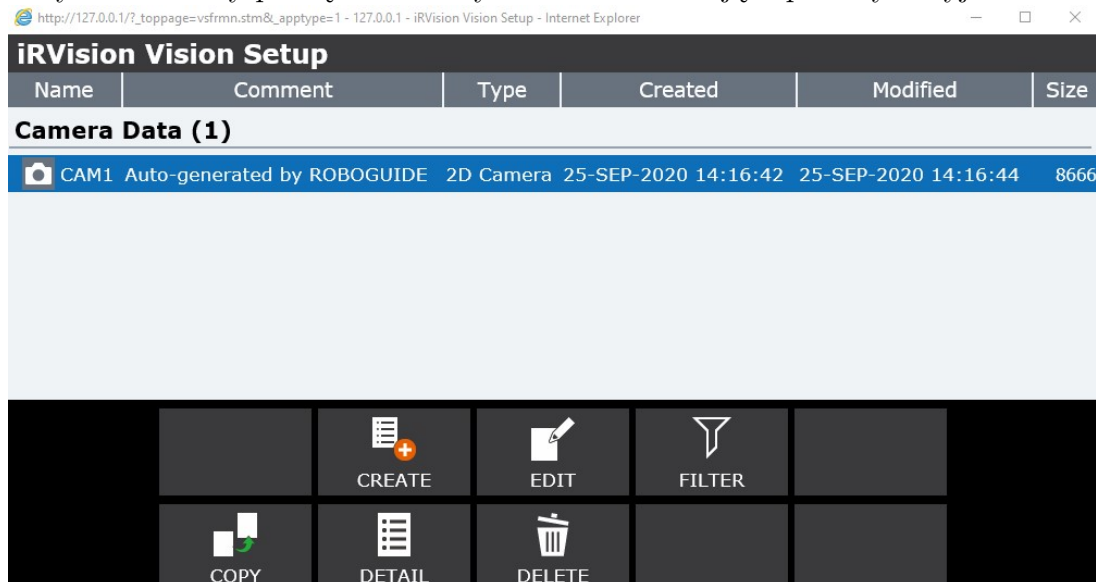
- [STAC] uruchomić przeglądarkę ... a następnie wpisać adres ...
- [ZDAL] w programie ROBOGUIDE wejść w menu **Robot** a następnie **Web Browser**. W lewej części przeglądarki wejść w iRVision.



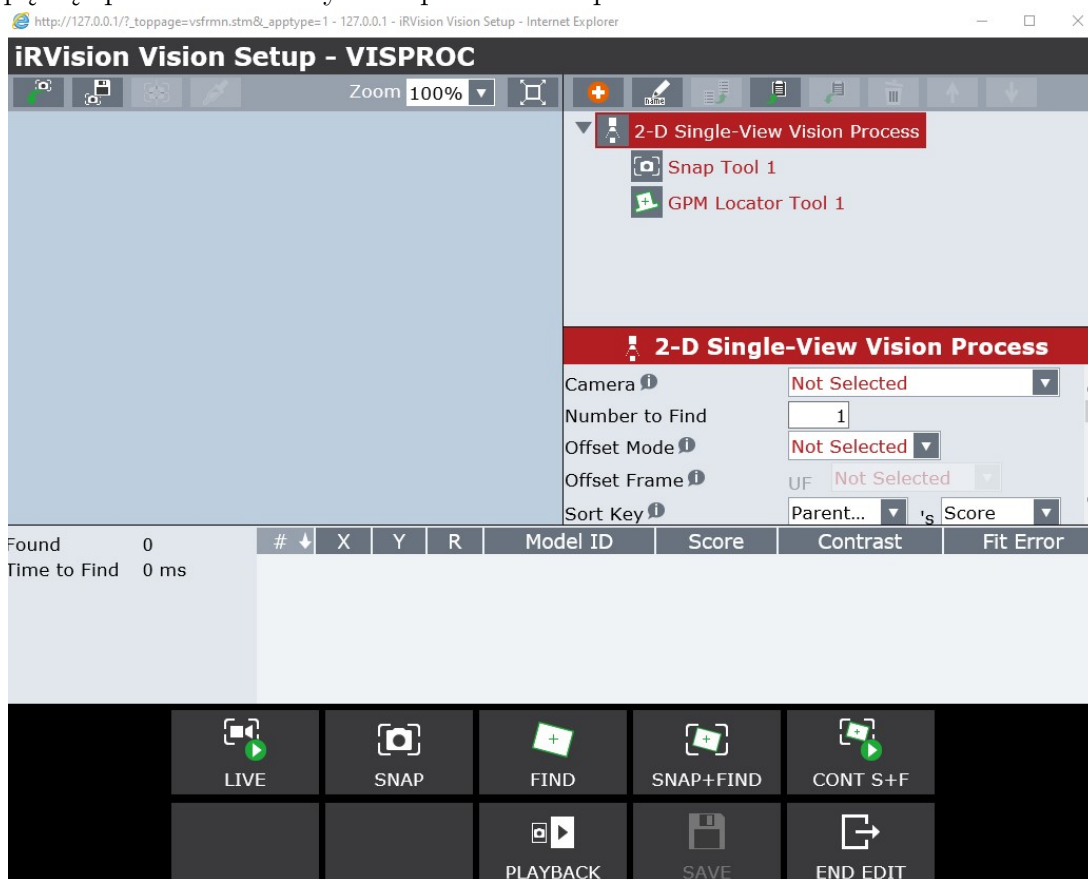
3.2 Procesy wizyjne

Proces wizyjny służy do pobierania pewnego typu informacji ze zdjęcia oraz zapisaniu go w odpowiednim rejestrze robota. Rodzajów dostępnych procesów wizyjnych jest sporo. Na potrzeby zadania odpowiednim procesem jest identyfikacja położenia detalu w środowisku.

W zakładce Teaching znajduje się iRVision Vision Setup. W tej podstronie znajdują się wszystkie kamery podłączone do systemu oraz istniejące procesy wizyjne.



Aby utworzyć nowy proces wizyjny należy nacisnąć **CREATE**. Następnie wpisać nazwę, wybrać typ **2-D Single-View Vision Process** i zatwierdzić. Klikając **EDIT** należy rozpuścić proces dostosowywana parametrów procesu.



Parametry jakie należy wybrać to:

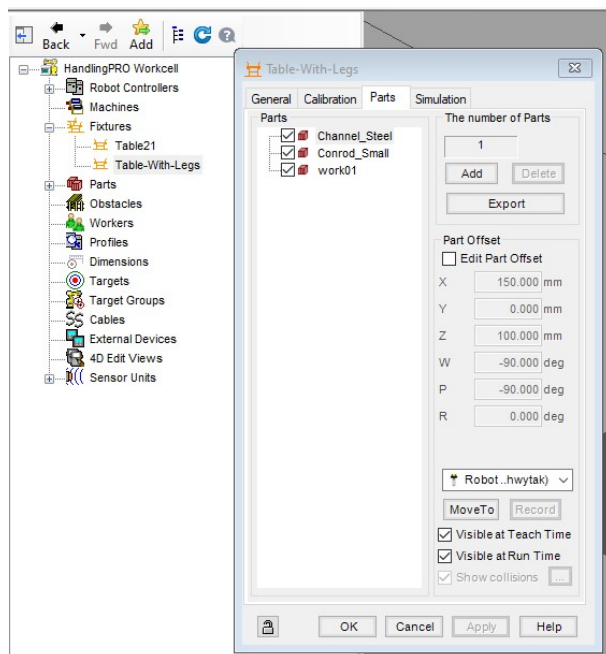
- Camera – jedyna istniejąca
- Offset Mode – Fixed Frame Offset. Opcja ta jest używana w przypadku określania położenia detalu przed podniesieniem.
- Offset Frame – User Frame w jakim chcemy dostać lokalizację.
- Part Z Height – wysokość detalu.
- Ref. Pos. Status – położenie "bazowe" detalu. Będzie to położenie wobec którego proces wizyjny będzie zwracał lokalizację. Aby wybrać położenie bazowe należy najpierw zdefiniować lokalizator GPM Locator Tool 1. Po zdefiniowaniu należy kliknąć SNAP+FIND (będąc w 2-D Single-View Vision Process, nie w GPM Locator Tool) a następnie wybrać zlokalizowany detal.

3.3 Nauka rozpoznawania detalu

Do rozpoznawania detalu służy GPM Locator Tool. Aby nauczyć system nowego detalu należy:

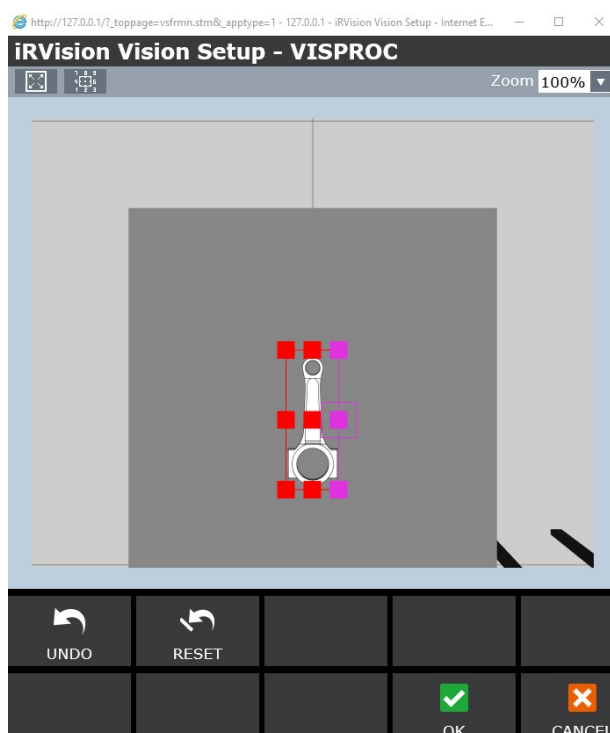
1. umieścić ramię robota poza obszarem kamery.
2. umieścić w wybranym miejscu bazowym detal którego chcemy nauczyć system, inne usunąć z obszaru kamery.

[ZDAL] Aby wykonać to w symulatorze należy rozwinąć listę Fixtures w drzewie projektu po lewej stronie. Kliknąć prawym przyciskiem na Table-With-Legs i wybrać Properties. Wejść w zakładkę Parts i odznaczyć opcje Visible at Teach Time oraz Visible at Run Time dla detali których obecnie nie chcemy na stole.



3. przejść do zakładki GPM Locator Tool.
4. zrobić zdjęcie kamerą za pomocą przycisku SNAP.

5. nacisnąć przycisk **Teach**. Zaznaczyć detal i zaakceptować.



6. Można ręcznie zedytować powstałą maskę detalu poprzez **GEdit**.
7. Opcjami **Orientation**, **Scale** i **Aspect** można określić wykrywanie detali obróconych o dany kąt, przeskalowanych (przybliżonych, oddalonych) oraz zniekształconych (np. poprzez nieznaczne obrócenie w innej osi niż Z lub zniekształcenie przez soczewkę kamery).
8. Po ustawieniu pożądanych opcji należy kliknąć **SAVE**.
9. Mając zdefiniowany detal można dokończyć konfigurację procesu wizyjnego o znalezienie bazowego położenia.

3.4 Użycie systemu wizyjnego programie

Poniżej opisano trzy funkcje możliwe do wywołania z poziomu programu.

3.4.1 VISION RUN_FIND '...'

Funkcja jest do znalezienia w **INST** → **VISION**. Argumentem jest proces wizyjny. Służy do wykonania zdjęcia przez kamerę dla danego procesu.

3.4.2 VISION GET_OFFSET '...' VR[...] JMP LBL[...]

Funkcja jest do znalezienia w **INST** → **VISION**. Argumentami są:

'...' proces wizyjny,

VR[...] rejestr wizyjny do którego należy wpisać wynik działania,

LBL[...] etykieta (label) do której program ma skoczyć jeśli nie udało się otrzymać wyniku.

Funkcja służy do określania przesunięcia względem pozycji bazowej elementu rozpoznawanego przez proces wizyjny.

3.4.3 VOFFSET, VR[...]

Opcja związana z instrukcją ruchu POINT. Argumentem jest numer rejestru wizyjnego. Do punktu określonego w instrukcji ruchu dodaje offset odczytany z rejestru wizyjnego. Sposob użycia i działanie analogiczne jak dla zwykłej opcji offsetu instrukcji ruchu z rejestru pozycyjnego PR.

4 Uwagi odnośnie wykonywanych zadań

- manipulatorem należy poruszać z małymi prędkościami,
- w każdym programie należy na początku określić układy: narzędzia (UTOOL_NUM=...) i użytkownika (UFRAME_NUM=...). Układy powinny się pokrywać z aktualnymi (SHIFT + COORD). Na potrzeby zadania zostało zdefiniowane narzędzie (UTOOL 8) oraz układ współrzędnych użytkownika (UFRAME 8).
- punkty zapamiętane za pomocą instrukcji Point pamiętają do jakiego User Frame się odnoszą. Inaczej jest w przypadku rejestrów pozycyjnych. W przypadku użycia takich w programie, należy upewnić się, że program jest uruchamiany z odpowiednim User Frame.
- na potrzeby zadania zostały zdefiniowane podprogramy umożliwiające chwytanie (GRASP) i puszczanie (RELEASE) elementu. [ZDAL] takie podprogramy są zdefiniowane dla każdego detalu osobno.
- [ZDAL] W komórce CELL_Viz na stole odkładczym detale są widoczne w położeniu docelowym.

5 Zadania do wykonania

1. Na podstawie swobodnego ruchu manipulatorem określić w przybliżeniu położenie i orientację używanego układu współrzędnych użytkownika (User Frame).
2. Zdefiniować proces wizyjny pozwalający na zidentyfikowanie oraz lokalizację detalu wskazanego przez prowadzącego.
3. Napisać program pozwalający na odkładanie detalu losowo położonego na blacie na określone miejsce.
4. Uogólnić powyższy program na przypadek z kilkoma różnymi detalami, odkładanymi w ściśle określonej kolejności.

Powyższa lista zadań jest propozycją, ostateczny zakres zadań do wykonania zostanie podany przez prowadzącego na zajęciach.

6 Sprawozdanie

Sprawozdanie z przebiegu ćwiczenia powinno zawierać:

- Imię i nazwisko autora, numer i termin grupy, skład grupy, temat ćwiczenia, datę wykonania ćwiczenia, datę dostarczenia sprawozdania.
- Cel ćwiczenia.
- Opis zrealizowanych zadań:
 - co dane zadanie miało na celu,
 - sposób przeprowadzenia zadania,
 - opracowanie otrzymanych wyników,
 - uzyskany rezultat – wnioski.
- Wnioski końcowe.

Literatura

- [1] Mirela Kaczmarek. *Programowanie robota FANUC w środowisku symulacyjnym Roboguide*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Programowanie robota FANUC w środowisku symulacyjnym Roboguide, 2002.
- [2] Marek Wnuk. *Zasady bezpiecznej pracy w Pracowni Robotyki - s. 010 C-3*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Zasady bezpiecznej pracy w Pracowni Robotyki - s. 010 C-3.
- [3] Katarzyna Zadarnowska, Jacek Jagodziński, Arkadiusz Mielczarek. *Obsługa i programowanie robota FANUC LR Mate 200iC*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, Obsługa i programowanie robota FANUC LR Mate 200iC, 2018.