Konstrukcja manipulatora 3-osiowego sterowanego za pomocą gestów

Łukasz Sawicki

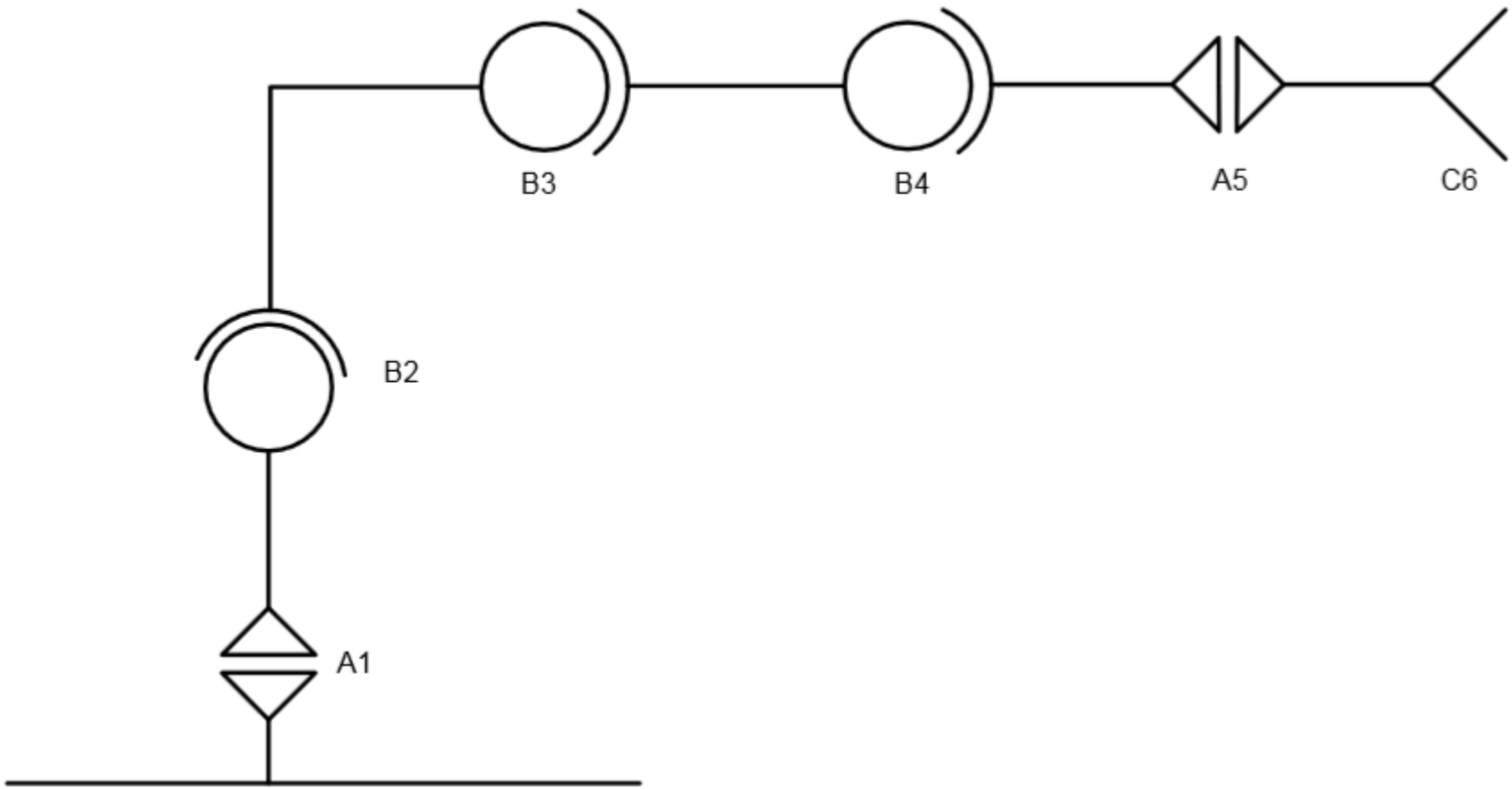
Zakres pracy:

- Rozpoznanie w zakresie wykorzystywanych manipulatorów
- Rozpoznanie w zakresie rozpoznawania gestów
- Zaprojektowanie manipulatora
- Opracowanie oprogramowania do sterowania manipulatorem
- Badania doświadczalne

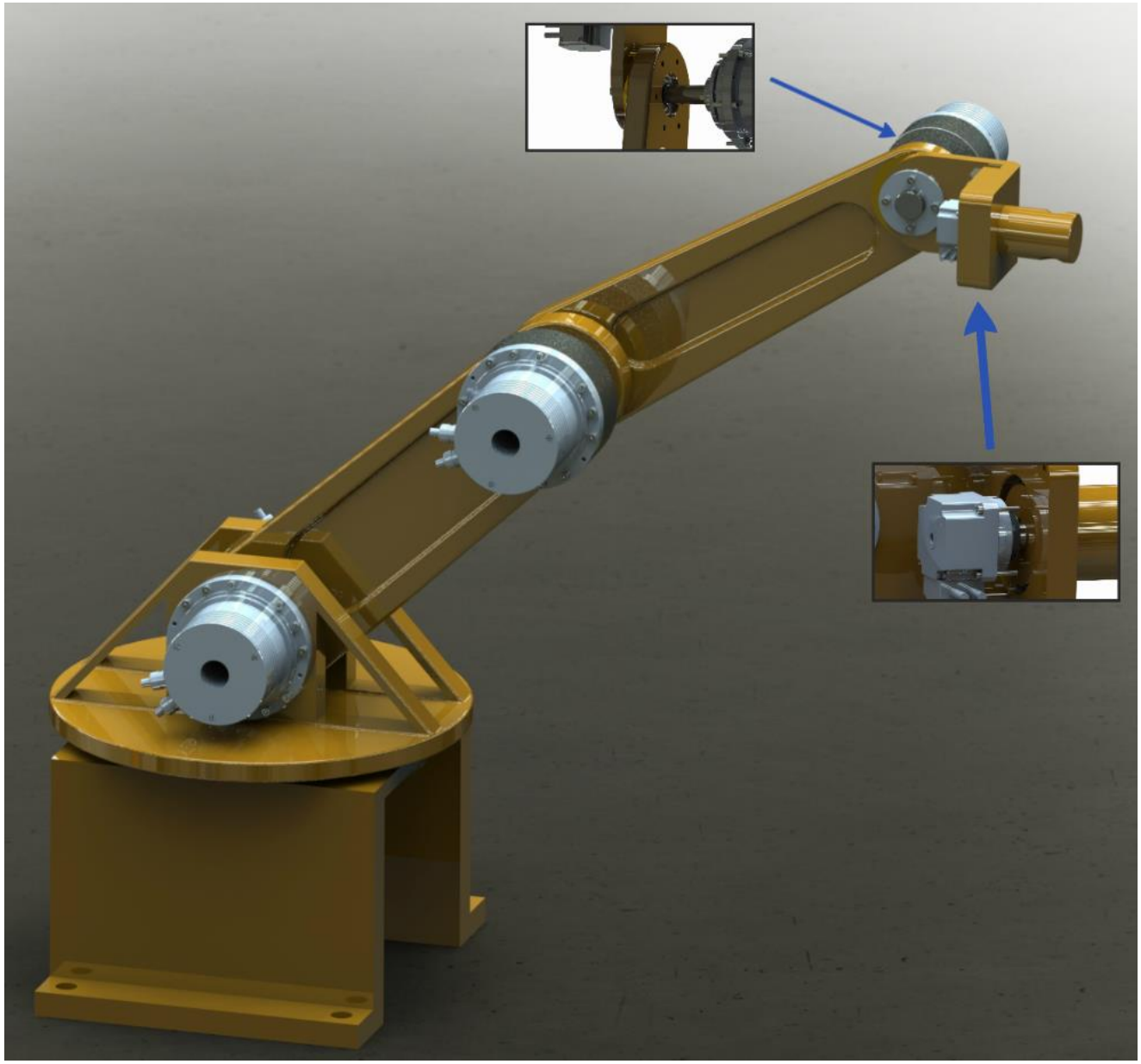
Dane wyjściowe:

- Nośność: 0,5 kg
- Rodzaj napędu: elektryczny
- Struktura kinematyczna: antropomorficzna
- Liczba stopni swobody: 5
- Kamera głębokości o rozdzielczości 300x200

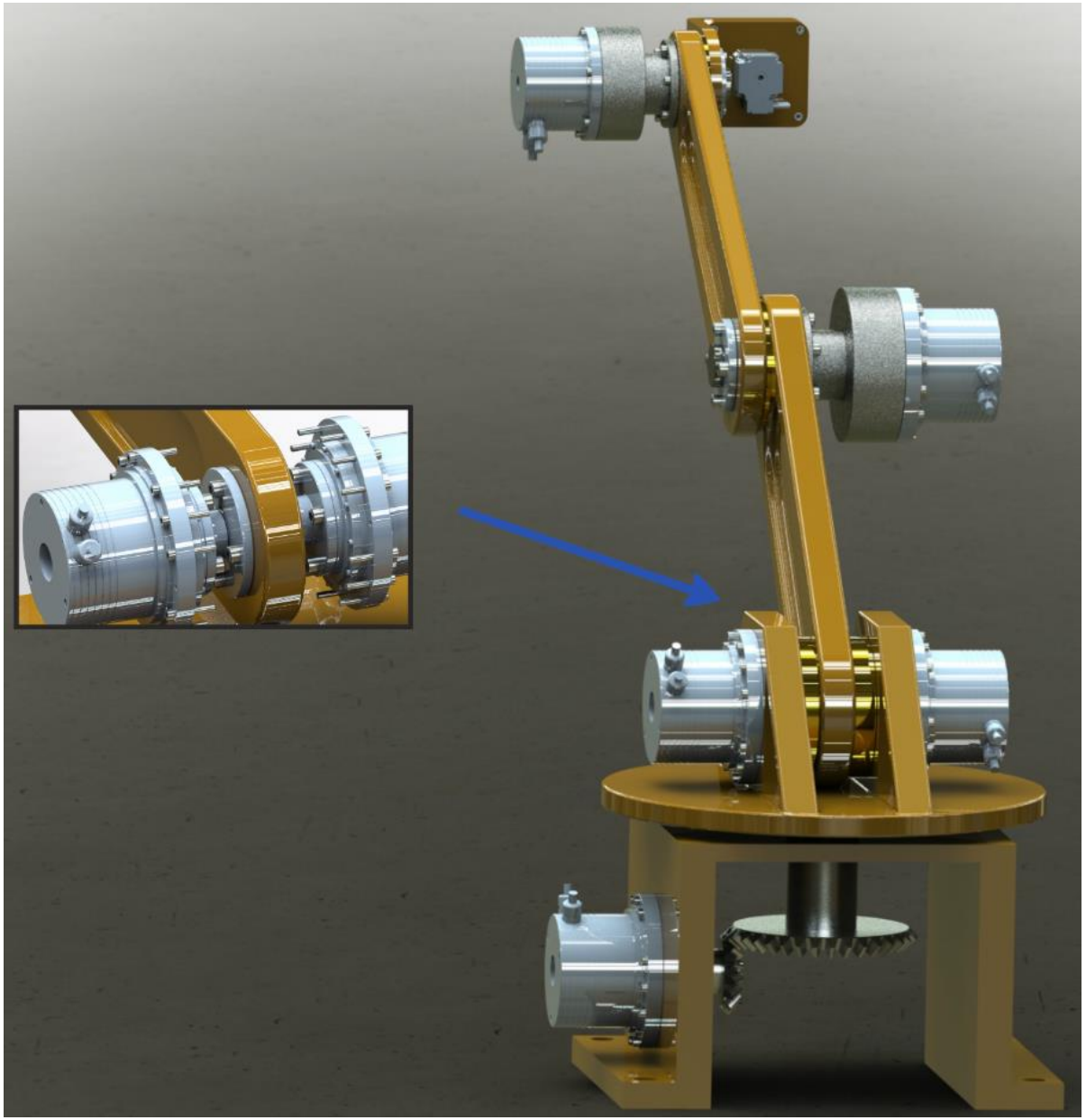
Struktura kinematyczna zaprojektowanego manipulatora



Projekt manipulatora

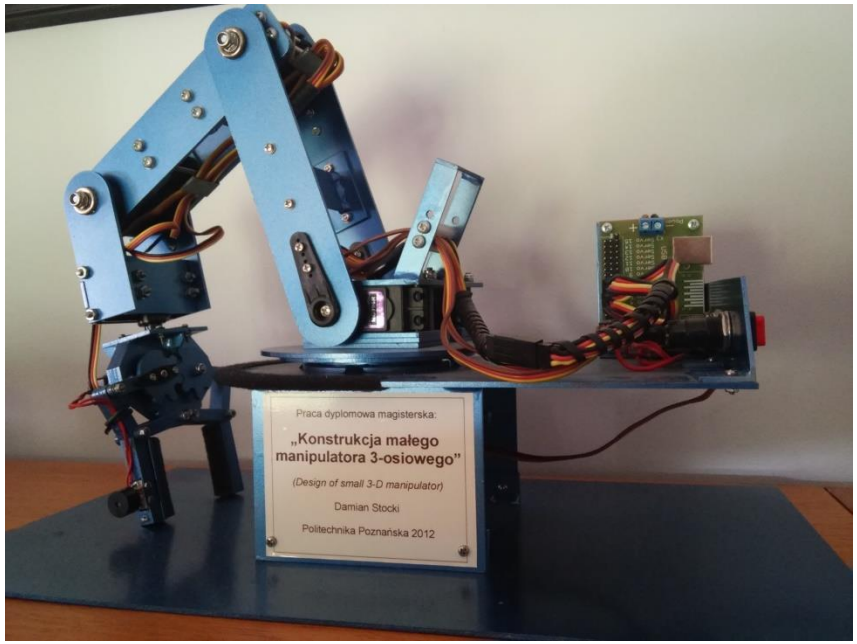


Projekt manipulatora – c.d.

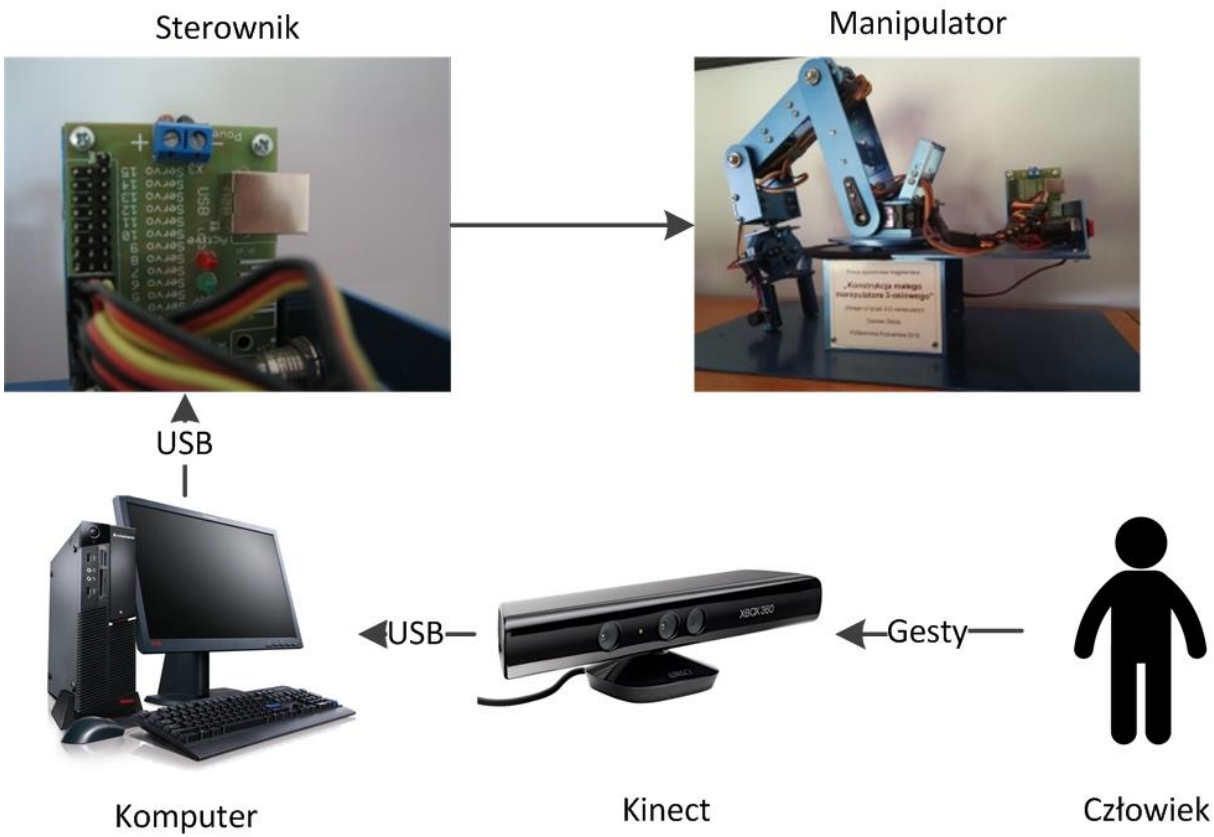


Mały 3-osiowy manipulator

- Manipulator został wypożyczony z Zakładu Urządzeń Mechatronicznych
- 5 stopni swobody
- zakres ruchów - 300mm
- udźwig w kiści - 0,05kg
- napędy – serwomechanizmy modelarskie



Schemat blokowy sterowania



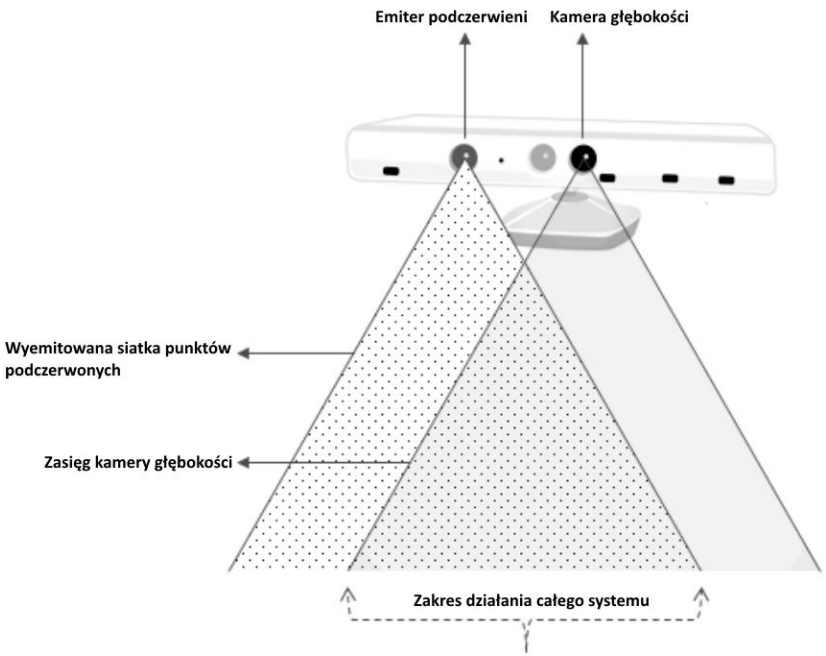
OpenCV

- Całkowicie darmowa biblioteka do obróbki obrazu do celów zarówno akademickich jak i komercyjnych
- Umożliwia programowanie m.in. w C, C++, Python
- Zawiera funkcje, które mogą zostać użyte w aplikacjach do rozpoznawanie twarzy, gestów, ruchu samochodowego i wielu innych.

Algorytm programu do sterowania manipulatorem

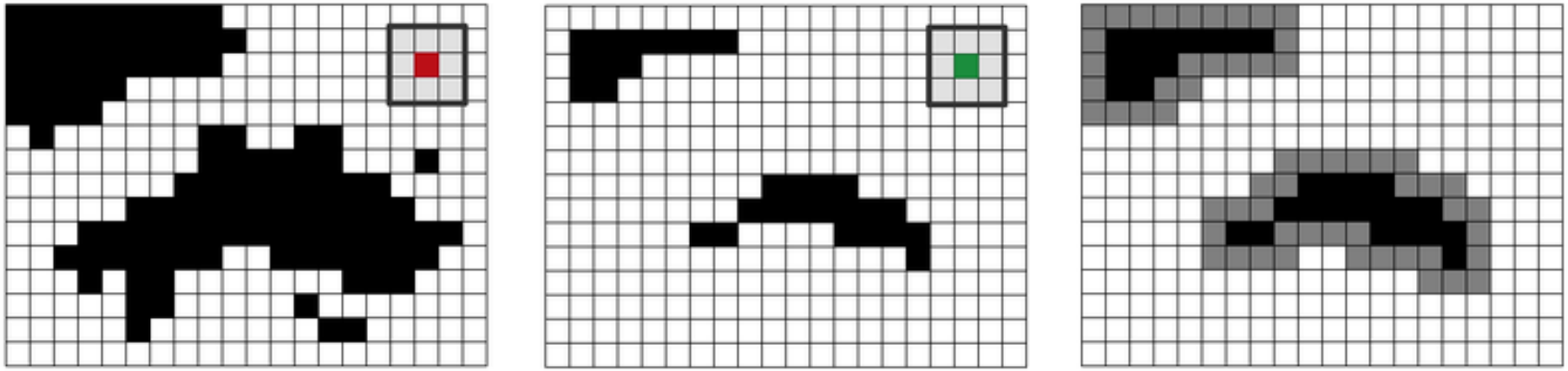
Przechwytywanie obrazu głębi

- Do przechwytywania obrazu wykorzystano urządzenie firmy Microsoft o nazwie Kinect
- System składający się z emitera podczerwieni oraz kamery głębokości pozwala na otrzymywanie trójwymiarowego obrazu



Przechwycony obraz z Kinect’a

Od szumienie obrazu



Erozja + Dylatacja = Otwarcie morfologiczne

Progowanie

Próg został zaimplementowany w programie z wykorzystaniem funkcji OpenCV o nazwie threshold(). Jest kilka rodzajów progów, do tego programu został wybrany próg o nazwie „Threshold to Zero, Inverted”.

$$\mathrm{dst}(x,y) = \begin{cases} 0 & \text{if } \mathrm{src}(x,y) > \text{próg} \\ \mathrm{src}(x,y) & \text{if } \mathrm{src}(x,y) \le \text{próg} \end{cases}$$

gdzie:

x, y – współrzędne poszczególnych pikseli,

src(x,y) – funkcja źródłowa oznaczają wartość skali szarości w punkcie x, y,

dst(x,y) – funkcja wyjściowa.

Wyszukiwanie największego konturu obrazu po progowaniu

Po wyszukaniu konturu (obrys lewej ręki):

Przed wyszukaniem konturu:

Obwiednia (ang. convex hull)

Po znalezieniu konturu znajdowana jest jego obwiednia (ang. convex hull). Do tego jest wykorzystana funkcja OpenCV o nazwie convexHull().

Wykorzystuje ona algorytm Sklansky'iego z 1982 roku o złożoności obliczeniowej O(N logN), czyli dość szybki algorytm, który można swobodnie zaimplementować w programie czasu rzeczywistego.

Momenty obrazu

$$m_{ij} = \sum_{x,y} \left(array(x,y) * x^j * y^i \right)$$

gdzie:

m_{ij} – oznacza moment obrazu ij,

array(x,y) – tablica współrzędnych x,y obwiedni (w tym przypadku).

$$x_c = \frac{m_{10}}{m_{00}}, \; y_c = \frac{m_{01}}{m_{00}}$$

gdzie:

x_c, y_c – współrzędne środka ciężkości obwiedni,

m_{00} – oznacza moment obrazu 00, czyli długość obwiedni,

m_{01} – oznacza moment obrazu 01, czyli wartości pikseli w osi Y,

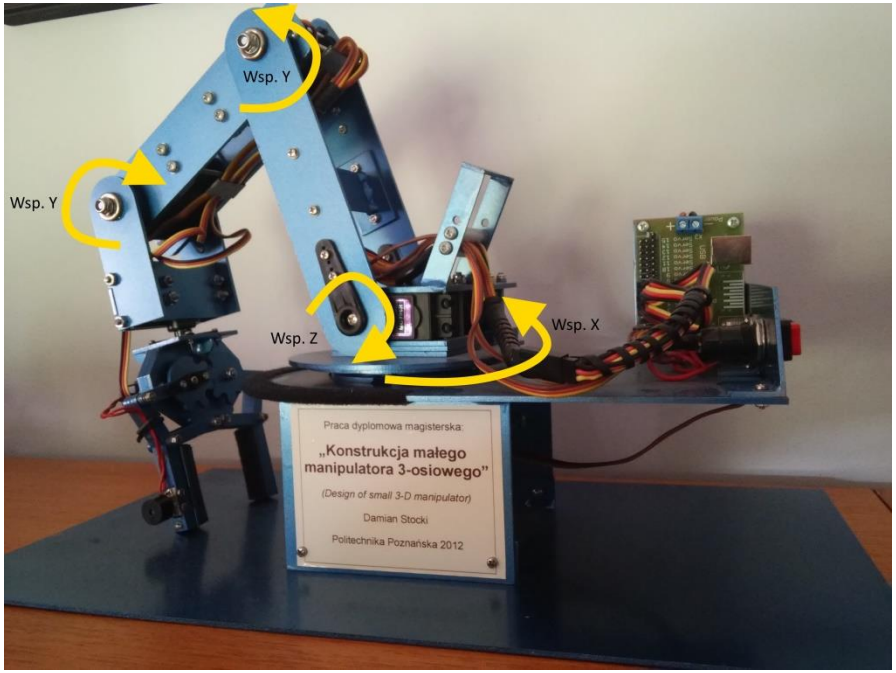
m_{10} – oznacza moment obrazu 10, czyli wartości pikseli w osi X.

Na podstawie otrzymanej obwiedni są obliczane jej momenty. Momenty służą do definiowania danego obrazu (lub jego części) poprzez sumowanie wartości pikseli w nim zawartych.

Momenty są jedynie środkiem do celu, czyli obliczeniu środka obwiedni ręki, zwanego również jej środkiem ciężkości xc, yc.

Wartości współrzędnych tego punktu będą śledzone i na ich podstawie będzie sterowany manipulator.

Sterowanie manipulatorem



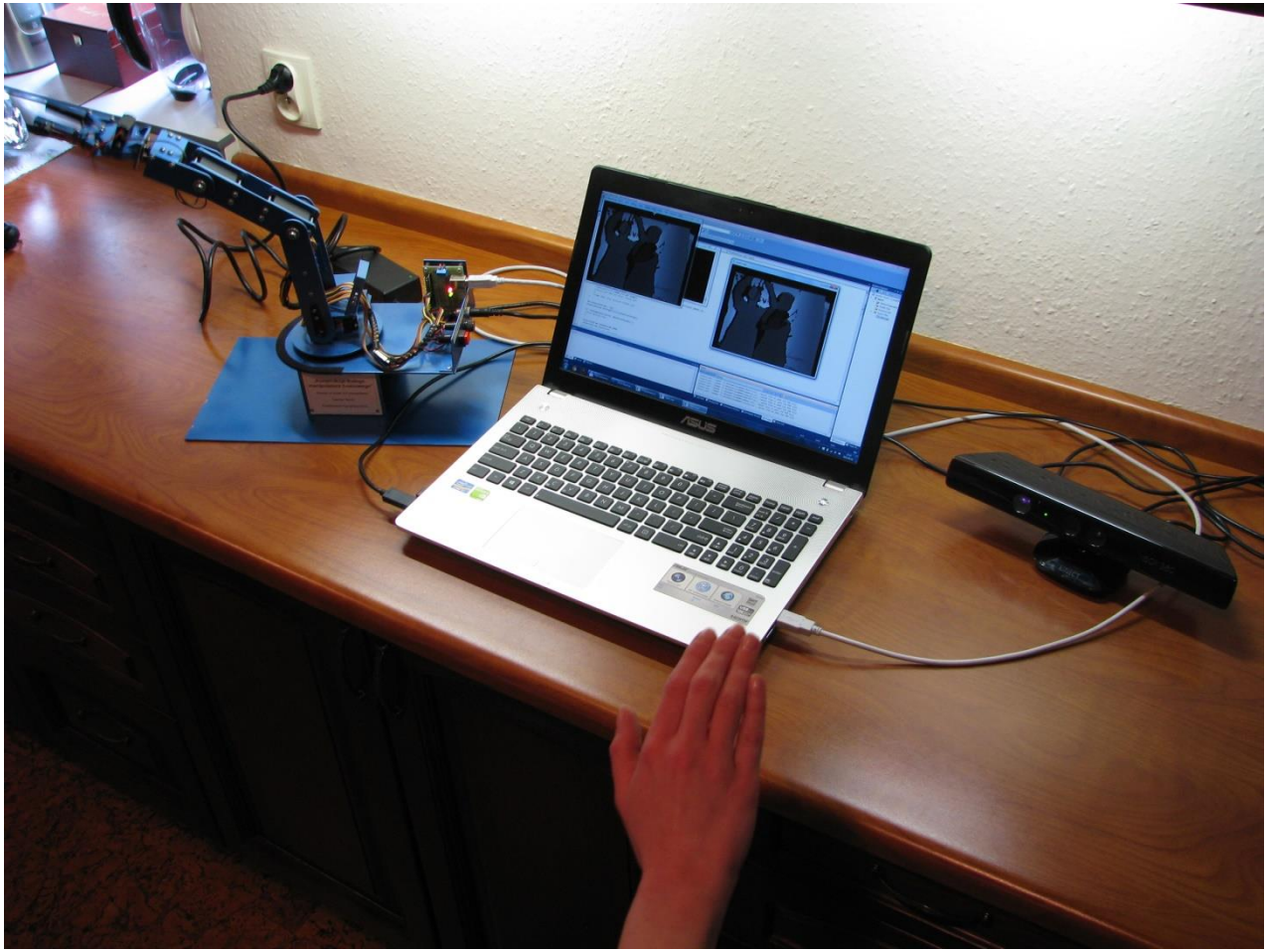
$$Xc = \frac{x_c}{Szerokość\ obrazu\ (640\ px)} * Rozdzielczość\ serva(16000)$$

$$Yc = \frac{y_c}{Wysokość\ obrazu\ (480\ px)} * Rozdzielczość\ serva(16000)$$

$$Zc = \frac{z_c}{Próg\ w\ skali\ szarości(45)} * Rozdzielczość\ serva(16000)$$

gdzie:

x_c, y_c, z_c – współrzędne środka ciężkości obwiedni,

Xc, Yc, Zc – wartości wysyłane do sterownika manipulatora.

Stanowisko sterowania oraz manipulator