Rozpoznawanie wypowiedzi migowych w systemie wizyjnym

Marian Wysocki, mwysocki@kia.prz.edu.pl
Politechnika Rzeszowska
Katedra Informatyki i Automatyki
Zespół wizji komputerowej i optymalizacji

W wykładzie wykorzystano wyniki prac prowadzonych przez autora wraz z
T. Kapuścińskim, J. Marnik, M. Oszustem, D. Warchołem

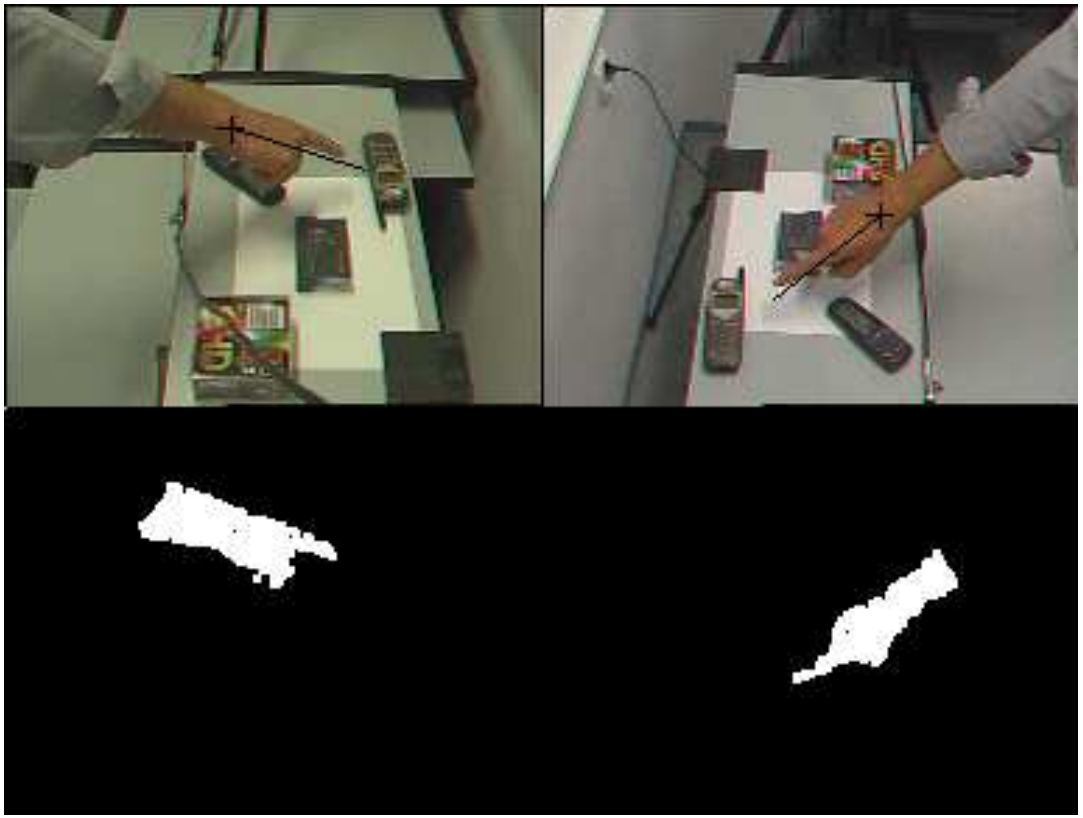
Wprowadzenie

- W bezpośredniej komunikacji między ludźmi około 70 % informacji jest przekazywane za pośrednictwem gestów, układu ciała, kontaktu wzrokowego, ubioru itp.
- Wykorzystanie gestów rąk i ciała stanowi atrakcyjną alternatywę dla uciążliwych interfejsów używanych w interakcji człowieka z komputerem.
- Ręka jest najbardziej funkcjonalną częścią ciała.

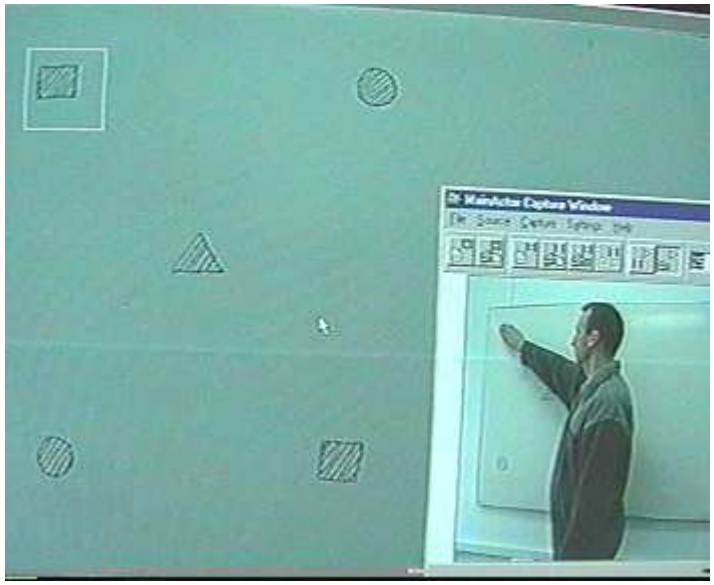
Przykładowe zastosowania rozpoznawania gestów wykonywanych rękami

- Interakcja człowiek-robot (np. instruowanie robota jak uchwycić przedmiot, wydawanie poleceń robotowi mobilnemu)
- Sterowanie urządzeniami w inteligentnym mieszkaniu (zwłaszcza jako pomoc dla osób starszych i niepełnosprawnych)
- Operowanie wirtualnymi obiektami
- Interpretacja wypowiedzi w języku migowym

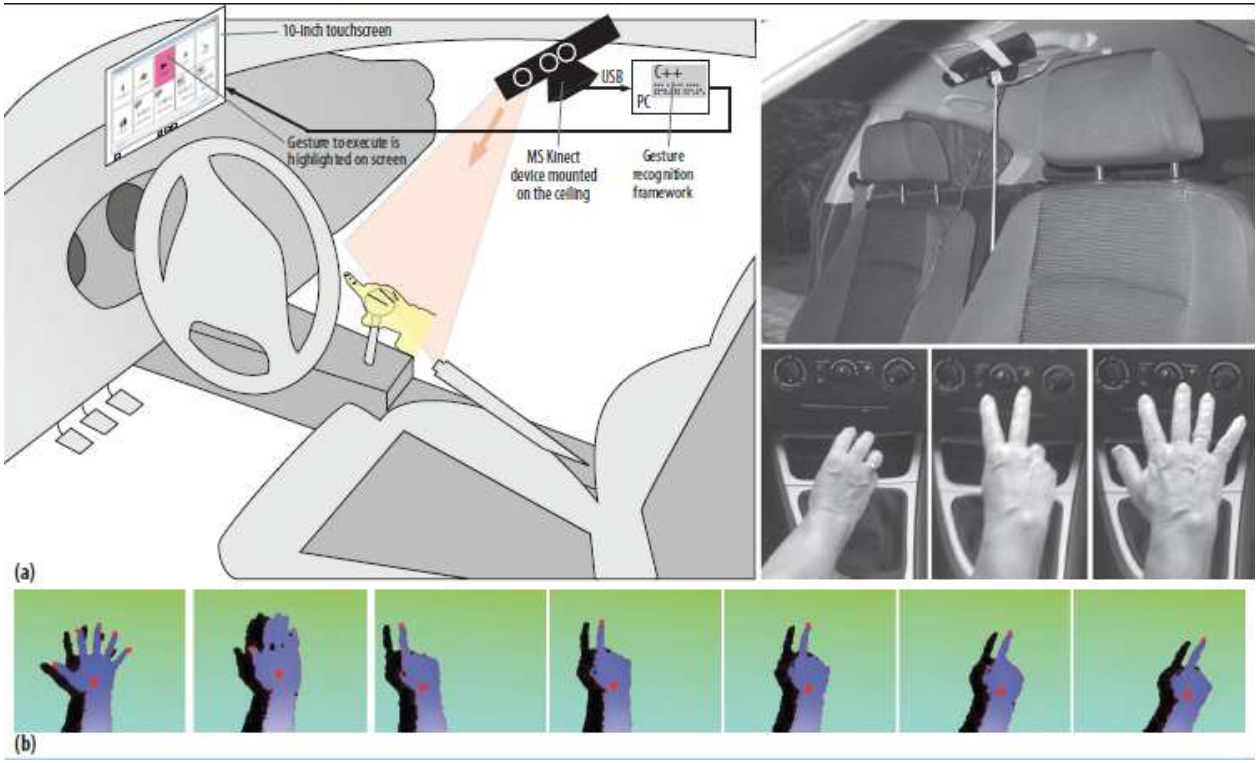
Gest wskazujący



Przesuwanie wirtualnych obiektów



Gesty w samochodzie



a) Sensor kinect zamontowany w dachu obserwuje gesty ręki ułożonej na dźwigni zmiany biegów.

b) Do sterowania panelem dotykowym kierowca wykorzystuje proste gesty statyczne (dwa pierwsze od lewej) i dynamiczne (np. przesuwanie palca wskazującego – pięć kolejnych obrazów).

Źródło: IEEE Computer, April 2012

Inne przykłady

Układ rąk obrazuje książkę. Tekst jest wyświetlany na dłoni za pomocą miniaturowego rzutnika. Zmiana strony następuje przez przełożenie odpowiedniej ręki

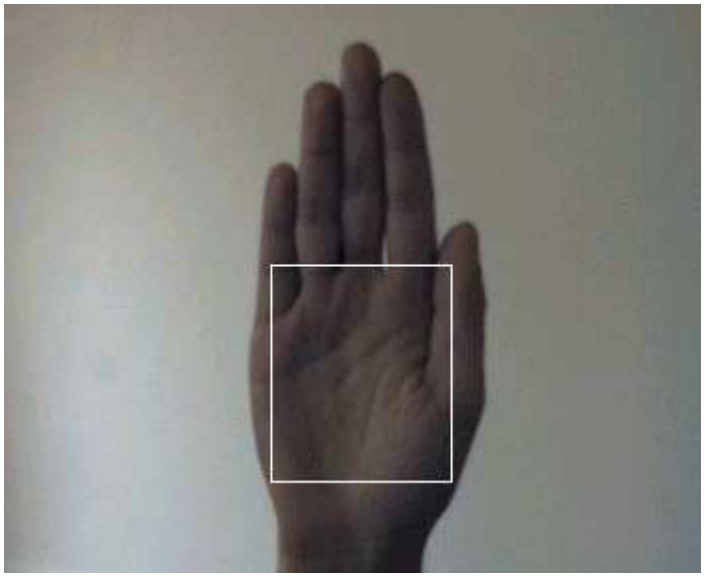
Źródło: *C. Harrison i in. TEI 2012,* Feb 19-22, 2012, Kingston, Ontario, Canada, Copyright 2012 ACM 978-1-4503-0541-9/11/08-09

Rozpoznawanie wypowiedzi języka migowego

Plan

- Detekcja skóry ludzkiej
- Rozpoznawanie kształtu dłoni
- Rozpoznawanie gestów dynamicznych
- Zastosowanie kamer ToF i Kinect
- Wykorzystanie jednostek mniejszych niż słowa.

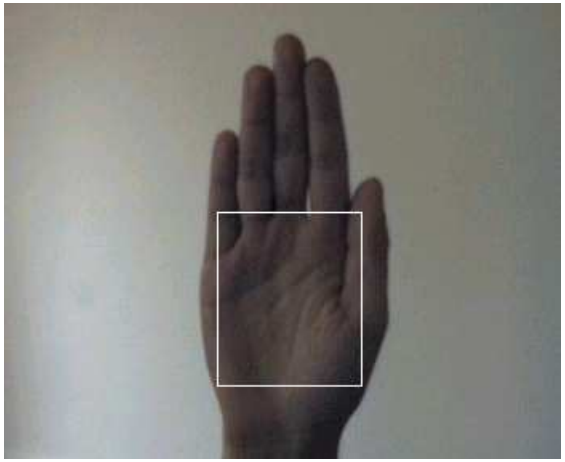
Model koloru skóry



- Znormalizowana przestrzeń RGB
  L=R+G+B
  r=R/L, g=G/L, b=B/L
- 2D histogram h(r,g)
- Aproksymacja rozkładem Gaussa

$$h(r,g) \cong G(r,g) = G(\xi) = \frac{1}{2\pi\sqrt{\det(C)}} e^{-\frac{1}{2}(\xi-\mu)^T C^{-1}(\xi-\mu)}$$

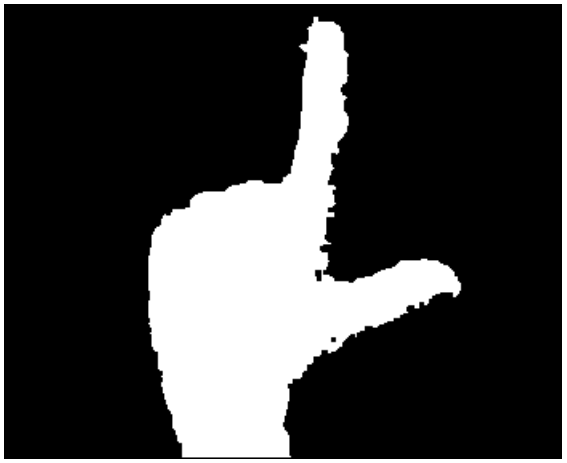
Wydzielanie dłoni



model

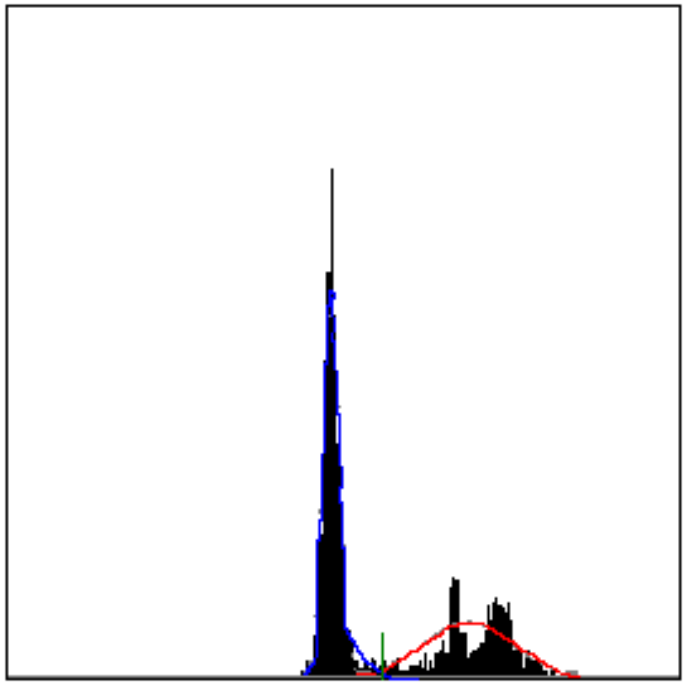
obraz wejściowy

obraz prawdopodobieństwa



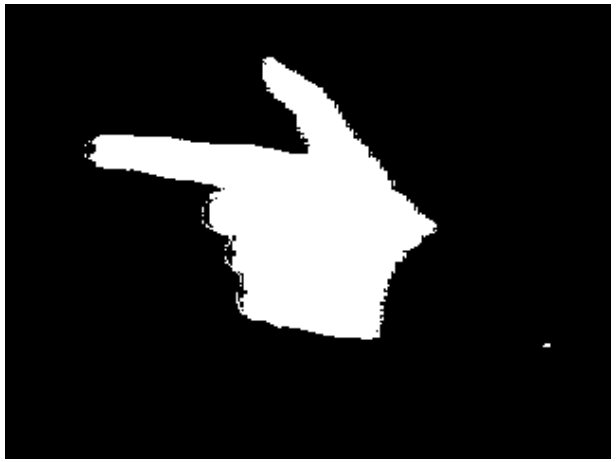
wydzielona dłoń

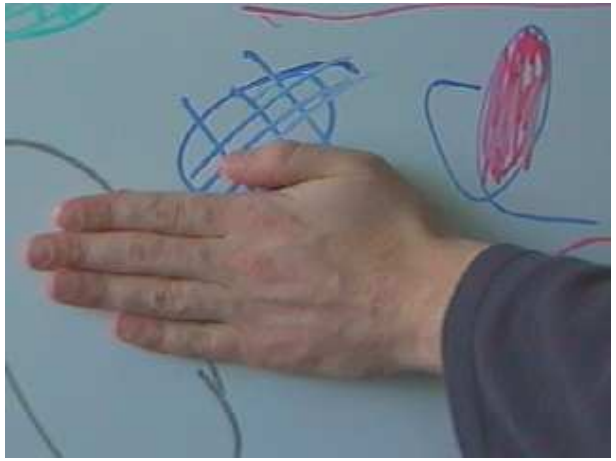
Adaptacyjne progowanie

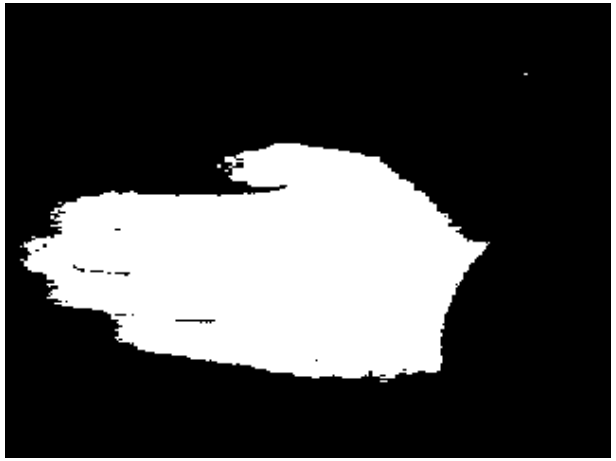


Próg jest wyznaczany na podstawie aproksymacji histogramu za pomocą dwóch rozkładów normalnych

Przykładowe rezultaty segmentacji







obraz wejściowy

wynik

Rozpoznawanie kształtu dłoni

Niektóre metody znane z literatury

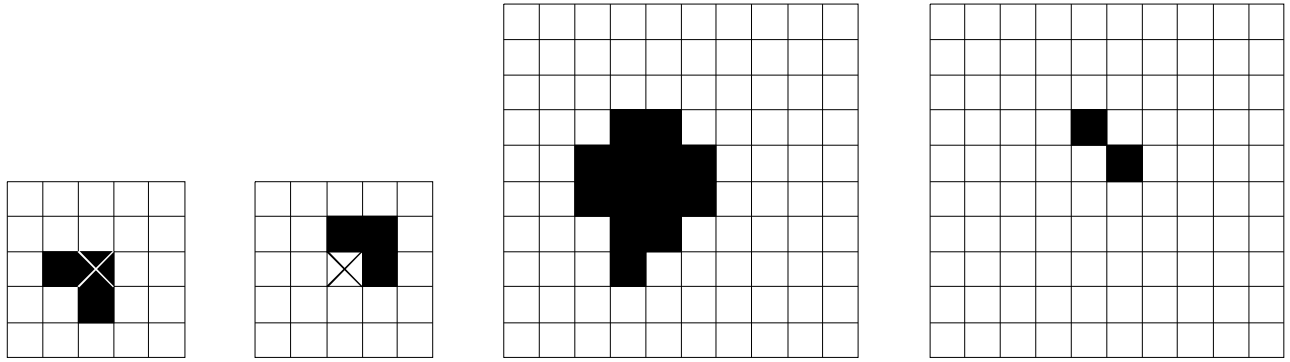
- Histogramy orientacji [Frieman, Roth 1995]
- Momenty geometryczne [Hunter et al. 1995]
- Elastyczne grafy [Triesch, v.d. Malsburg 2001]
- Morfologia matematyczna [Marnik, Wysocki 2004]

Transformacja trafi-nie trafi

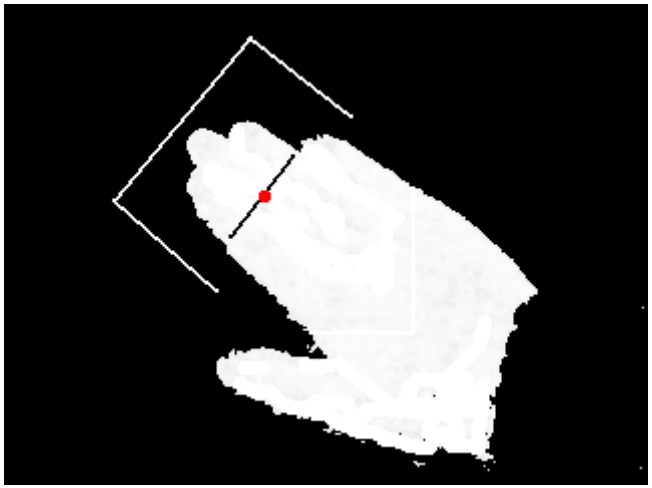
Element strukturalny *B* = (*B1, B2*) składa się z dwóch rozłącznych elementów *B1* and *B2,* z wyróżnionym punktem odniesienia (oznaczonym krzyżykiem).

Ten punkt jest przesuwany po obrazie.

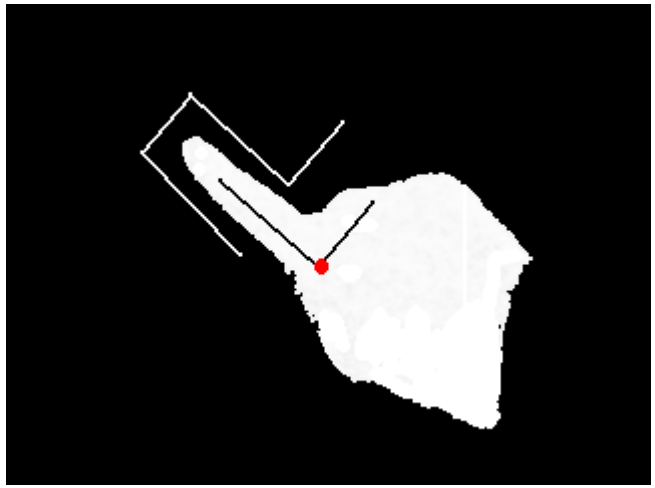
Punkt ***p*** należy do zbioru stanowiącego wynik transformacji, jeśli część *B1* zawiera się w obiekcie, a część *B2* w tle.



Klasyfikacja prostych kształtów



przesuń

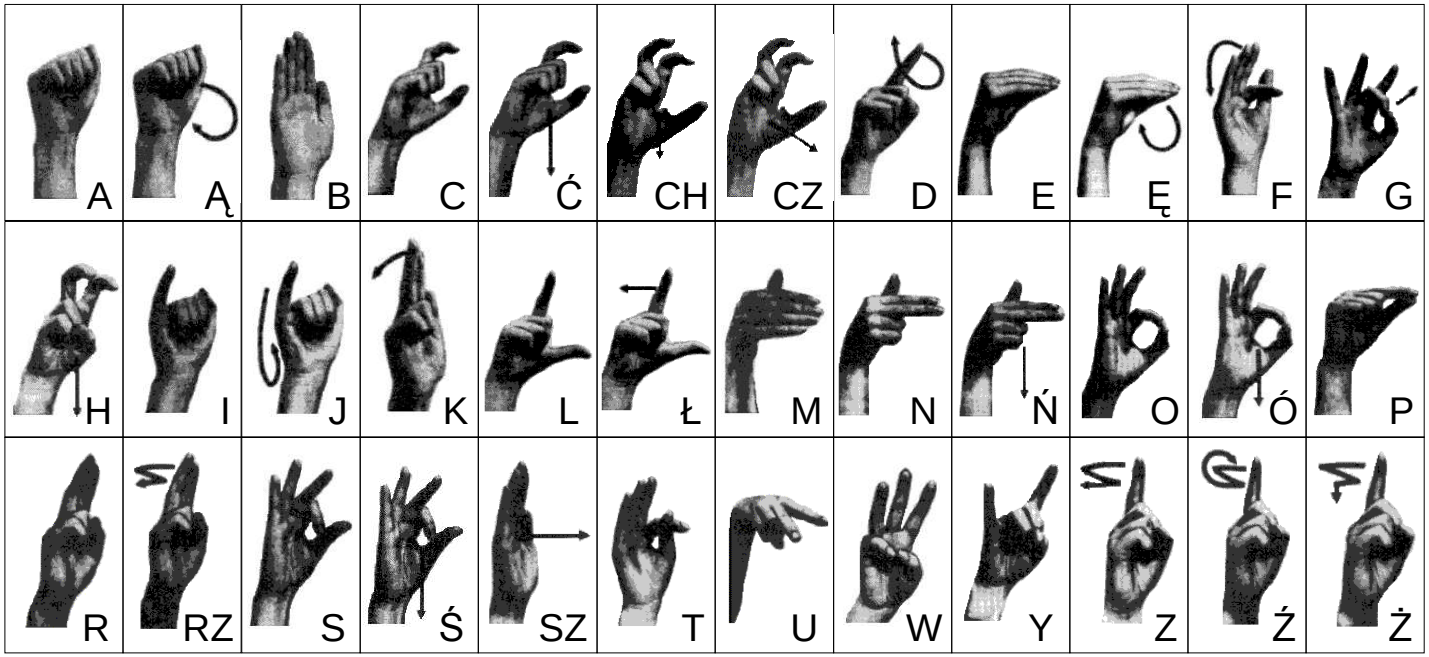


rysuj/pisz

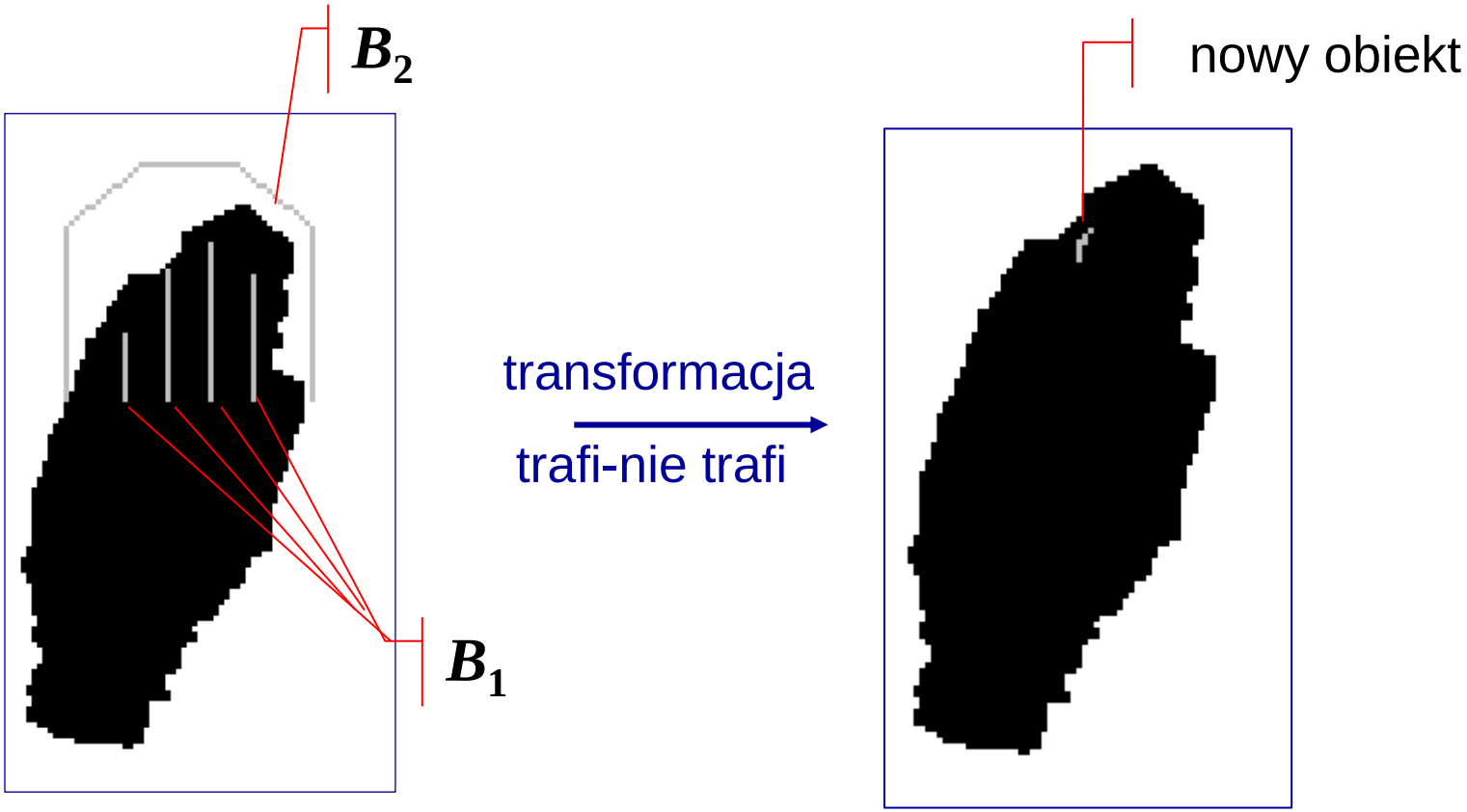
Transformacja wykrywa kształt zakodowany w elemencie strukturalnym. Konstrukcja tego elementu ma więc podstawowe znaczenie.

Rozpoznawanie prostych gestów wykonywanych na tablicy

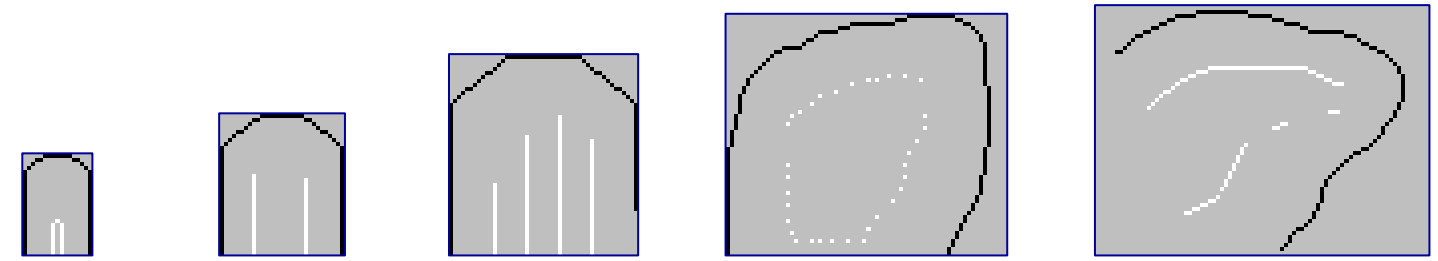
Polski alfabet palcowy



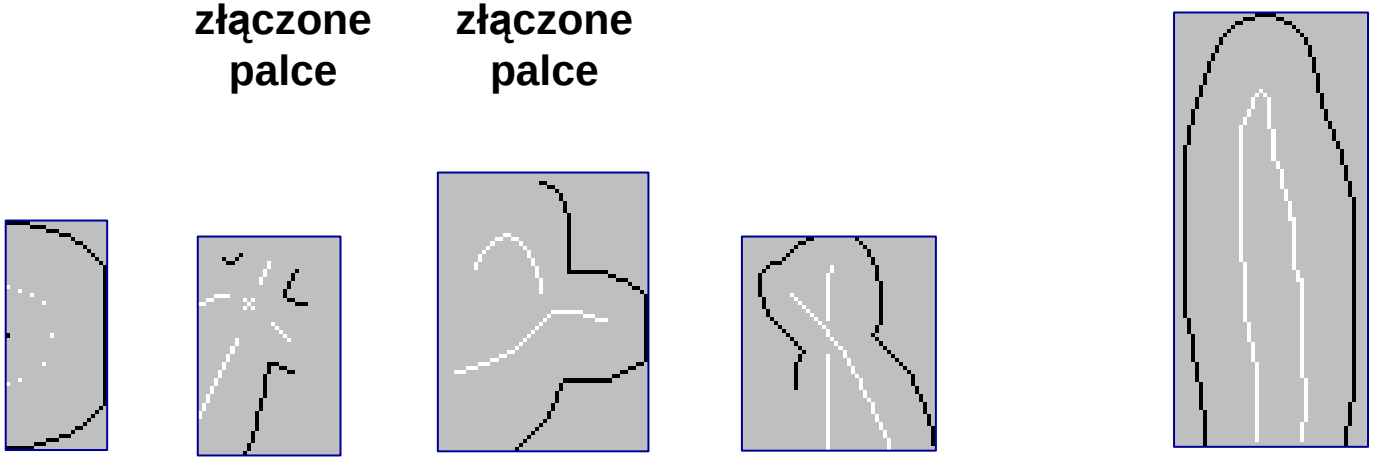
Analiza kształtu dłoni metodą trafi-nie trafi



Elementy strukturalne

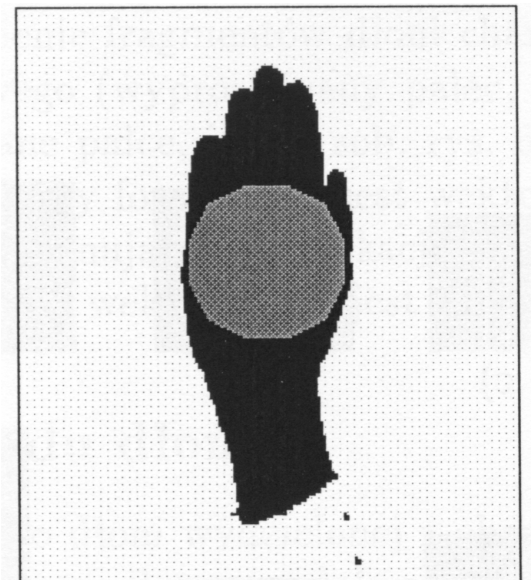


Palec | Dwa złączone palce | Cztery złączone palce | A | E / P



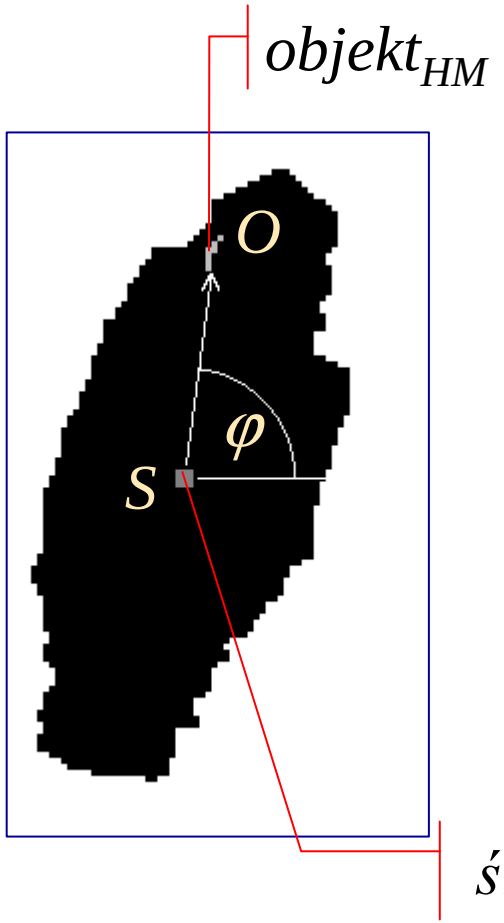
O | F / T | S | R | Sz

Skalowanie



Na początku sesji użytkownik przedstawia się pokazując literę B. Współczynnik skali jest wyznaczany na podstawie promienia największego wpisanego koła.

Konstrukcja wektora cech



Wektor cech = $\{c_1, c_2, ..., c_n\}$

c_i =(kod, φ, $|\overrightarrow{SO}|$, pole($objektu_{HM}$))

$i = 1, 2, ..., n$

Konstrukcja wektora cech

Wykorzystywane elementy

czwórka	kod	powierzchnia	kąt	odległość
1.	+	+	+	+
2.	+	+	+	
3.	+			
4.	+		+	
5.	+			

Wybór dokonany na podstawie analizy wpływu na skuteczność rozpoznawania

Wyniki eksperymentów

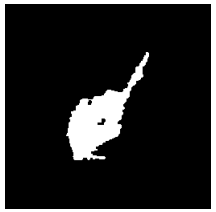
- zbiór danych 6361 obrazów
- zbiór uczący 5361
- zbiór weryfikujący 500
- zbiór testowy 500
- Statistica Neural Networks
  (sieć czterowarstwowa 18, 25, 25, 20 neuronów)
- skuteczność rozpoznawania
  uczenie 97.5 %
  weryfikacja 96.2 %
  testowanie 97.8 %

  nowe dane (ok. 10000 gestów) 88.1 %

Przykład:Rozpoznawanie prostych gestów do komunikacji z robotem

pięść

dłoń



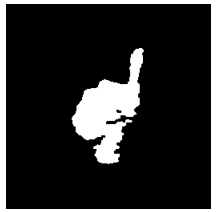
1 palec

2 palce

3 palce

4 palce

5 palców



kciuk

Baza danych

- **Dane podstawowe**: 4790 obrazów binarnych otrzymanych na podstawie detekcji skóry na obrazach kolorowych

gest	1p	2p	3p	4p	5p	pięść	dłoń	kciuk
liczebność	651	526	591	616	541	588	501	776

- **Dane obrócone**: 4790 obrazów podstawowych obróconych losowo o kąt z zakresu [–20 , +20] stopni
- **Dane zakłócone**: 4790 obrazów podstawowych zakłóconych szumem pieprz i sól

Wektor cech, dane uczące i testowe

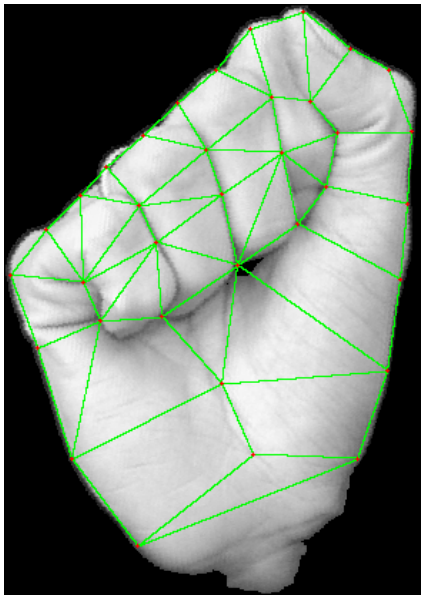
- **Wektor cech**: 11 elementów $c1, \ldots, c11$ otrzymanych metodami przetwarzania obrazów
- **Dane uczące**: dla 1696 obrazów (po 101 obrazów podstawowych, 100 obróconych i 11 zakłóconych)

Skuteczność rozpoznawania na zbiorze testowym [%]

kształt	1p	2p	3p	4p	5p	pięść	dłoń	kciuk	razem
liczeb-ność	1751	1354	1561	1635	1410	1551	1290	2112	12664
NN	97.2	92.2	92.1	95.8	96.9	99.9	98.4	98.1	96.4
DT	94.4	95.0	96.2	97.6	98.8	98.3	97.8	96.1	96.7
GP	97.5	95.3	88.0	96.1	98.3	97.2	98.3	93.2	95.3

NN – sieć neuronowa, **DT**- drzewo decyzyjne, **GP**-programowanie genetyczne

Metoda elastycznych grafów



[Triesch, von der Malsburg 2001]

- Rozpoznawany kształt jest reprezentowany przez graf o dwuwymiarowej topologii.
- Każdemu węzłowi grafu jest przypisany lokalny opis obiektu (*dżet*) w otoczeniu położenia tego węzła. Dżet jest w istocie wektorem cech.
- Oprócz dżetów opis grafu zawiera wagi krawędzi, które wyrażają geometryczne odległości między węzłami.

Metoda elastycznych grafów

Ogólna zasada postępowania

- Każdy wzorcowy kształt zostaje na etapie uczenia zamodelowany przez odpowiedni graf.
- Klasyfikacja opiera się na dopasowywaniu wzorcowych grafów do rozpoznawanego obiektu.
- Dopuszcza się nie tylko przesunięcie, obrót i zmianę skali grafu, ale także jego drobne deformacje. Zmiana skali i dopuszczalne deformacje implikują nazwę *elastyczny graf*.
- Dopasowanie modelu w formie grafu wzorcowego do badanego obrazu oznacza znalezienie położeń węzłów spełniających dwa ograniczenia:

(1) lokalna informacja o obrazie otoczenia każdego węzła w modelu musi pasować do aktualnej sytuacji,

(2) odległości między położeniami dopasowywanych węzłów nie powinny różnić się znacznie od odległości oryginalnych (dopuszczalna jest niewielka deformacja grafu).

Metoda elastycznych grafów

Sposób dopasowania

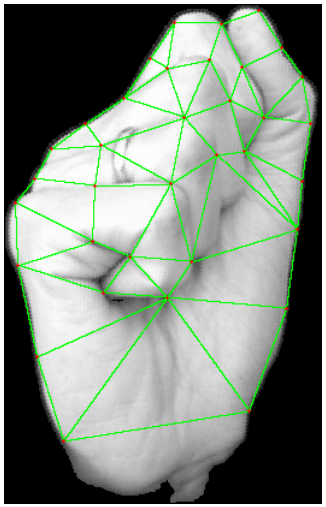
Dopasowanie grafu przebiega w dwóch etapach:

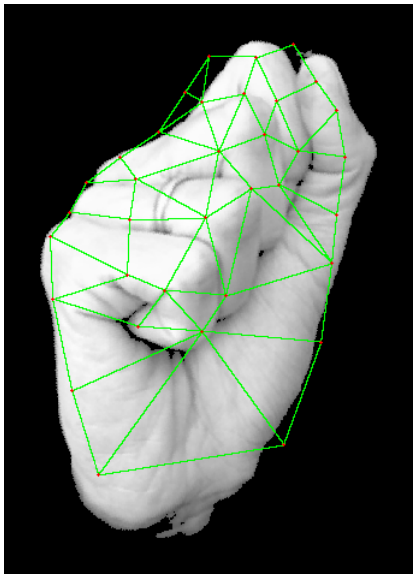
- Najpierw zmieniane jest położenie grafu w zakresie zadanym parametrem funkcji dopasowującej, by osiągnąć najlepszy stopień dopasowania. Zmiana polega na przesuwaniu i obracaniu grafu wokół środka ciężkości obiektu, a stopień dopasowania wynika z podobieństwa dżetów otrzymanych i wzorcowych.
- W drugim etapie zmienia się położenie poszczególnych węzłów grafu, przemieszczając je w zadanym otoczeniu. Dla każdego nowego położenia wyznaczana jest miara dopasowania grafu, w której teraz oprócz podobieństwa dżetów uwzględniana jest kara za deformację grafu, ujmująca zmianę odległości danego węzła od węzłów połączonych z nim krawędziami.

Przykładowe obrazy otrzymane na kolejnych etapach dopasowania grafu

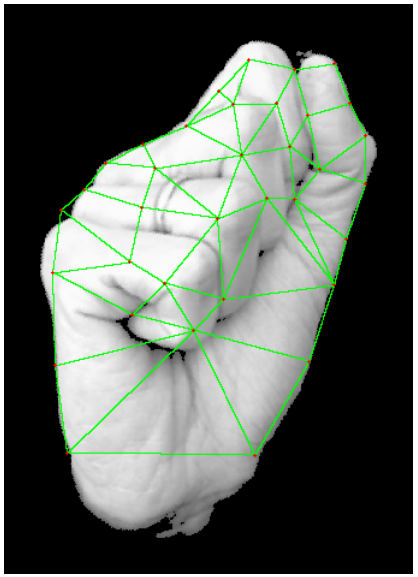
obrazy testowe z grafem wzorcowym

wzorzec

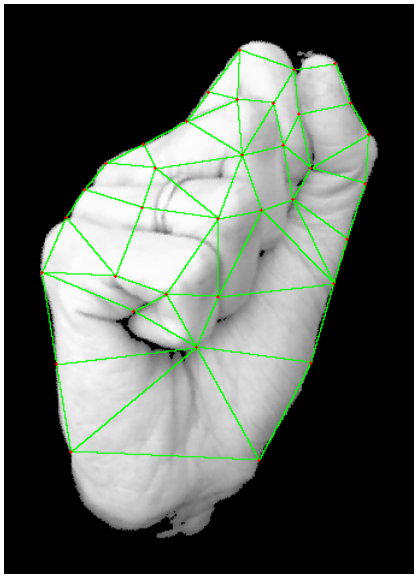




przed dopasowaniem



po dopasowaniu położenia



po dopasowaniu punktów

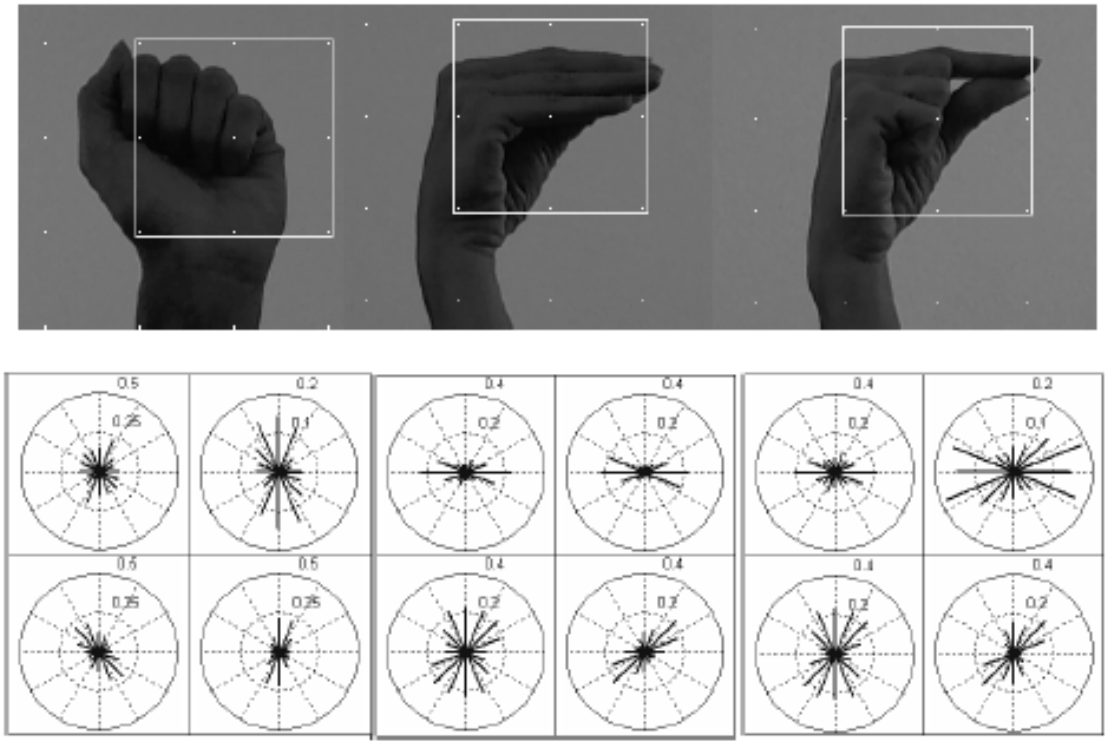
Wykorzystanie histogramów orientacji gradientów

[Dalal 2005]

- Podział obrazu (lub jego fragmentu zawierającego badany obiekt) na mniejsze obszary (komórki)
- Zbudowanie, dla każdej komórki, na podstawie wszystkich jej pikseli, jednowymiarowego histogramu kierunku gradientów
- Przypisanie orientacji do lokalnego pojemnika (binu) z wagą odpowiadającą modułowi gradientu
- Liniowa interpolacja między sąsiednimi binami w celu uniknięcia artefaktów kwantyzacji orientacji
- Normalizacja histogramu w celu uniezależnienia od zmiany skali

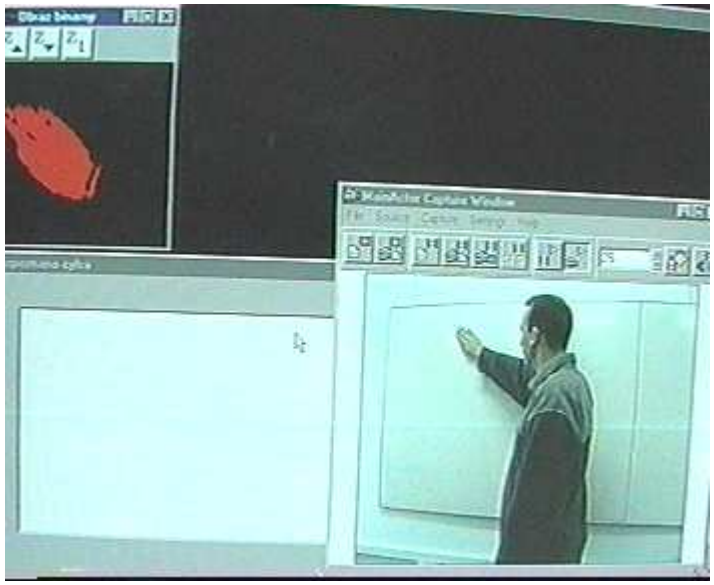
Rozpoznawany obiekt, np. dłoń, może być reprezentowany przez deskryptor uwzględniający histogramy orientacji gradientów we wszystkich komórkach.
Komórki przestrzenne są na ogół budowane na siatce prostokątnej określonej przez punkt centralny, zwykle ulokowany w środku ciężkości obiektu, i zadany rozmiar.

Wykorzystanie histogramów orientacji gradientów

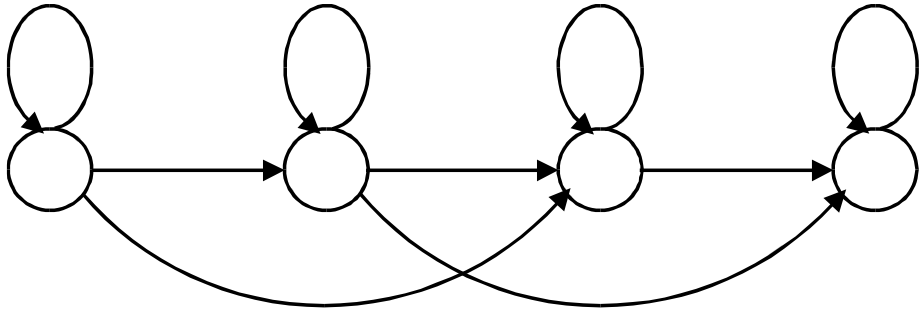


Przykładowe obrazy dla znaków A, E, P

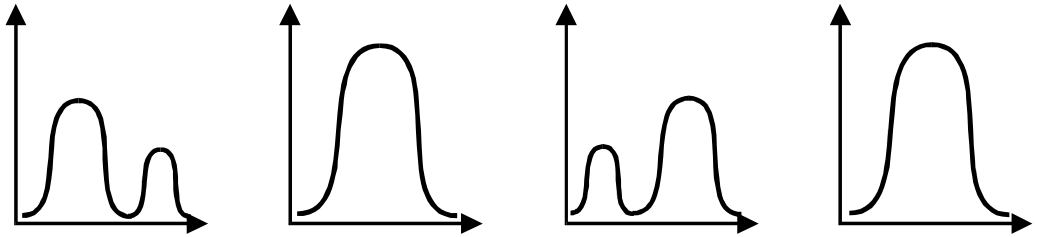
Rozpoznawanie gestów dynamicznych



Ukryte modele Markowa



wektory obserwacji

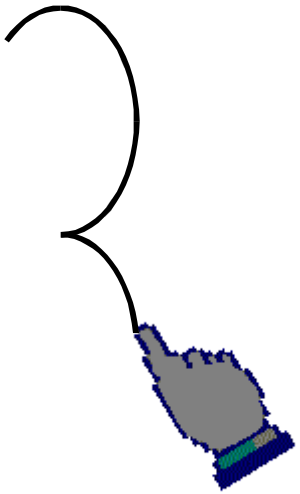


rozkłady prawdopodobieństwa

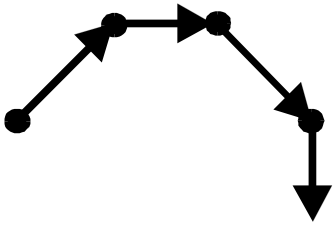
czas	t:	1,	2,	...	T
stan	s(t):	s(1),	s(2),	...	s(T)
obserwacja	o(t):	o(1),	o(2),	...	o(T)

Przykład

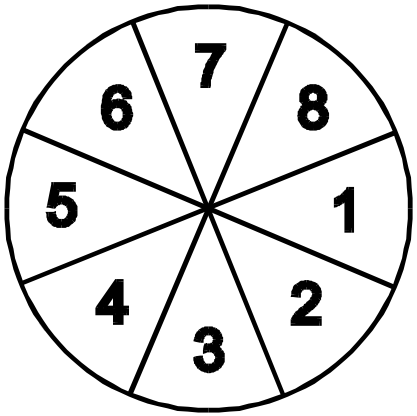
pisana cyfra



obserwacja



model dyskretny



model ciągły

(i) orientacja $\varphi(t)$

(ii) położenie $l_x(t)$, $l_y(t)$

Opis matematyczny

- macierz przejścia:

$$A = \begin{bmatrix} a(1|1) & a(1|2) & \dots & a(1|S-1) & a(1|S) \\ \vdots & \vdots & a(i|j) & \vdots & \vdots \\ a(S|1)` & a(S|1) & \dots & a(S|S-1) & a(S|S) \end{bmatrix} \qquad a(i|j) \overset{def}{=} P(s(t)=i \mid s(t-1)=j)$$

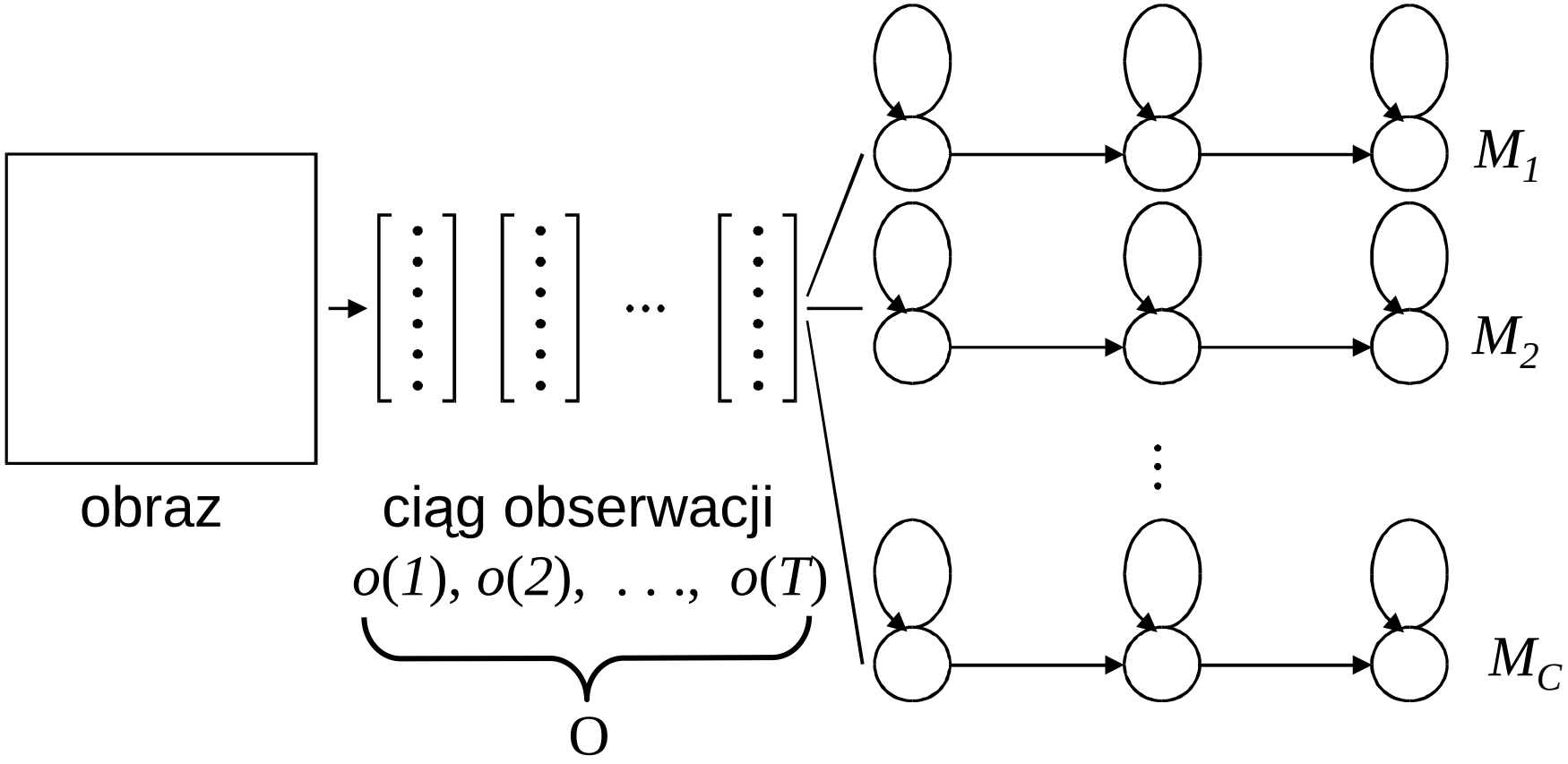
- macierz prawdopodobieństwa obserwacji:

$$B = \begin{bmatrix} b(1|1) & b(1|2) & \cdots & b(1|S-1) & b(1|S) \\ \vdots & & b(k/i) & & \vdots \\ b(K|1) & b(K|2) & \cdots & b(K|S-1) & b(k|S) \end{bmatrix} \qquad b(k|i) \overset{def}{=} P(o(t)=k \mid s(t)=i)$$

- początkowy rozkład prawdopodobieństwa:

$$\Pi(1) \overset{def}{=} \begin{bmatrix} P(s(1)=1) \\ P(s(1)=2) \\ \vdots \\ P(s(1)=S) \end{bmatrix}$$

Klasyfikacja



c^* - rozpoznana klasa, $c^* = \max_c p(O|M_c)$

Problemy

- wyznaczenie prawdopodobieństwa $p(O|M_c)$ – algorytm w przód

- wyznaczenie najbardziej prawdopodobnej sekwencji stanów σ^*

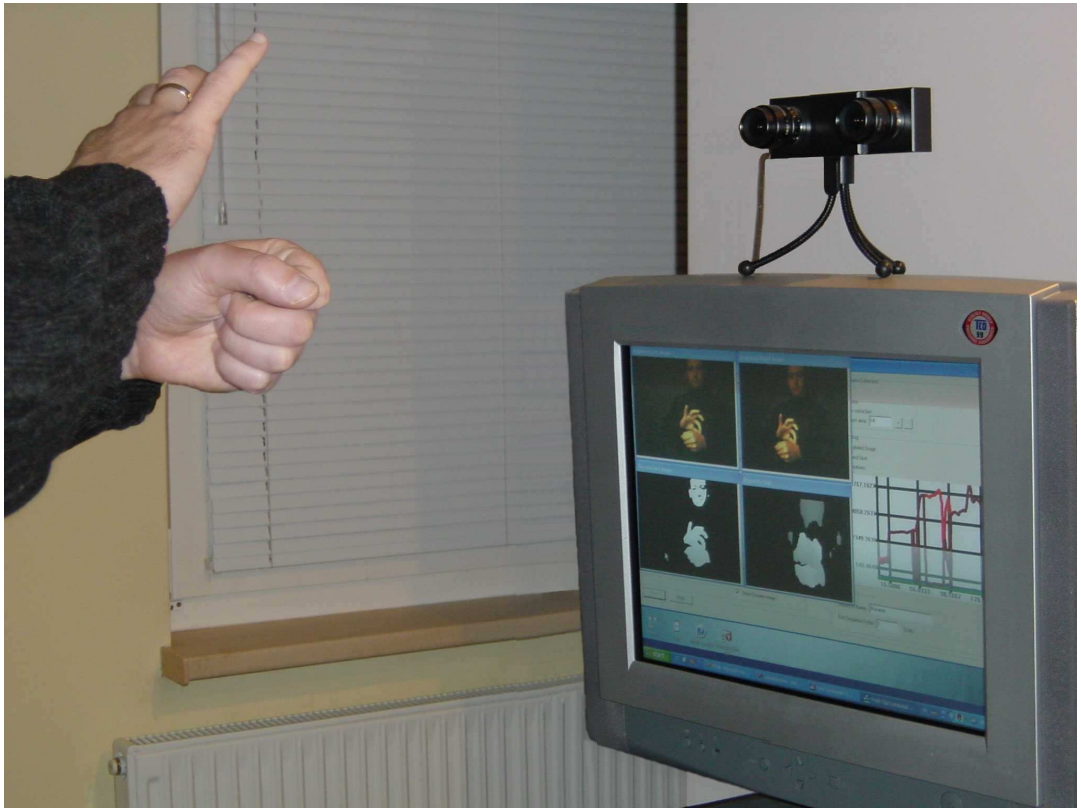
  $\sigma^* = \arg\max_{\sigma} p(O, \sigma|M_c)$ - algorytm Viterbiego

- wyznaczenie elementów macierzy *A*, *B* (uczenie):
  metoda Bauma-Welcha
  metoda Viterbiego

Charakterystyka polskiego języka migowego

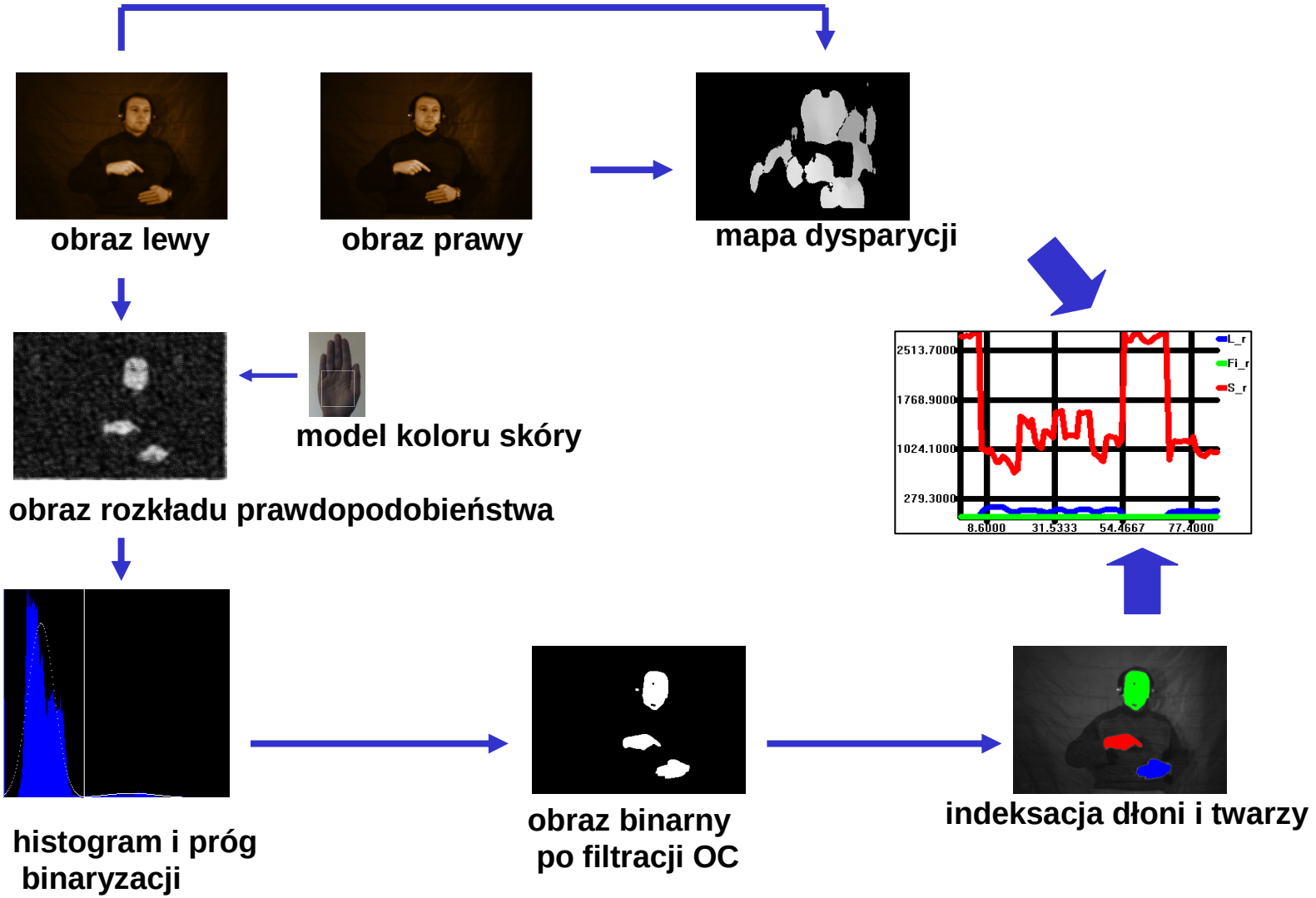
1. Większość gestów to gesty dynamiczne.
2. Większość gestów to gesty oburęczne.
3. Ręce mogą się przysłaniać lub ukazywać na tle twarzy.
4. Wykonanie zależy od osoby.

Rozpoznawanie wypowiedzi języka migowego

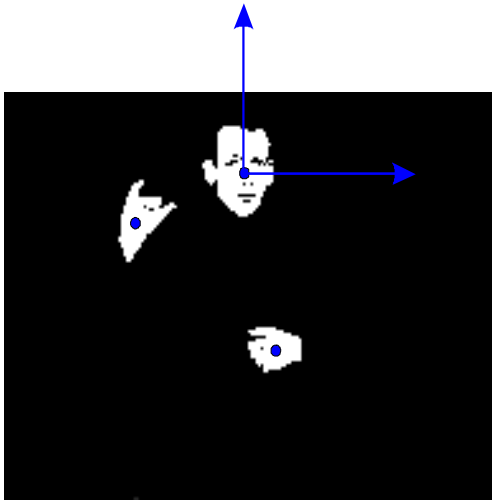


stereo, kolor,
obrazy 320x240,
25obrazów/s

Przetwarzanie obrazów



Elementy wektora cech



x_r, y_r - współrzędne środka ciężkości prawej dłoni względem początku układu współrzędnych w środku ciężkości twarzy

A_r - powierzchnia prawej dłoni

γ_r - zwartość prawej dłoni

ε_r - niecentryczność prawej dłoni

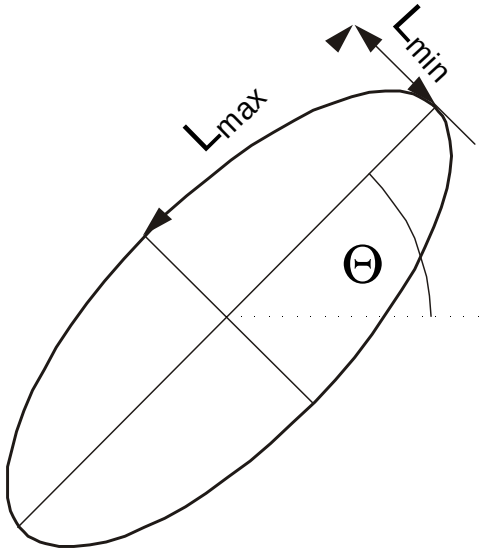
θ_r - orientacja prawej dłoni (kąt między główną osią dłoni i osią x)

z_r - różnica między średnią głębią twarzy i prawej dłoni

odpowiednie parametry dla dłoni lewej

Orientacja

$$\Theta = 0.5\ tan^{-1}[2m_{11}/(m_{20} - m_{02})]$$



Niecentryczność

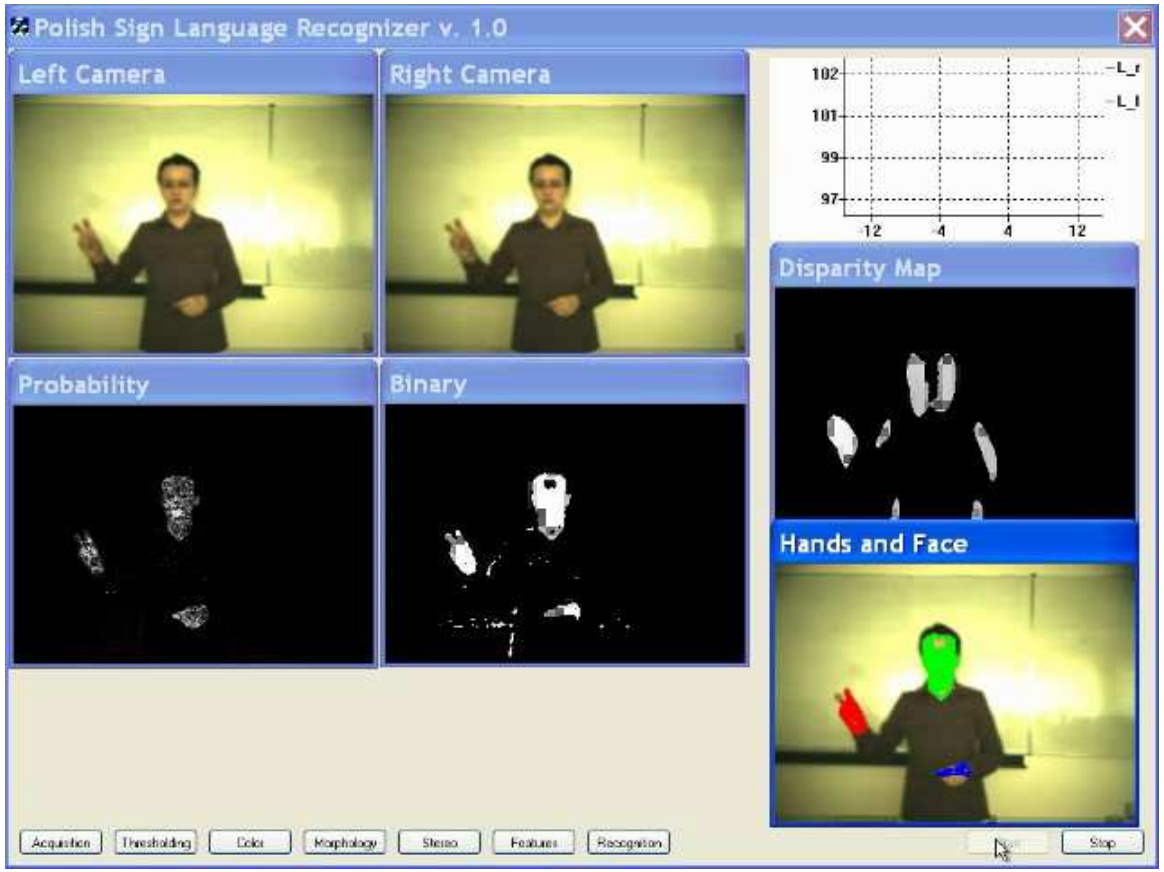
$$\varepsilon = L_{max}/L_{min} = [(m_{20} - m_{02})^2 + 4(m_{11})^2]/(A)^4$$

Zwartość (współczynnik cyrkularności)

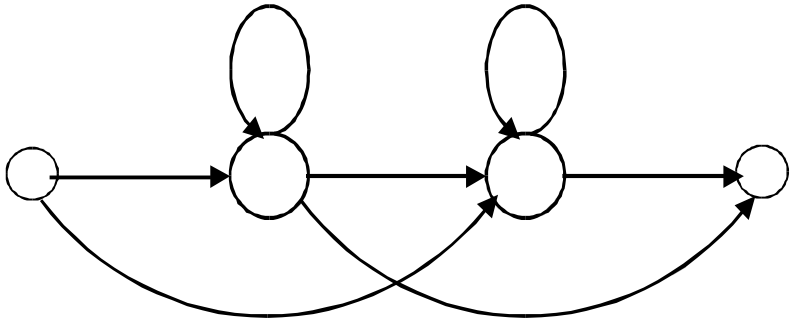
$$\gamma = P^2/(4\pi A)$$

A - pole, P – obwód
m_{ij} – momenty centralne

Rozpoznawanie wypowiedzi języka migowego

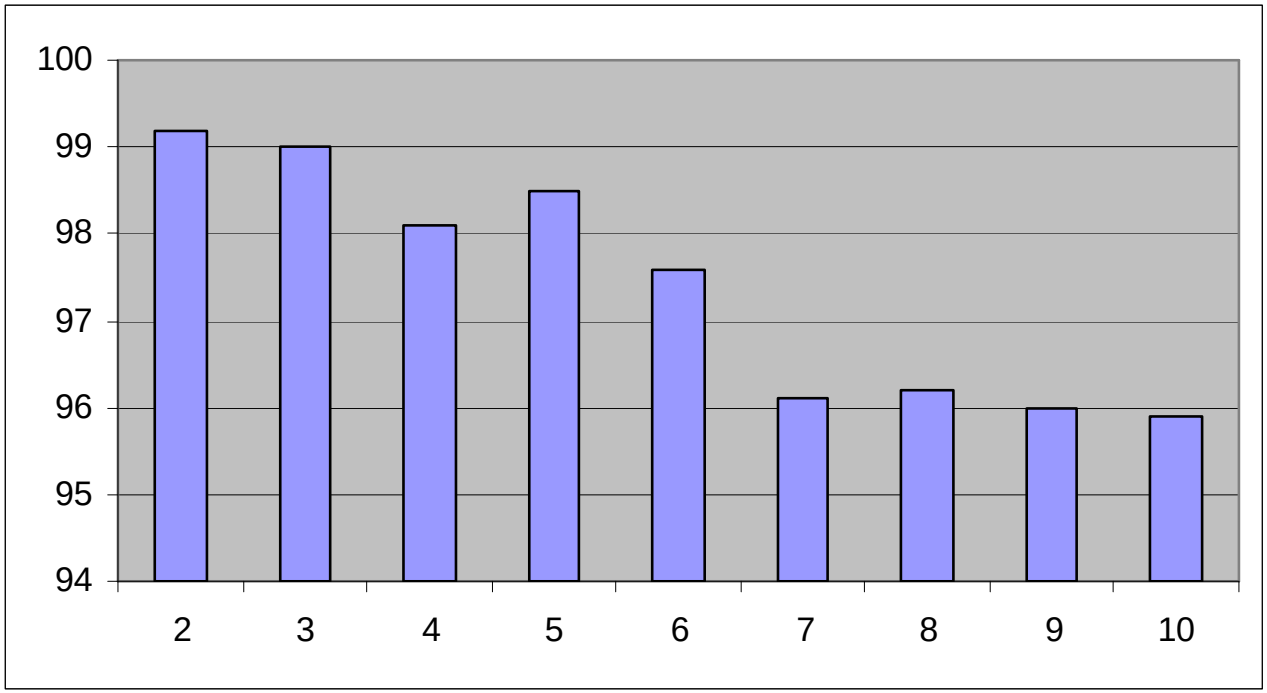


Rozpoznawanie izolowanych słów



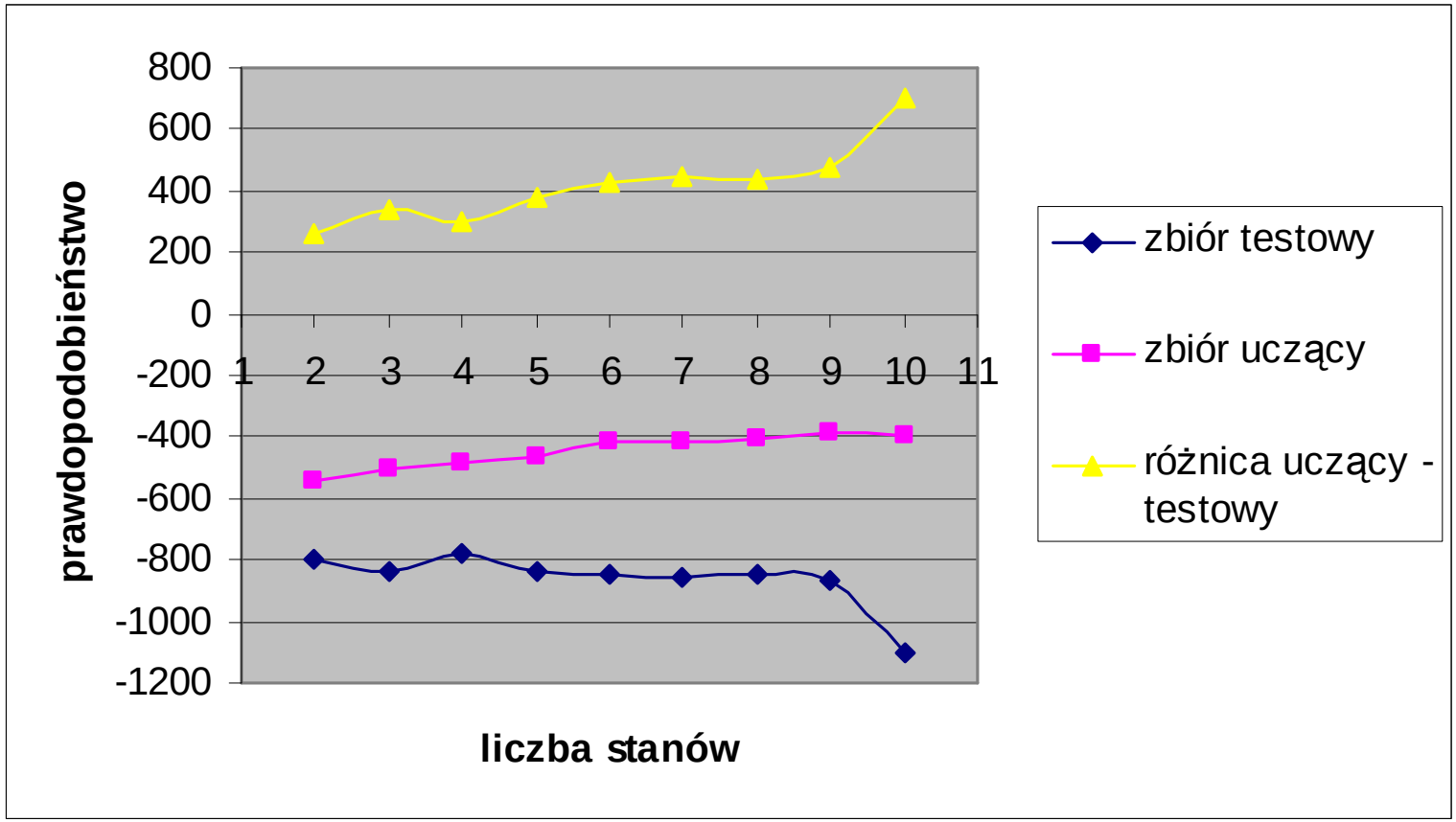
HMM z dwoma stanami emitującymi
(pakiet HTK)

Wybór liczby stanów



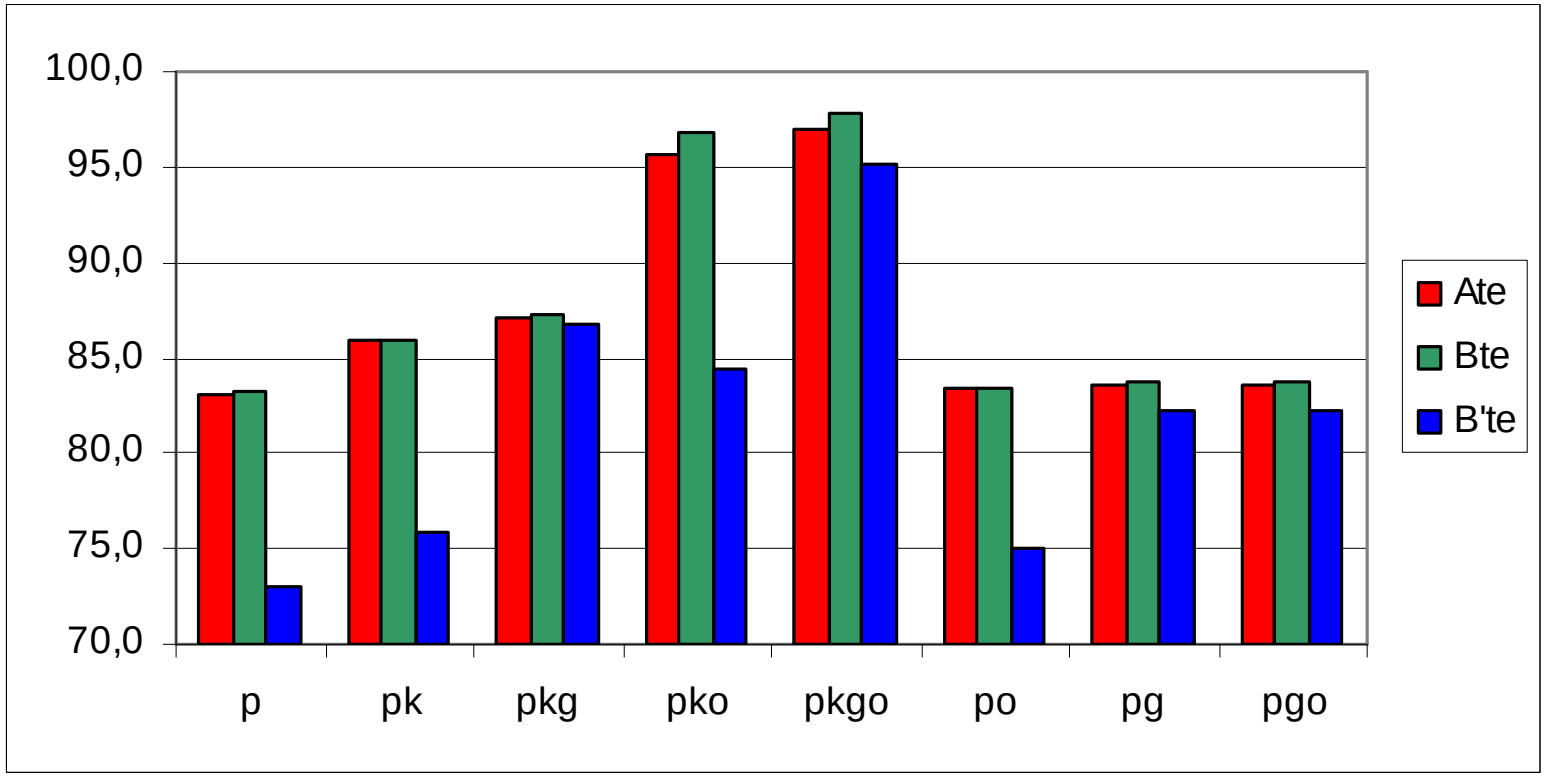
Skuteczność rozpoznawania na zbiorze testowym
w zależności od liczby stanów

Wybór liczby stanów



Średnia wartość prawdopodobieństwa $P(O|M_i)$ wygenerowania słowa przez odpowiadający mu model (skala logarytmiczna)

Wybór cech

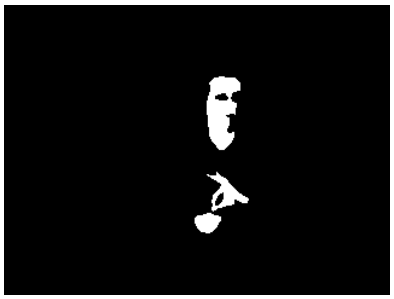


p – położenie (x,y), k – ksztalt (S, ε, γ), g – głębia (z), o – orientacja (θ)
te- zbiór testowy, A -osoba A, B - osoba B,
B’ – osoba B i trudniejsze warunki oświetlenia

Znaczenie informacji o głębi w trudniejszych warunkach oświetlenia

początkowa faza gestu „skierowanie”

obraz wejściowy



obraz binarny

mapa dysparycji

końcowa faza gestu

obraz wejściowy

obraz binarny

mapa dysparycji

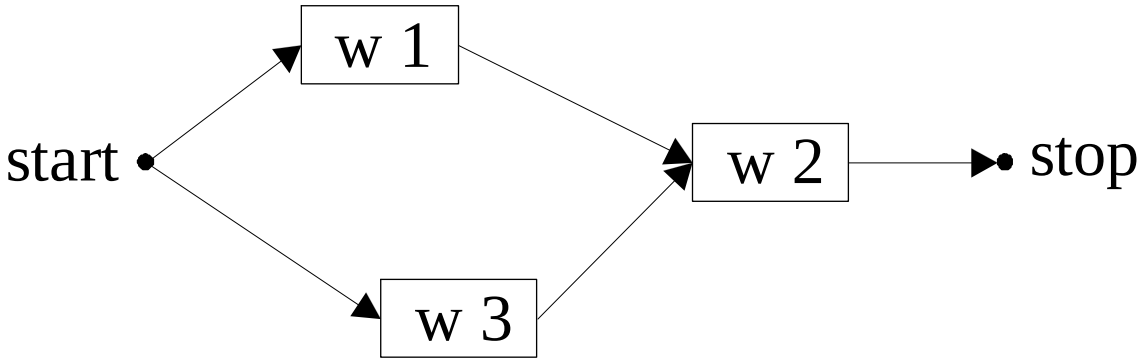
Eksperyment – izolowane słowa

101 słów powtórzonych 20 razy przez dwie osoby
10 razy do uczenia, 10 razy do testowania

Osoba	Skuteczność rozpoznawania [%]
uczenie A/ testowanie A	97.4
uczenie B / testowanie B	97.4
uczenie A/ testowanie B	93.6
uczenie B/ testowanie A	93.3
uczenie A,B/ testowanie A	95.3
uczenie A,B/ testowanie B	96.2

Rozpoznawanie w czasie rzeczywistym (obrazy 320x240, 25f/s)

Rozpoznawanie zdań



1) w1 w2
2) w3 w2

Zdania rozpoznawano wykorzystując złożony model w formie sieci modeli izolowanych słów, dostrojony na podstawie przykładowych wypowiedzi (embedded training). Procedura dostrajania wymaga tylko podania transkrypcji każdego zdania w ciągu uczącym. Wskazanie granic poszczególnych wyrazów nie jest potrzebne.

Model bigram

Uwzględnia prawdopodobieństwa następstwa wyrazów określane na podstawie liczności występujących po sobie wyrazów w sekwencjach wykorzystanych do uczenia

$$p(i \mid j) = \frac{N(i, j)}{N(j)}$$

$N(i, j)$ – liczba wystąpień słowa i po słowie j
$N(j)$ – liczba wystąpień słowa j w bazie danych uczących

Eksperyment - zdania

35 zdań powtórzonych 20 razy przez dwie osoby
10 razy do uczenia, 10 razy do testowania

Osoba	Skuteczność rozpoznawania [%]
uczenie A/ testowanie A	97.1
uczenie B/ testowanie B	97.4
uczenie A/ testowanie B	93.4
uczenie B/ testowanie A	92.9
uczenie A,B/ testowanie A	96.3
uczenie A,B/ testowanie B	97.4

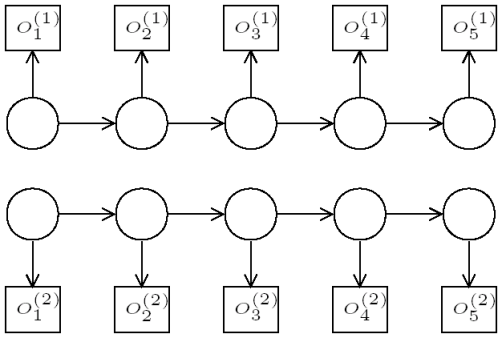
Rozpoznawanie w czasie rzeczywistym (obrazy 320x240, 25f/s)

Procesy równoległe

Modelując wypowiedzi języka migowego trzeba wziąć pod uwagę, że obserwowane cechy manualne występują zarówno szeregowo jak i równolegle. Przykładowo, kształt i położenie dłoni mogą się zmieniać równocześnie lub obie ręce wykonują gest równolegle.

Równolegle procesy można zamodelować wyróżniając grupy cech (kanały), z założeniem, że te procesy występują niezależnie od siebie.

Równoległe ukryte modele Markowa PaHMM



PaHMM

Rozszerzenie zwykłych ukrytych modeli Markowa

- Równoczesne procesy są reprezentowane przez odrębne kanały.
- Kanały są modelowane jako zwykle ukryte modele Markowa uczone niezależnie.

Możliwe podejścia

(i) Odrębny kanał dla każdej cechy

(ii) Odrębny kanał dla grupy cech, np. dla każdej dłoni

Równoległe ukryte modele Markowa PaHMM

Odrębny kanał dla grupy cech dla każdej dłoni
20 wykonań każdego z 35 zdań – 700 wypowiedzi
Modele zdań zbudowano z modeli wyrazów.
Nie dostrajono ich na przykładach zdań

Skuteczność rozpoznawania HMM 28.4%
Skuteczność rozpoznawania PaHMM 64.4%

Wykorzystanie nowych metod obrazowania 3D

Kamery ToF

Sensor Kinect

Kamera ToF

Kamery ToF uzyskują informację przestrzenną na zasadzie pomiaru czasu przepływu światła.

Scena jest oświetlana impulsem świetlnym (zwykle w zakresie podczerwieni), a a kamera w każdym punkcie obrazu mierzy czas potrzebny na przepływ światła do obserwowanego obiektu i z powrotem na podstawie przesunięcia fazowego między zmodulowanym sygnałem wysłanym i odbitym. Ten czas jest proporcjonalny do odległości, więc w każdym punkcie obrazu otrzymywana jest informacja odległości od odwzorowanego w nim punktu obiektu.

Pomiar odległości jest bardzo wydajny i niezawodny w porównaniu do stereowizji, gdzie trzeba stosować złożone algorytmy korelacji.
Metoda jest też mniej zależna od tekstury obiektu.

Kamera ToF

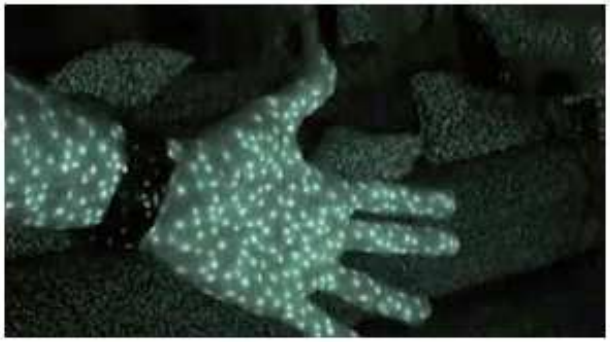
Kamery ToF podają obraz odległości i obraz amplitudy z intensywnością modulacji podczerwieni i częstotliwością wideo.

Obraz odległości (lub głębokości) zawiera dla każdego piksela mierzoną promieniowo odległość między badanym pikselem i jego rzutem na obserwowany obiekt, natomiast obraz amplitudowy odzwierciedla dla każdego piksela moc sygnału odbitego przez obiekt.

Kamery TOF są bardzo obiecującym narzędziem w systemach interakcji człowiek-komputer. Dzięki nim redukuje się znaczenie ograniczeń odnośnie do oświetlenia, tła i ubioru, obowiązujących w wizyjnych systemach rozpoznawania gestów.

W badaniach została użyta kamera ToF MESA Swiss Ranger 4000.
Ma ona zakres od 30 cm do 5 m, rozdzielczość obrazu 176 × 144, rozdzielczość odległości 1 cm i szybkość do 50 obrazów/s .

Sensor Kinect



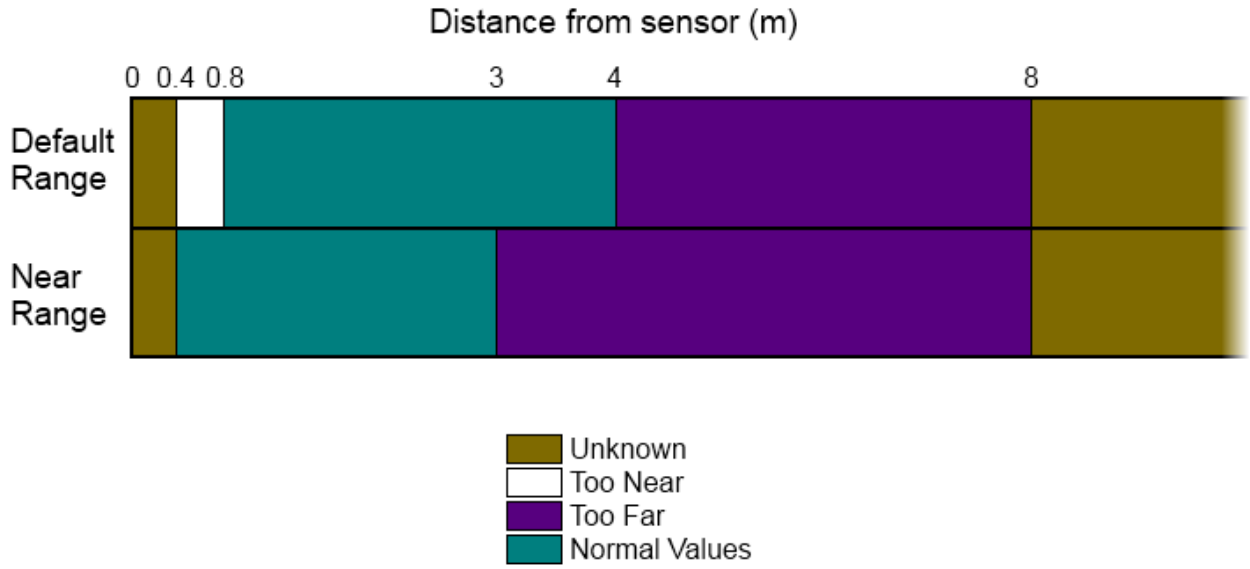
www.youtube.com

Urządzenie ma dwie kamery. Jedna daje obraz RGB o rozdzielczości 640x480. Druga jest częścią podsystemu sensora zwracającego informację o głębokości.

Promiennik podczerwieni wyświetla przed kamerą punkty, których położenie jest rejestrowane przez kamerę z filtrem podczerwieni.
Informacja o głębi o rozdzielczości 320x240 jest programowo interpolowana do rozdzielczości kamery wizyjnej. Zakres działania czujnika odległości to 0.8-4.0 m. Oprogramowanie generuje szkielet (20) przegubów
Częstość – 30 klatek/s

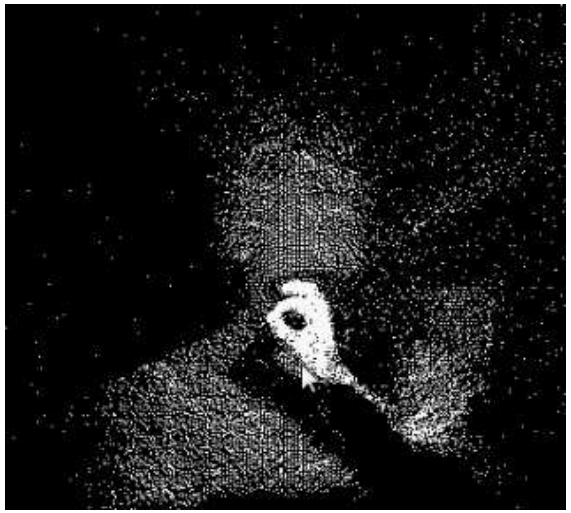
Sensor Kinect 2

Urządzenie ma dwie kamery 512x 424. Jedna daje obraz RGB.
Druga jest kamerą ToF.
Działanie czujnika odległości 0.8-4.0 m.
Oprogramowanie generuje szkielet (25) przegubów
Częstość – 30 klatek/s



Struktury danych 3D – chmury punktów i mapy głębi

- Chmura punktów - reprezentuje zbiór punktów w przestrzeni trójwymiarowej. Jej atrybutami są współrzędne geometryczne *x*, *y*, *z* punktów.



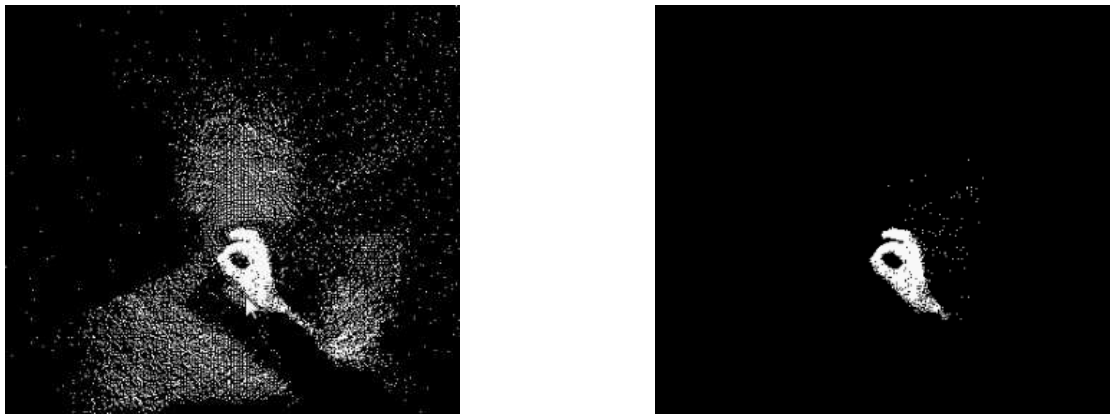
- Mapa głębi to obraz monochromatyczny, w którym jasność pikseli oznacza głębię obiektów.

Wybrane elementy przetwarzania chmury punktów

Funkcja	Komentarz
Filtracja chmury	Odrzuca punkty o wartościach wskazanej współrzędnej poza zadanym przedziałem
Usuwanie punktów izolowanych	Usuwa rzadko rozłożone punkty, znacznie odległe od swoich sąsiadów
Redukcja liczby punktów chmury	Zastępuje punkty w obrębie każdego woksela zdefiniowanej siatki prostokątnej jednym punktem
Wyznaczanie wektorów normalnych do powierz-chni rozpiętej na chmurze	Wykorzystuje analizę wartości i wektorów własnych macierzy kowariancji obliczanej w zadanym sąsiedztwie punktu
Budowa deskryptorów lokalnych (PFH)	Buduje dokładniejsze niż wektor normalny opisy lokalnych właściwości powierzchni (w formie histogramów)
Budowa deskryptora globalnego (VFH)	Buduje opis całej chmury (w formie histogramu)

Filtracja chmury punktów

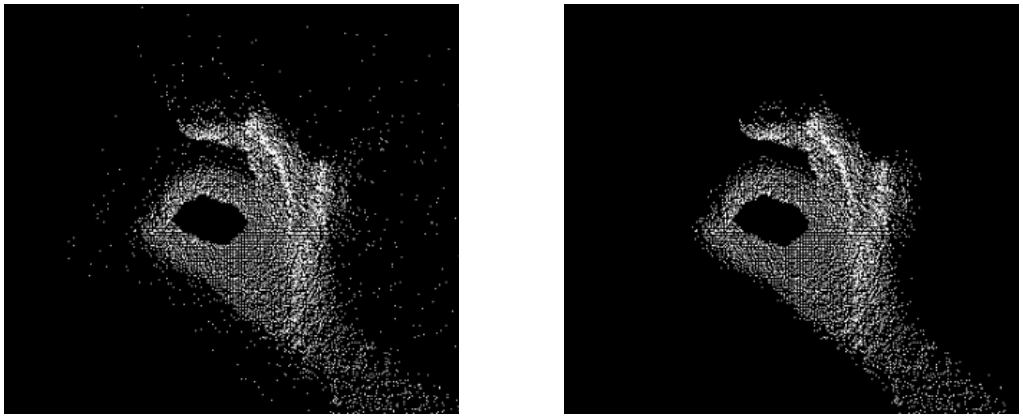
Filtracja jest przeprowadzana dla wybranego wymiaru i polega na odrzuceniu tych punktów, dla których wartości odpowiedniej współrzędnej nie leżą w zadanym przedziale.



Chmura punktów przed i po filtracji

Usuwanie punktów izolowanych

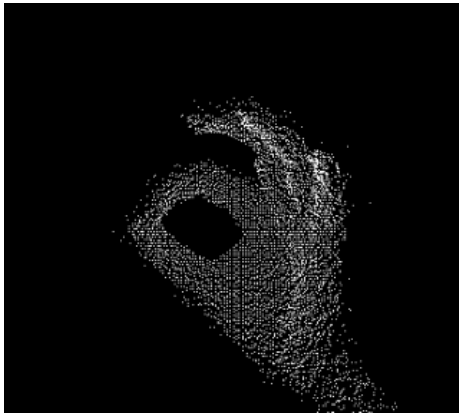
Dla każdego punktu chmury wyznaczana jest jego średnia odległość od wszystkich punktów należących do jego sąsiedztwa. Rozmiar sąsiedztwa jest parametrem filtru. Otrzymuje się w ten sposób rozkład średnich odległości o liczności równej rozmiarowi chmury punktów.
Przyjmując, że rozkład ten jest normalny, oblicza się jego wartość średnią i odchylenie standardowe i odrzuca się wszystkie punkty, dla których średnie odległości nie mieszczą się w przedziale $\langle \mu - n\sigma, \mu + n\sigma \rangle$, gdzie *n* jest parametrem filtru.



Chmura punktów przed i po usunięciu punktów izolowanych

Redukcja liczby punktów chmury

Scena 3D może być przedstawiona jako siatka lub trójwymiarowa macierz tzw. **wokseli**. Woksel (volumetric element) jest trójwymiarowym odpowiednikiem piksela. Parametrami filtru są fizyczne rozmiary woksela rozumianego jako prostopadłościan. Po ich podaniu tworzona jest siatka pokrywająca obserwowaną scenę. Następnie dla każdego woksela, wszystkie leżące w jego obrębie punkty zastępowane są jednym punktem, zlokalizowanym w ich środku ciężkości.



Chmura składająca się z 25344 punktów po ustawieniu rozmiarów woksela na 0.005 m × 0.005 m × 0.005 m została zredukowana do 2423 punktów.

Wyznaczanie wektorów normalnych do powierzchni rozpiętej na chmurze

Wektor normalny do powierzchni w danym punkcie to wektor prostopadły do płaszczyzny stycznej do tej powierzchni w tym punkcie. Jedną z metod wyznaczania wektorów normalnych polega na analizie wartości i wektorów własnych macierzy kowariancji obliczanej w zadanym sąsiedztwie punktu, dla którego wyznaczany jest wektor normalny.
Macierz ta obliczana jest ze wzoru:

gdzie:

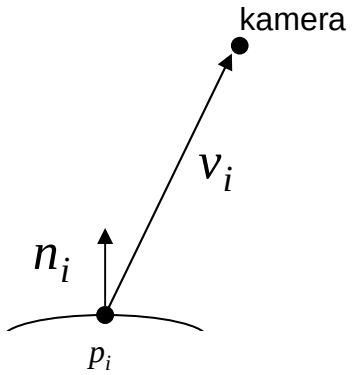
$$C_i = \frac{1}{k}\sum_{j=1}^{k}\left(p_j - \overline{p}\right)\left(p_j - \overline{p}\right)^T$$

i – indeks punktu, w którym jest wyznaczany wektor normalny,
C_i - macierz kowariancji,
k - liczba punktów wchodzących w skład sąsiedztwa,
p_j - j-ty punkt sąsiedztwa,
$\overline{p}$ - środek ciężkości punktów wchodzących w skład sąsiedztwa.

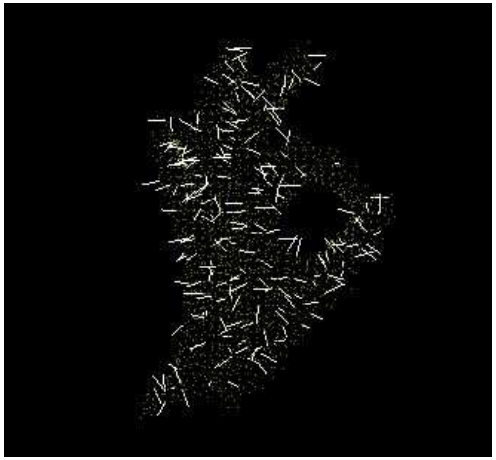
Wyznaczanie wektorów normalnych do powierzchni rozpiętej na chmurze

Przyjmuje się, że wektor normalny ma kierunek pokrywający się z kierunkiem najkrótszego wektora własnego macierzy kowariancji C_i i jest zaczepiony w punkcie p_i.
Zwrot wektora normalnego wybiera się tak, aby kąt pomiędzy wektorami n_i i v_i był ostry:

n_i - wektor normalny wyznaczony w punkcie p_i,
v_i - wektor rozpięty pomiędzy punktem p_i a punktem, w którym zlokalizowany jest obserwator (kamera).

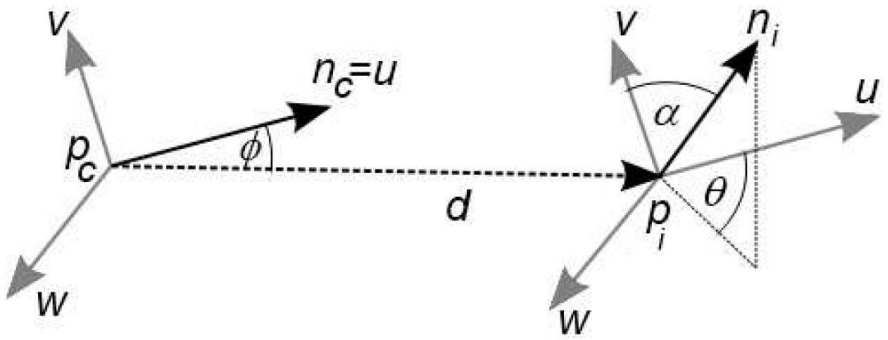


Wyznaczanie wektorów normalnych do powierzchni rozpiętej na chmurze



Wektory normalne do powierzchni dłoni
(dla przejrzystości rysunku pokazano tylko niektóre)

Deskryptor VFH (Viewpoint Feature Histogram)



$$u = n_c\,,\ v = \frac{d}{|d|} \times u\,,\ w = u \times v$$

$$\cos\alpha = v \cdot n_i\,,\ \cos\phi = u \cdot \frac{d}{|d|}$$

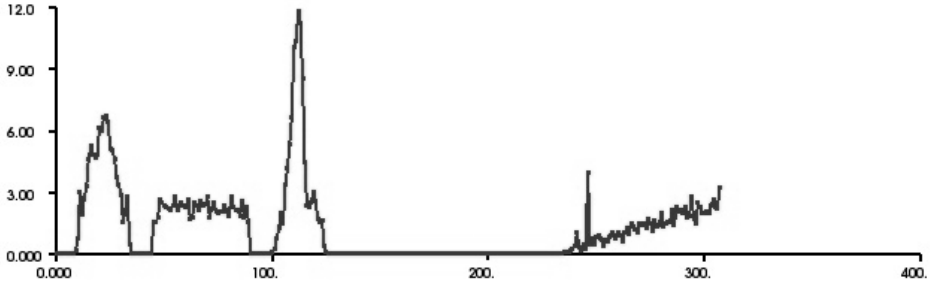
$$\theta = actg\,\frac{w \cdot n_i}{u \cdot n_i}$$

Dwa składniki:

(i) kształt powierzchni: θ, cos(α), cos(φ), $|d|$

(ii) kierunek widzenia: obliczany dla każdej pary punktów (p_c, p_i), gdzie p_c jest środkiem ciężkości chmury punktów, p_i *jest* punktem chmury, n_c jest wektorem zaczepionym w p_c stanowiącym uśrednienie wektorów normalnych do powierzchni we wszystkich punktach p_i, n_i jest wektorem normalnym do powierzchni w punkcie p_i

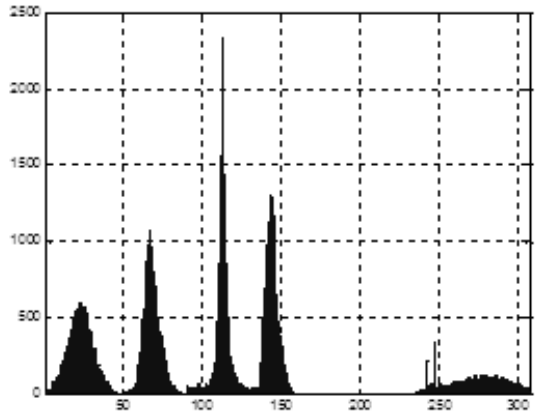
Deskryptor VFH (Viewpoint Feature Histogram)

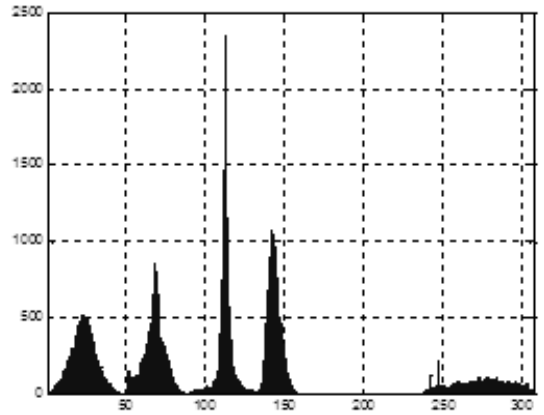


(a) (b)

(a) Chmura punktów przedstawiająca dłoń, (b) Deskryptor.
Na osi x są zaznaczone etykiety odpowiadające kolejnym wartościom kątów: cosα (etykiety od 1 do 45), cosφ (46-90), θ (91-135), d (136-180) - brak,
ψ (181-308) – ψ oznacza kąt między kierunkiem patrzenia a kierunkiem normalnym do powierzchni w każdym punkcie chmury.
Na osi y jest liczba punktów chmury, dla której zaobserwowano daną wartość kąta.

Rozpoznawanie gestów statycznych

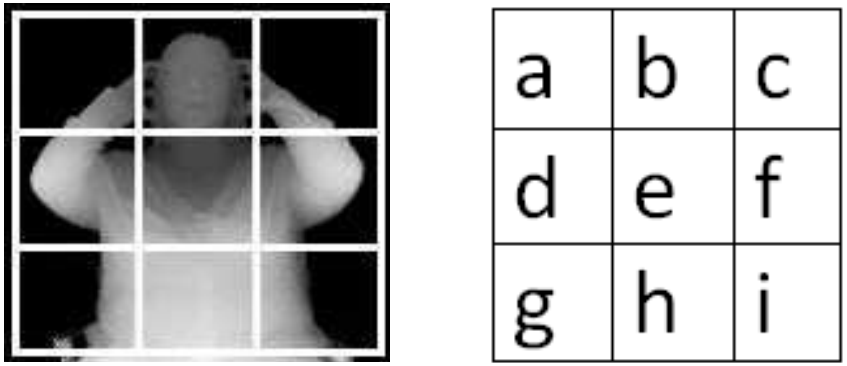




gest

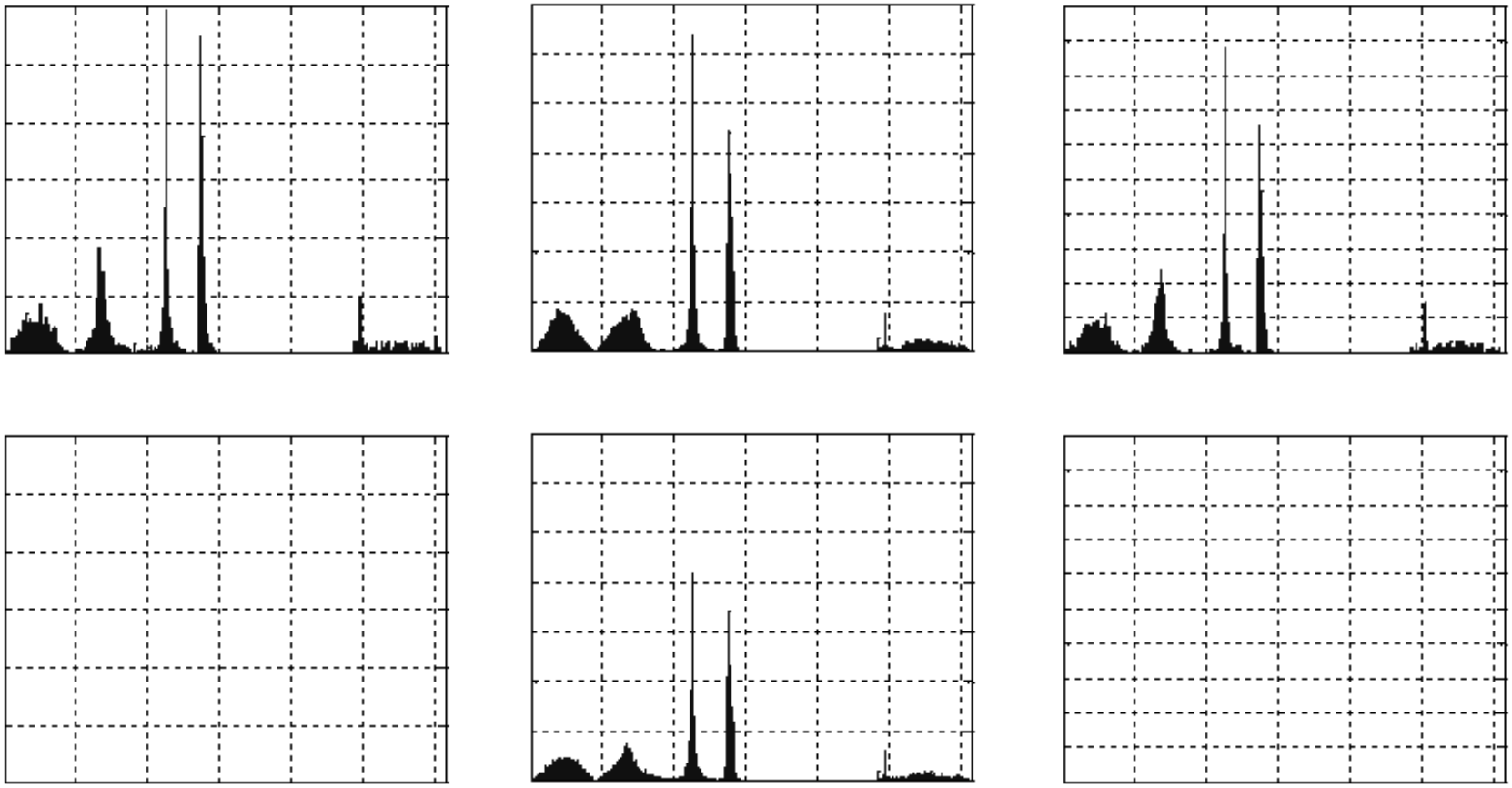
deskryptor VFH (α,φ,θ,d,Ψ)

Rozpoznawanie gestów



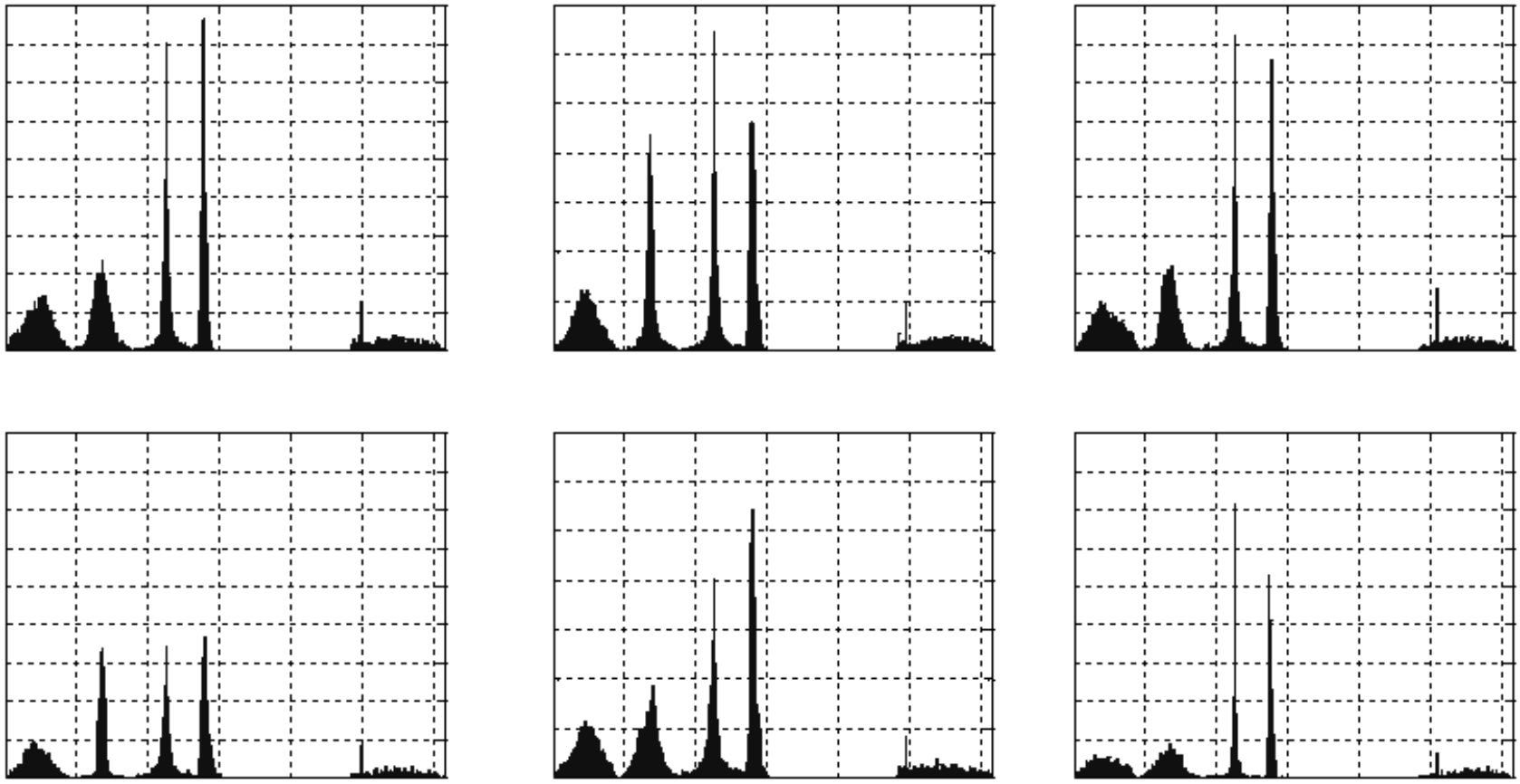
Podział mapy głębi na komórki

Rozpoznawanie gestów



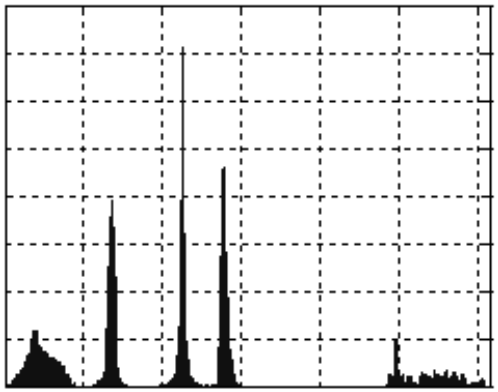
(a) (b) (c)

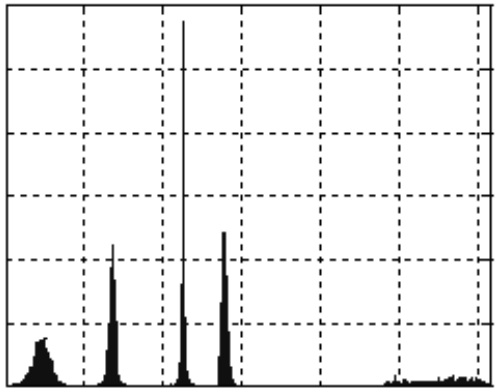
Rozpoznawanie gestów

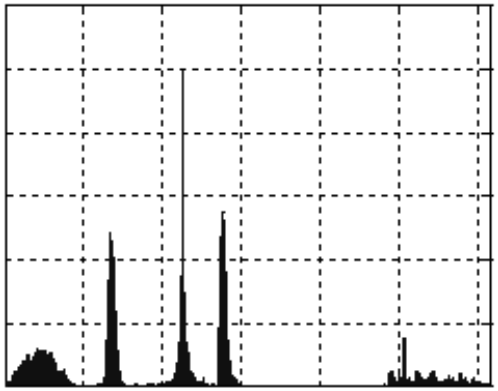


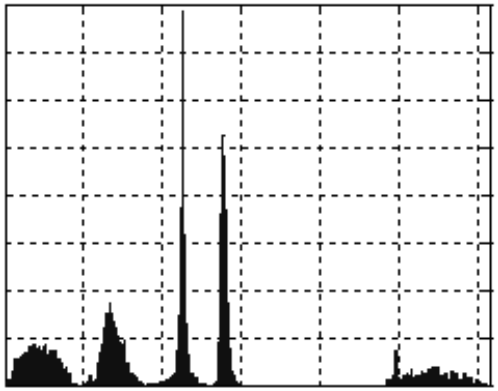
(d) (e) (f)

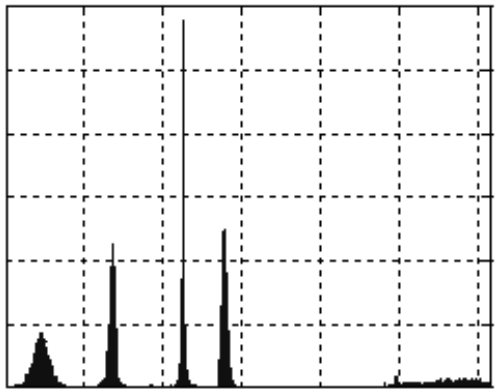
Rozpoznawanie gestów

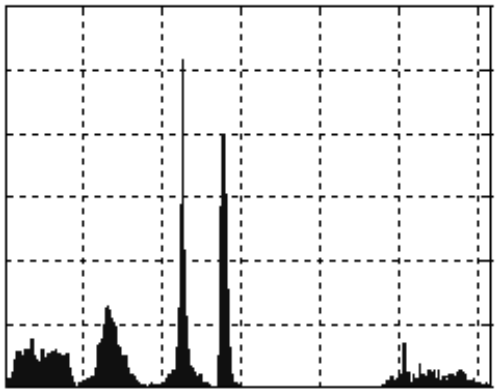












(g) (h) (i)

Rozpoznawanie gestów

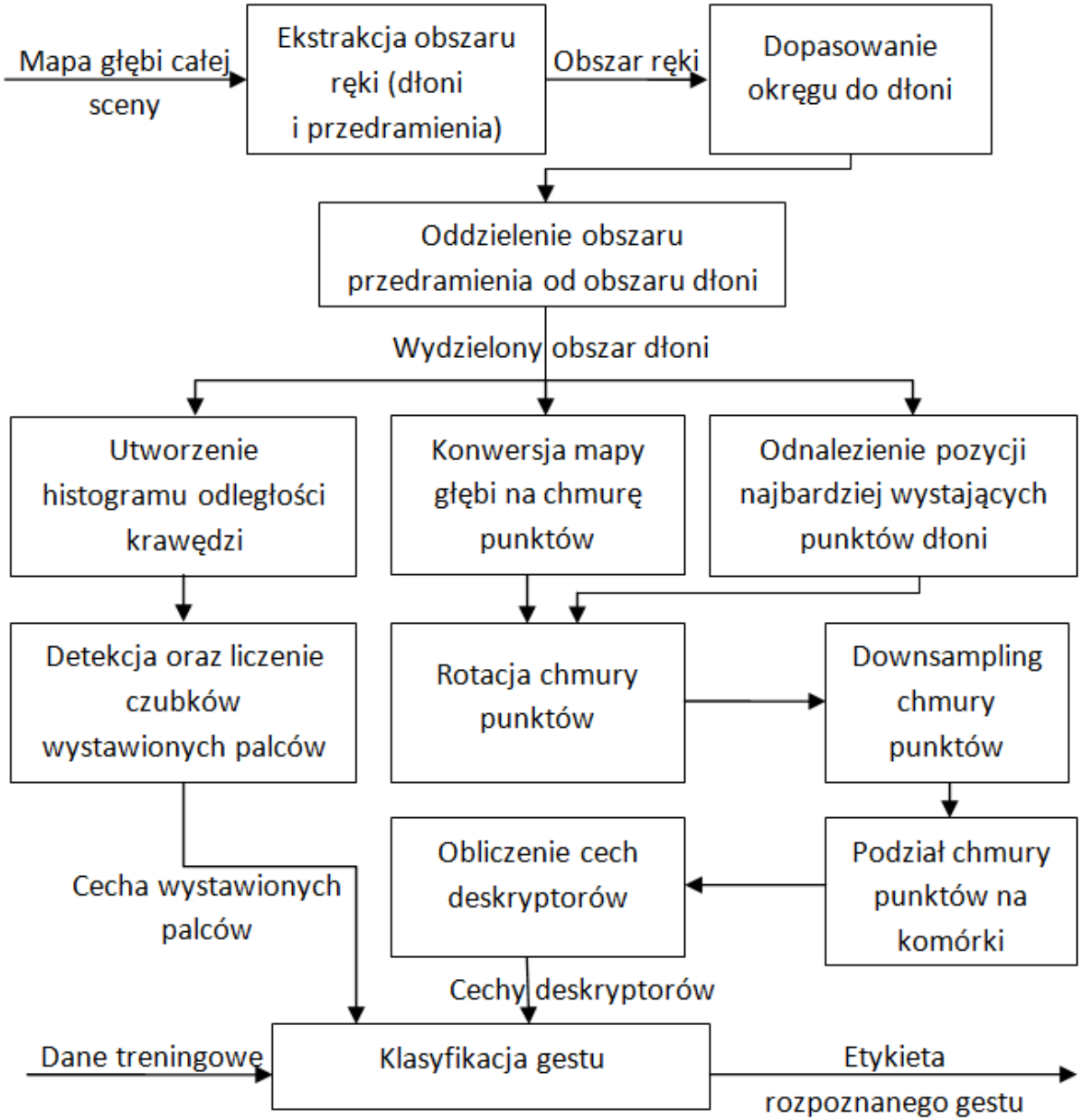
	Jedna komórka		9 komórek	
Deskryptor	**9u,1t**	**1u,9t**	**9u,1t**	**1u,9t**
cos(α)	50.0	44.3	93.0	89.9
cos(φ)	96.0	95.4	97.0	94.1
θ	63.0	60.5	91.0	82.9
|d|	69.0	59.5	99.0	94.1
Ψ	46.0	44.4	88.0	76.8

20 wykonań każdego gestu
10-krotna walidacja krzyżowa
u - l. podzbiorów uczących
t - l. podzbiorów testowych
32 kombinacje cech:
średnia i odch. stand. histogramu.
Kombinacja cos*φ, |d|*
100%(9u), 95.0%(1u) **1**kom.
100%(9u), 97.3%(1u) **9**kom.

Rozpoznawanie statycznych układów dłoni

1. Dłoń musi być obiektem najbliżej kamery.
2. Dłoń musi znajdować się w pewnej ustalonej odległości od tułowia/głowy.
3. Musi być widoczna przynajmniej niewielka część przedramienia.

Proponowana metoda rozpoznawania statycznych układów dłoni

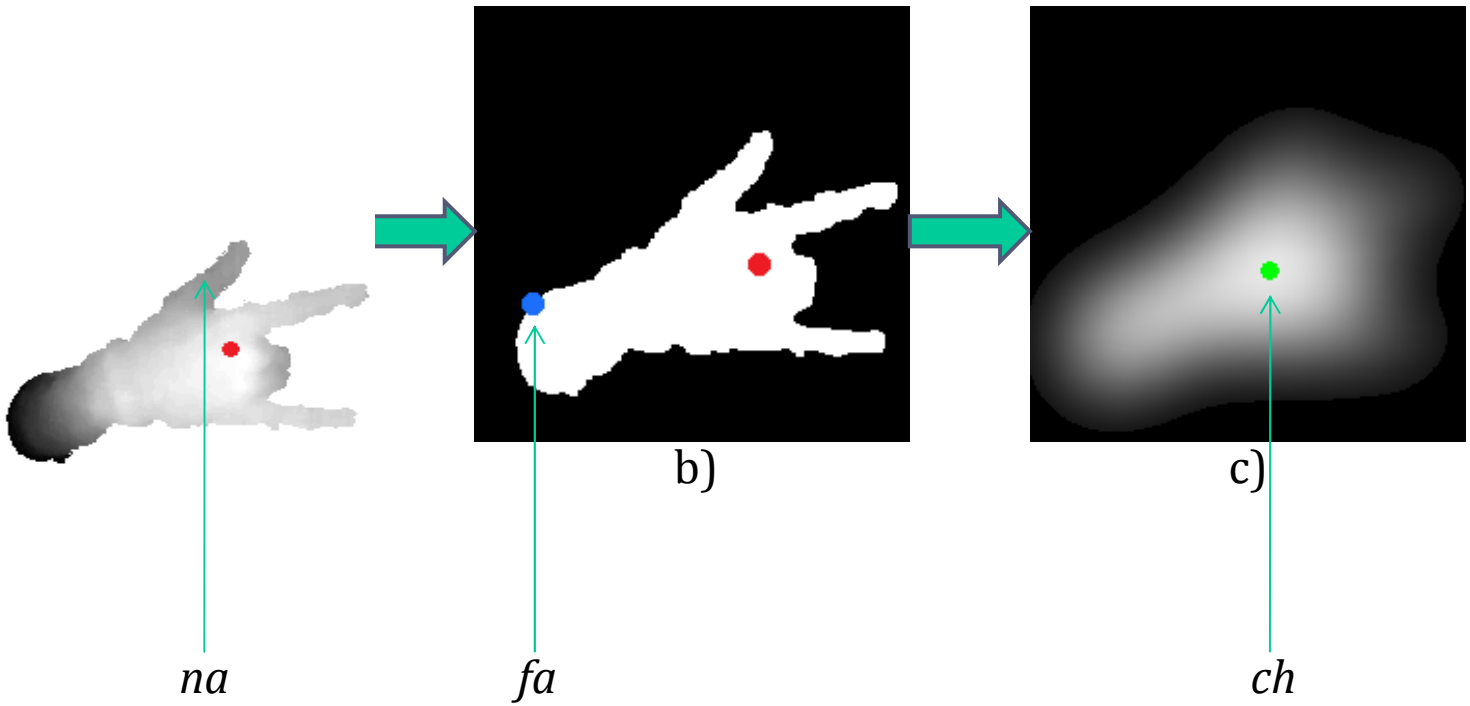


81

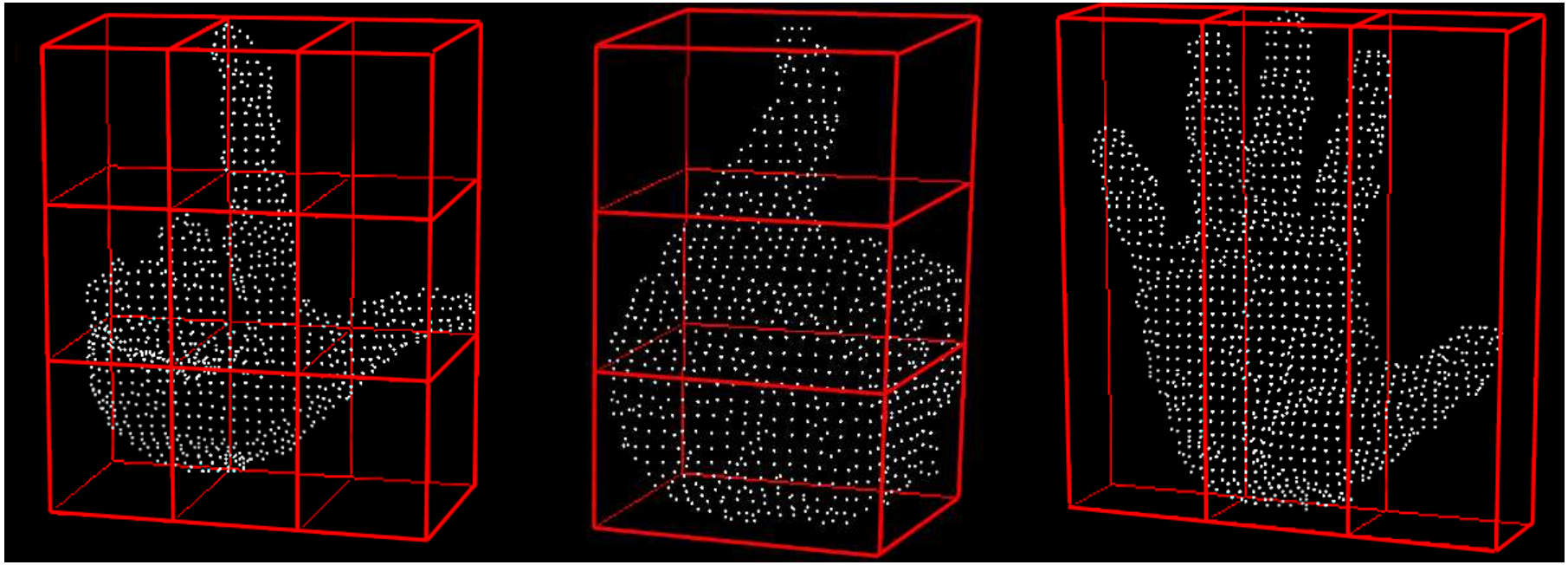
Dane wejściowe

Źródło bazy danych: *http://eeeweba.ntu.edu.sg/computervision/people/home/renzhou/HandGesture.htm*

Segmentacja dłoni, część I



Podział chmury punktów na komórki



9 komórek

3 horyzontalne komórki

3 wertykalne komórki

Eksperymenty dotyczące zastosowania deskryptora VFH w rozpoznawaniu układów dłoni

- Zbiór danych: 1000 map głębi: 10 gestów, 10 osób, 10 wykonań.
- Reprezentacja VFH: średnia i odchylenie standardowe histogramów.
- Klasyfikator: najbliższego sąsiada z odległością euklidesową.
- Testy: 5-krotna walidacja krzyżowa.

Wektor cech

Przykładowy wektor cech w przypadku podziału dłoni na 3 komórki. Wybrane zostały 2 cechy VFH: Φ i d reprezentowane przez średnią i odchylenie standardowe histogramów.

Komórka I				Komórka II				Komórka III			
m(Φ)	sd(Φ)	m(d)	sd(d)	m(Φ)	sd(Φ)	m(d)	sd(d)	m(Φ)	sd(Φ)	m(d)	sd(d)

m – średnia
sd – odchylenie standardowe

Eksperymenty – cechy deskryptora VFH

Założenie: 3 komórki horyzontalne

Cecha	Średnia poprawność klasyfikacji [%]
VFH(θ)	50.8
VFH(α)	40.6
VFH(φ)	62.2
VFH(d)	76.1

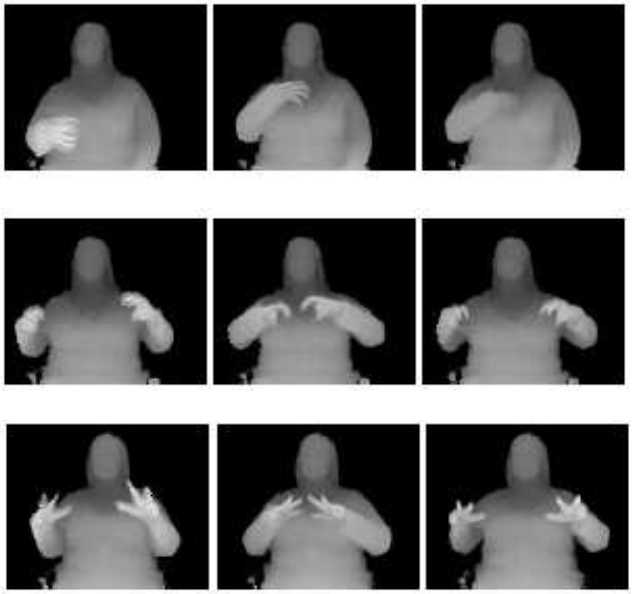
Najlepszy wynik dla VFH: połączenie wszystkich czterech cech (łącznie 24 cechy) - **92.3%**, czas potrzebny na wyznaczenie wektora cech na laptopie z procesorem Intel Core i7-4710HQ, taktowanie 2500-3500 MHz: **4.5ms.**

Włączenie cechy wyciągniętych palców (EF)

Metoda	Skuteczność [%]	Średni czas rozpoznawania [ms]
VFH, 9 komórek, **bez EF**	82.6	
VFH, 9 komórek	95.4	
VFH, 3 komórki H **bez EF**	88.9	56
VFH, 3 komórki H	**98.0**	57
VFH, 3 komórki W **bez EF**	83.4	
VFH, 3 komórki W	91.1	

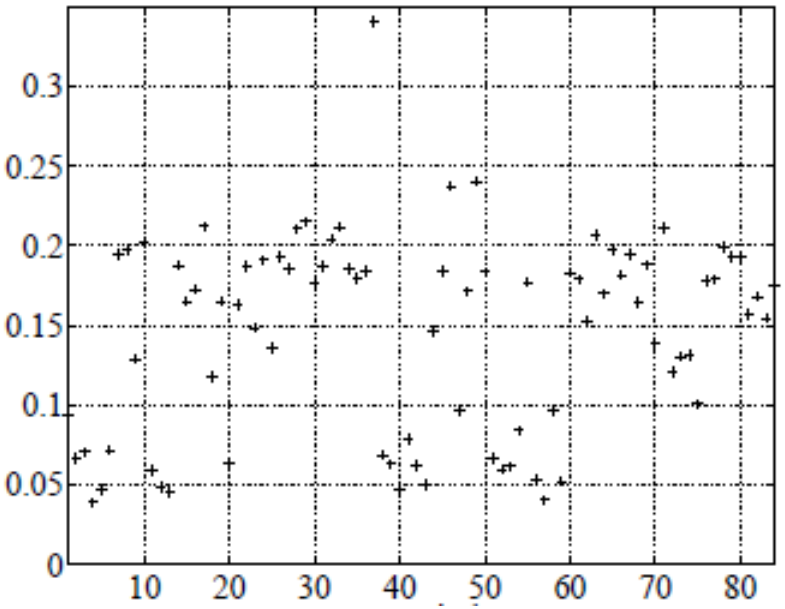
VFH: d,cos(φ)
Dominio, F., Donadeo, M., Zanuttigh, P: Combining multiple depth-based descriptors for hand gesture recognition. *Pattern Recognition Letters*, 2014, (**99%, 114 ms**)

Rozpoznawanie gestów dynamicznych



Mapy głębi dla przykładowych gestów dynamicznych na początku, w fazie pośredniej i na końcu; od góry: boleć, analiza, rodzinny.

Rozpoznawanie gestów dynamicznych



Rozrzut długości trwania gestów dynamicznych; zależność ilorazu (odch. stand./średnia) od wypowiadanego słowa oznaczonego tu numerem - 84 słowa powtarzane 20 razy.

Metoda DTW

Metoda nieliniowej transformacji czasowej (DTW - *Dynamic Time Warping*) służy do porównywania dyskretnych ciągów czasowych. Dwa ciągi czasowe

$$Q = \{q(1), q(2), \ldots, q(T_q)\} \text{ i } R = \{r(1), r(2), \ldots, r(T_r)\},$$

gdzie indeksy $1, 2, \ldots$ odpowiadają kolejnym równoodległym chwilom próbkowania, a elementy $q(\cdot)$ i $r(\cdot)$ są skalarami lub, w ogólniejszym przypadku, wektorami liczb rzeczywistych o jednakowych wymiarach, przekształca się za pomocą pewnej nieliniowej transformacji czasu tak, by odległość między nowymi przebiegami była minimalna.

Należy zauważyć, że proste sumowanie odległości między odpowiadającymi sobie punktami ciągów wyjściowych na ogół nie wchodzi w grę z dwóch powodów:

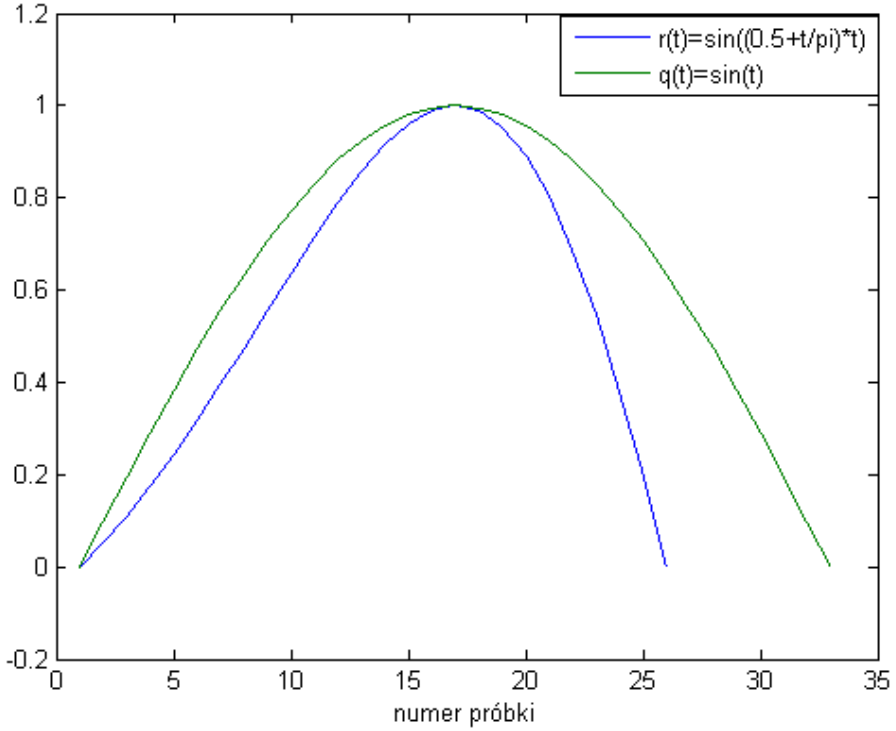
(1) przebiegi Q i R mają różne długości,

(2) liniowa zmiana skali czasu nie jest uzasadniona.

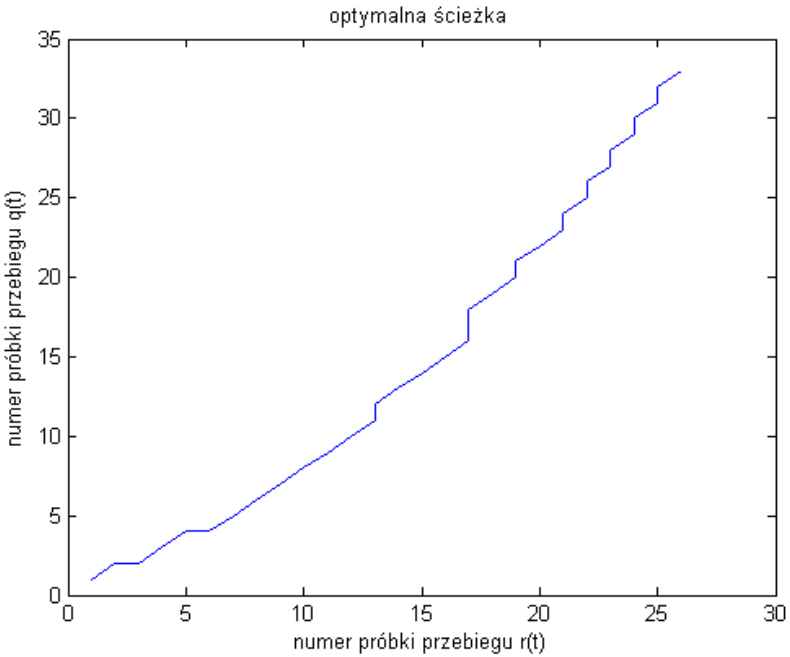
Przykład

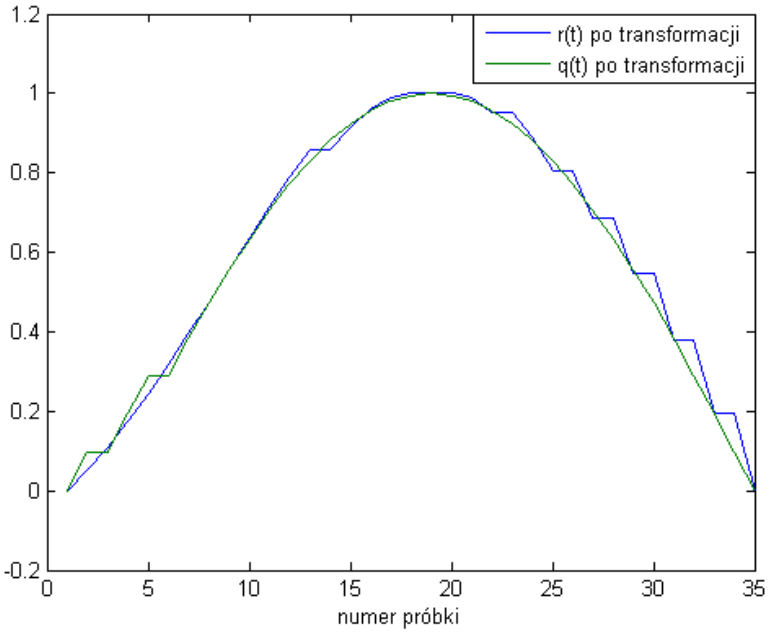
$$q(t) = \sin(t), \quad r(t) = \sin\left(t*(0.5+t/\pi)\right)$$

Oba przebiegi są próbkowane z okresem $\pi/32$. Ciąg Q zawiera 33 próbki, ciąg R ma ich 26.



Przykład





Obliczenia z oknem o szerokości 8

Metoda DTW

W wyniku zastosowania DTW powstają nowe przebiegi o jednakowej długości, a miara odległości między nimi może być interpretowana w naturalny sposób.

Metoda opiera się na zbudowaniu macierzy M o wymiarach T_q x T_r , gdzie elementem (i, j) macierzy jest odległość $d(q(i), r(j))$ między punktami $q(i)$ oraz $r(j)$.

Najczęściej wykorzystuje się odległość euklidesową.

Ścieżka transformacji

$$W = \{w_1, w_2, \ldots, w_K\}, \text{ gdzie } max\,(T_q, T_r) \leq K \leq T_q + T_r - 1,$$

jest zbiorem elementów macierzy *M*, których położenia spełniają trzy ograniczenia:

- **(i) brzegowe** - nakazuje, by ścieżka rozpoczynała się w punkcie $(1,1)$, a kończyła w punkcie (T_q, T_r),
- **(ii) ciągłości** - ogranicza dopuszczalne kroki na ścieżce do sąsiednich elementów macierzy,
- **(i) monotoniczności** - wymaga dodatkowo, by te sąsiednie elementy były opisane niemalejącymi wartościami indeksów i, j, a więc by odnosiły się do punktów przebiegów wyjściowych o niemalejących znacznikach czasu.

Metoda DTW

Poszukiwana jest ścieżka transformacji, która zapewnia minimum średniej z sumy wartości odpowiadających poszczególnym jej elementom, tzn.

$$d_{DTW} = \min \sum_{k=1}^{K} \frac{w_k}{K}$$

Do eleganckiego wyznaczenia optymalnej ścieżki wykorzystuje się metodę programowania dynamicznego. Aby uniknąć patologicznych sytuacji, kiedy względnie krótki fragment jednego z przebiegów wyjściowych zostanie dopasowany do długiego fragmentu drugiego przebiegu, wprowadza się dodatkowe ograniczenie na szerokość tzw. okna transformacji, które definiuje obszar poszukiwań jako zbiór komórek w wąskim pasku wokół przekątnej macierzy M łączącej początkowy i końcowy element ścieżki. To ograniczenie dodatkowo znacznie przyspiesza obliczenia.

Skuteczność rozpoznawania

Liczba kom.		Kinect				ToF			
		9u,1t		1u,9t		9u,1t		1u,9t	
		DTW	HMM	DTW	HMM	DTW	HMM	DTW	HMM
1	Mean	79.2	80.3	59.0	68.0	60.9	88.9	46.0	88.3
	Min	60.0	60.0	41.5	54.1	52.4	78.0	42.5	78.0
	Max	93.3	96.7	68.9	75.6	70.2	92.3	48.7	84.2
	StDev	12.3	10.2	8.7	7.5	5.4	4.5	2.1	2.2
9	Mean	99.7	98.7	97.4	96.9	99.9	99.0	98.8	97.9
	Min	97.7	93.3	88.1	90.4	99.4	94.6	98.2	96.6
	Max	100.0	100.0	99.6	99.3	100.0	100.0	99.3	99.0
	StDev	1.1	2.3	3.4	2.5	0.2	1.8	0.4	0.7

Dziesięciokrotna walidacja krzyżowa, 84 słowa, 20 powtórzeń;
u - liczba podzbiorów uczących, **t** - liczba podzbiorów testowych;
Cechy: cos(*α*), |*d* | - wartość średnia, odchylenie standardowe

Skuteczność rozpoznawania

Kąt obrotu	Cecha	DTW	HMM
-10^o	μ(cosα), σ(cosα), μ(|d|), σ(|d|)	86.1	98.9
	μ(cosα), μ(|d|)	77.8	97.4
	μ(cosα), σ(cosα)	85.5	98.0
	μ(cosα)	75.6	92.8
10^o	μ(cosα), σ(cosα), μ(|d|), σ(|d|)	75.5	99.0
	μ(cosα), μ(|d|)	63.5	91.2
	μ(cosα), σ(cosα)	65.9	98.3
	μ(cosα)	50.7	93.9

Uczenie na danych dla osoby ustawionej na wprost kamery

Wykorzystanie jednostek mniejszych niż słowa

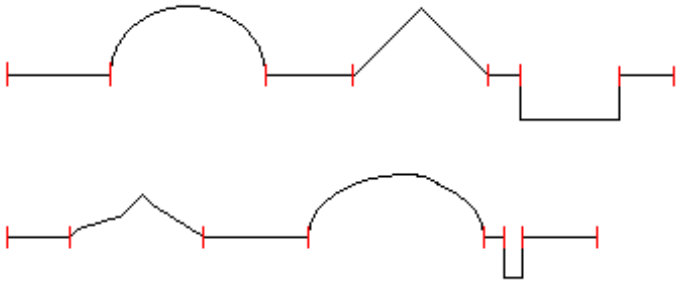
Model całych słów

Zbiór uczący złożoność uczenia wzrastają wraz z rozbudową słownika.

Rozwiązanie

Modelowanie słów za pomocą mniejszych jednostek (cheremów), co przypomina modelowanie za pomocą fonemów w przypadku mowy.

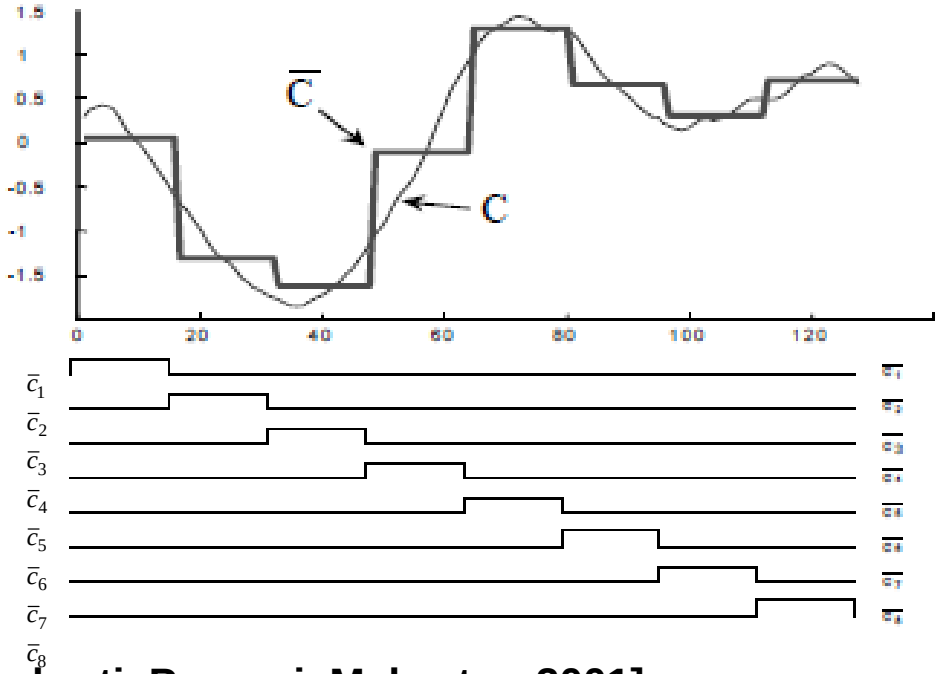
Rozbudowa słownika o nowe słowa wiąże się z modelowaniem gestów przez konkatenacje modeli cheremów.



Problem

Jak wyodrębnić cheremy?

Zagregowana aproksymacja odcinkowa (PAA)

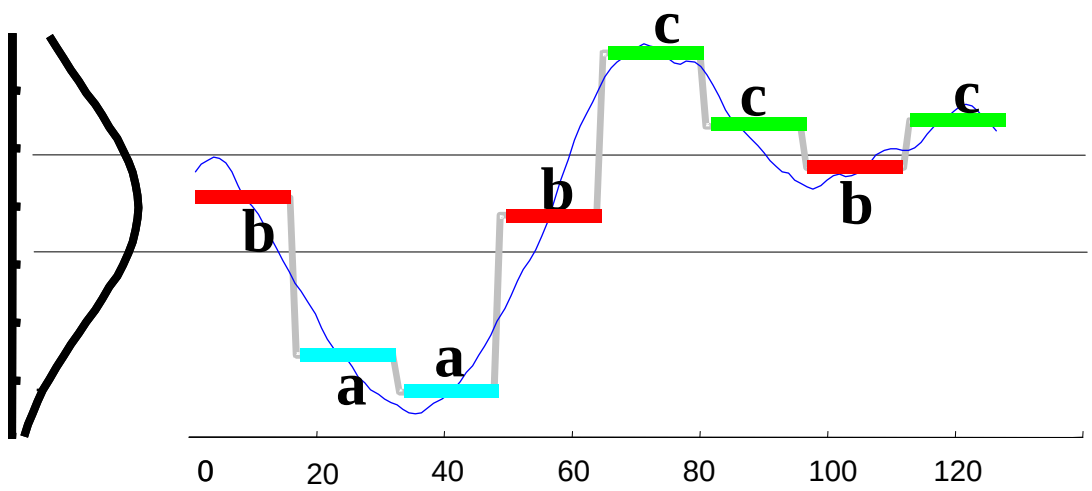


[Keogh, Chakrabarti, Pazzani, Mehrotra, 2001]:

Zagregowana aproksymacja odcinkowa (*Piecewise Aggregate Approximation* – PAA) służy do redukcji wymiarowości.

- Szereg czasowy, zawierający n danych, jest normalizowany tak, by średnia wartości była równa 0, a odchylenie standardowe 1 (powstaje przebieg C).
- Następnie dane są grupowane w *w* ramkach jednakowej długości ($w << n$), a każda ramka jest reprezentowana przez średnią zawartych w niej wartości. Przebieg C zostaje aproksymowany sumą funkcji schodkowych o amplitudach równych odpowiadającym średnim, tworzącą przebieg $\overline{C}$.

Zagregowana aproksymacja symboliczna (SAX)

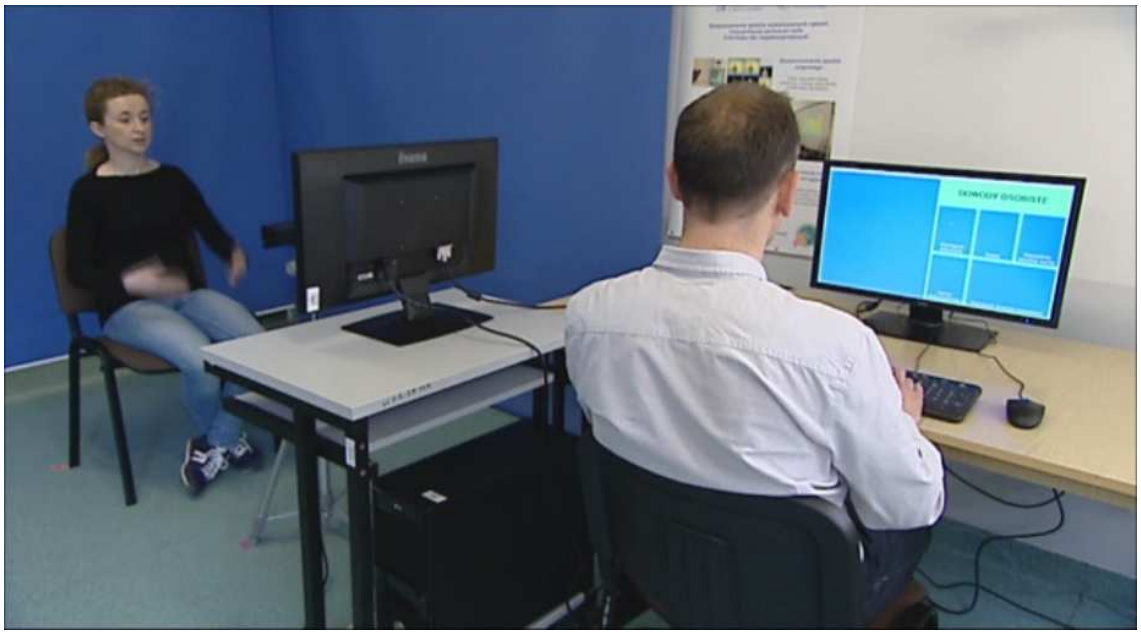


[Keogh, Lin, Host, 2005]

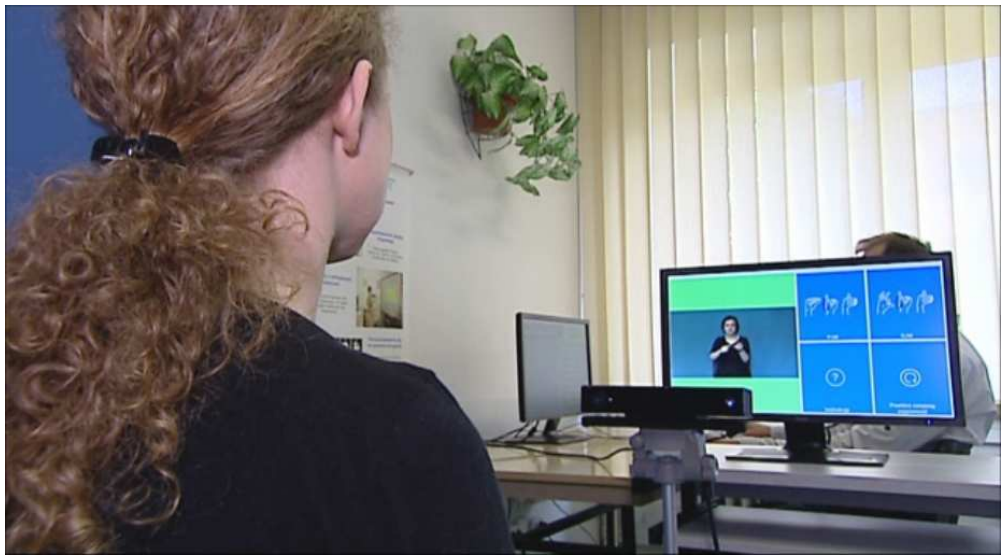
Z aproksymacji PAA można przejść do dyskretnej aproksymacji za pomocą łańcucha, wykorzystując α symboli (*Symbolic Aggregate aproXimation* – SAX). Badania eksperymentalne pokazują, że wartości znormalizowanych szeregów czasowych mają rozkład Gaussa, więc oś rzędnych dzieli się na α przedziałów, takich że prawdopodobieństwo wystąpienia wartości należącej do dowolnego z nich jest równe $1/\alpha$. Przebieg na rysunku jest scharakteryzowany przez wartości $n = 128$, $w = 8$, $\alpha = 3$.

System informatyczny wspierający komunikację w języku migowym w instytucjach użyteczności publicznej

(Projekt TANGO)



- Głuchoniema kobieta, obserwowana przez kamerę, posługuje się językiem migowym.
- Urzędnik otrzymuje tłumaczenie na swoim ekranie.
- Jego odpowiedzi są wyświetlane na ekranie osoby głuchoniemej jako tłumaczenie migowe w formie filmu.



Osoba głuchoniema otrzymuje na swoim ekranie film z przetłumaczoną na język migowy odpowiedzią urzędnika.

W razie potrzeby, tłumaczenie może być uzupełnione o przekazane przez urzędnika informacje graficzne (np. rysunek z dokumentem, plan pomieszczenia, itp.).

Wybrane publikacje

1. M. Wysocki, T. Kapuściński, J. Marnik, M. Oszust: ***Rozpoznawanie gestów wykonywanych rękami w systemie wizyjnym,*** Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2011.
2. M. Oszust, M. Wysocki: ***Recognition of Signed Expressions Observed by Kinect Sensor***, Proc of the 10th IEEE Int. Conf on Advanced Video and Signal-Based Surveillance, AVSS 2013, (IEEE Xplore).
3. M. Oszust, M. Wysocki: ***Recognition of Signed Expressions Using Symbolic Aggregate Approximation***. Lecture Notes in Artificial Intelligence, vol. 8467, (L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, A. Zadeh, J. Żurada, Eds.), Springer, 2014, 745-756.
4. T. Kapuściński, M. Oszust, M. Wysocki, D. Warchoł: ***Recognition of Hand Gestures Observed by Depth Cameras***, International Journal of Advanced Robotics Systems, 12, 36, 2015.
5. D. Warchoł, M. Wysocki: ***Recognition of Hand Postures Based on Point Cloud Descriptor and a Feature of Extended Fingers***, Journal of Automation, Mobile Robotics and Intelligent Systems, 10, 1, 2016, 48-57.

Dziękuję za uwagę!