UNIWERSYTET ZIELONOGÓRSKI
WYDZIAŁ ELEKTROTECHNIKI, INFORMATYKI
I TELEKOMUNIKACJI

mgr inż. Tomasz Kapuściński

ROZPOZNAWANIE POLSKIEGO JĘZYKA MIGANEGO W SYSTEMIE WIZYJNYM

Rozprawa doktorska

Promotor: dr hab. inż. Marian Wysocki, prof. PRz

Zielona Góra 2006

Pragnę wyrazić serdeczne podziękowanie
Panu Profesorowi Marianowi Wysockiemu
za wielką życzliwość i wszelką pomoc udzieloną
w czasie powstawania niniejszej pracy.

Tomasz Kapuściński

Spis treści

Rozdział 1

Wstęp

1.1 Motywacja

W ostatnich latach duże zainteresowanie wzbudzają komputerowe metody interpretacji działania i zachowań ludzi. Jeden z ważnych elementów badań dotyczy gestów wykonywanych rękami. Rozumienie przez maszynę wydawanych tak poleceń może stanowić atrakcyjne uzupełnienie obecnych sposobów komunikacji człowieka z komputerem (robotem). Systemy interpretacji gestów wykonywanych rękami mają szereg istotnych, potencjalnych zastosowań, jak np.:

- ułatwienie pracy oraz kontaktu z otoczeniem osobom niepełnosprawnym i w podeszłym wieku,
- zdalne sterowanie maszynami w środowisku uniemożliwiającym komunikację dźwiękową,
- poprawa warunków pracy z komputerem,
- działanie człowieka w środowisku wirtualnej rzeczywistości.

Niniejsza praca nawiązuje do pierwszego z wymienionych zagadnień i zmierza do opracowania metod i narzędzi informatycznych, które mogłyby pomóc osobom głuchoniemym w Polsce. Niemożność porozumiewania się za pomocą głosu stanowi dużą barierę w kontaktach społecznych. Należy dodać, że osoby niesłyszące od urodzenia lub od wczesnego dzieciństwa mają też spore trudności w wyrażaniu myśli na piśmie, bo nie zetknęły się z charakterystyczną dla ojczystego języka składnią i gramatyką. Dla nich język migowy, który jest ich językiem naturalnym, pozostaje podstawowym narzędziem komunikacji, zarówno między sobą jak i z osobami słyszącymi. Uzasadnione stają się więc badania zmierzające do zastosowania technik informatycznych w rozpoznawaniu języków migowych, co w konsekwencji w przyszłości umożliwiłoby budowę translatorów z języka migowego do mówionego lub pisanego. W pierwszej kolejności translator taki miałby ograniczony słownik wyrazów oraz ustalony zbiór predefiniowanych wypowiedzi, typowych w wybranej sytuacji życiowej, np. w firmie, u lekarza, na poczcie. Tak skonstruowany system, mógłby służyć także jako interfejs użytkownika do komputerów PC lub elektronicznych serwerów informacji, które

stają się obecnie stałym elementem dworców, terminali lotniczych i innych miejsc użyteczności publicznej. Byłby to krok w kierunku, zalecanego przez *Deklarację na temat Praw Osób Niepełnosprawnych* ONZ [83] oraz ratyfikowaną przez kraje członkowskie UE *Europejską Kartę Społeczną* [84], wyrównywania szans w dostępie osób upośledzonych do informacji i najnowszych technologii.

Innym zastosowaniem mogłaby być redukcja pasma komunikacyjnego przy przekazywaniu sygnału w języku migowym. Po stronie nadawcy przekaz migowy byłby przekształcany do formalnego opisu gestów, który to opis przesyłany byłby do odbiorcy i odtwarzany z wykorzystaniem grafiki komputerowej.

Doświadczenia z rozpoznawaniem języków migowych mogą być również pomocne w pracach nad wykorzystaniem gestów wykonywanych rękami w zagadnieniach interakcji człowieka z obiektami rzeczywistości wirtualnej i komunikacji z robotem.

1.2 Przegląd literatury

Komputerowe metody rozpoznawania gestów są intensywnie rozwijane na świecie. Prace w tym zakresie pojawiają się na konferencjach z zakresu wizji i robotyki oraz specjalnych konferencjach "Face and Gesture Recognition".

W zależności od sposobu pozyskiwania danych do rozpoznawania prace te możemy podzielić na takie, w których wykorzystuje się czujniki przymocowane do specjalnie skonstruowanych rękawic [17, 25, 33, 44, 50, 57, 62, 74, 75, 77, 78] oraz na rozwiązania z wykorzystaniem kamer [2, 3, 10, 26, 27, 30, 41, 49, 61, 68, 69, 80]. Zaletą zastosowania specjalnych czujników jest precyzja wynikająca z bezpośredniego pomiaru, wadą natomiast ograniczenie swobody użytkownika, który musi ubierać specjalne rękawice, najczęściej połączone z krępującym jego ruchy okablowaniem. W nowoczesnych rozwiązaniach powinno się dążyć do tego, aby komunikacja człowieka z komputerem odbywała się w sposób przypominający porozumiewanie się z drugą osobą.

W literaturze dostępne są publikacje dotyczące rozpoznawania języków migowych: amerykańskiego (ASL) [10, 25, 61, 75, 80], japońskiego (JSL) [30, 41, 49, 50, 57, 68, 69], niemieckiego (GSL) [3], chińskiego (CSL) [17, 77, 78], tajwańskiego (TSL) [27, 44, 62], holenderskiego (NSL) [2] i australijskiego (Auslan) [26, 33, 74]. Każdy język ma swoją własną specyfikę. Świadczy o tych chociażby fakt, że języki migowe: amerykański i angielski są zupełnie inne, pomimo iż język mówiony jest praktycznie ten sam. Jak wynika z wiedzy autora, nie istnieją dotychczas opracowania dotyczące rozpoznawania języka używanego w Polsce. Pokrewne zagadnienie rozpoznawania Polskiego Alfabetu Palcowego przedstawiono w pracy [47]. W pracach [63, 64] rozważano natomiast problem odwrotny polegający na graficznej syntezie sekwencji w Polskim Języku Miganym na podstawie zapisanej w formie tekstowej wypowiedzi w języku ojczystym.

Do rozpoznawania języków migowych wybierane były podzbiory gestów. Ich liczebność zestawiono w tab. 1.1. Najliczniejszy składający się z 5119 wyrazów CSL badano w pracy [77]. W pracy [78] rozpoznawano 274 wyrazów CSL, w [2] 262 wyrazów NSL, w [44] - 250 wyrazów TSL, [17] - 208 zdań CSl, [25] - 176 wyrazów

Tab. 1.1. *Charakterystyka układów rozpoznawania języków migowych*

praca	język	sł./zd.	wielkość słownika	dłonie	źródło danych	metoda klasyfikacji	skuteczność [%]
Hernandez, 2004 [25]	ASL	sł.	176	1	rc	inna[7]	94
Yang, 2002 [80]	ASL	sł.	40	2	k	NN[1]	96.2
Vogler, 2001 [75]	ASL	zd.	22	2	rc	HMM	95.5
Cui, 2000 [10]	ASL	sł.	28	1	k	inna[9]	93.2
Starner, 1998 [61]	ASL	zd.	40	2	k	HMM	98
Holden, 2000 [26]	Auslan	sł.	22	1	k + rk	inna[4]	95
Vamplew, 1998 [74]	Auslan	sł.	52	1	rc	NN[5]	94.2
Kadous, 1996 [33]	Auslan	sł.	95	1	rc	inna[7]	80
Wang, 2002 [77]	CSL	sł.	5119	2	rc	HMM	92.8
Fang, 2001 [17]	CSL	zd.	208	2	rc	HMM	92.1
Bauer, 2002 [3]	GSL	sł.	100	2	k + rk	HMM	92.5
Tanibata, 2002 [69]	JSL	sł.	65	2	k	HMM	100
Imagawa, 2000 [30]	JSL	sł.	33	2	k	inna[8]	94
Sagawa, 2000 [57]	JSL	zd.	17	2	rc	inna[10]	86.6
Kobayashi, 1997 [41]	JSL	sł.	6	1	k	HMM	98.8
Matsuo, 1997 [49]	JSL	sł.	38	2	k + rk	inna[4]	79
Murakami, 1991 [50]	JSL	sł.	10	1	rc	NN[3]	96
Tamura, 1988 [68]	JSL	sł.	10	1	k	inna[4]	45
Assan, 1997 [2]	NSL	sł.	262	2	k + rk	HMM	91.3
Su, 2000 [62]	TSL	sł.	90	2	rc	NN[6]	94.1
Huang, 1998 [27]	TSL	sł.	15	1	k	NN[2]	96
Liang, 1998 [44]	TSL	sł.	250	1	rc	HMM	84.7

sł. - pojedyncze słowa / zd. - zdania
rc - rękawice z czujnikami / rk - rękawice kolorowe / k - kamera
[1]z opóźnieniem, [2]Hopfielda, [3]rekurencyjna, [4]zbiór reguł, [5]perceptron wielowarstwowy,
[6]układ sieci generujący reguły rozmyte, [7]warunkowe dopasowanie wzorca, [8]PCA+klasteryzacja,
[9]PCA, MDA + drzewa decyzyjne, [10]dopasowanie do wzorca

ASL, natomiast w [3] 100 wyrazów GSL. W pozostałych znanych autorowi opracowaniach liczebność rozpatrywanego słownika była mniejsza od 100. Charakterystyczne jest, że w zadaniach z obszerniejszymi słownikami wykorzystywano rękawice z czujnikami [77, 78] lub kolorowe [2]. Wadą większości z dostępnych w literaturze opracowań jest to, że wybór słownika podyktowany był raczej prostotą aniżeli przydatnością i częstością występowania wybranych wyrazów i sekwencji w typowych sytuacjach życiowych. Nagranie odpowiednio licznej bazy danych jest zadaniem kluczowym i wymagającym sporego nakładu pracy. W dotychczasowych pracach bardzo rzadko wykorzystywano gesty wykonywane przez profesjonalistów, którzy posługują się językiem migowym na codzień i wykonują gesty w sposób spontaniczny. Często, a szczególnie w odniesieniu do liczniejszych słowników, opierano się na gestach przygotowanych przez jednego wykonawcę.

Znak migowy pokazany w sekwencji może odbiegać znacznie od wykonania pojedynczego, zwłaszcza w przypadku, gdy przekaz migowy ma charakter spontaniczny. Dlatego fakt, że dany wyraz jest bardzo dobrze rozpoznawany podczas wykonywania pojedynczo nie oznacza, że będzie on tak samo dobrze rozpoznawany w całym zdaniu. W przypadku rozpoznawania całych zdań pojawia się dodatkowy problem

segmentacji ciągłej sekwencji gestów. Pojedyncze wyrazy rozpoznawane były w pracach [2, 3, 10, 25, 26, 27, 30, 33, 41, 44, 49, 50, 62, 68, 69, 74, 77, 78, 80], całe sekwencje natomiast w [2, 17, 44, 57, 61, 75, 77].

W pracach [10, 25, 26, 27, 33, 41, 44, 50, 68, 74] rozważano jedynie gesty wykonywane jedną ręką, natomiast w pracach [2, 3, 17, 30, 49, 57, 61, 62, 69, 75, 77, 78, 80] uwzględniano także gesty dwuręczne.

Przy wyznaczaniu wektorów cech z wykorzystaniem układów wizyjnych przyjmowano szereg założeń wstępnych. Najczęściej zakładano, że osoba wykonująca gest ma ubranie z długim rękawem [2, 3, 27, 30, 61, 68, 69, 80]. Często osoba wykonująca gest miała kolorowe rękawice ułatwiające identyfikację dłoni i poszczególnych palców w obrazie [2, 3, 26, 49]. Zakładano, że tło musi być jednorodne [2, 3, 26, 27, 49, 68, 80] albo złożone lecz stacjonarne [10, 30, 61, 69]. Przyjmowano także, że głowa osoby wykonującej gest pozostaje nieruchoma, albo jej ruchy są znikome w porównaniu z ruchami dłoni [10, 27, 30, 61, 68, 69]. W pracy [10] dłonie muszą być w ciągłym ruchu, natomiast w pracach [61, 69] muszą one przyjmować pewną ustaloną pozycję początkową przed rozpoczęciem nagrywania. W metodach [10, 27, 68] przyjęto, że kamera obserwuje tylko jedną dłoń.

Kształty przyjmowane przez dłonie i ich ruchy mają charakter przestrzenny. W większości dostępnych w literaturze rozwiązań do wyznaczania wektorów cech stosowano jednak układy wizyjne z jedną kamerą. Układ stereowizyjny wykorzystano w pracach [26, 49].

Klasyfikację przeprowadzano najczęściej za pomocą różnych wariantów sztucznych sieci neuronowych [27, 50, 62, 74, 80] i ukrytych modeli Markowa [2, 3, 17, 41, 44, 61, 69, 75, 77]. Sieci neuronowe wykorzystywano przede wszystkim do rozpoznawania kształtu dłoni w gestach statycznych. Ponieważ większość znaków migowych to gesty dynamiczne, ich rozpoznawanie wiąże się z klasyfikacją szeregów czasowych. Tu zastosowanie sieci neuronowych jest trudniejsze. Spotyka się podejścia polegające na uwzględnieniu na wejściu sieci danych reprezentatywnych tylko dla trzech etapów, tj. początkowej, środkowej i końcowej fazy gestu [76]. Inne rozwiązania polegają na zastosowaniu sieci rekurencyjnej [50] lub sieci z opóźnieniem [80]. Jak widać z tab. 1.1, sieci neuronowe służyły raczej do rozpoznawania słów. W przypadku zdań konieczne jest uprzednie wyodrębnienie poszczególnych słów. Propozycję segmentacji szeregu czasowego opartą na detekcji zmiany kierunku ruchu dłoni przedstawiono w pracy [57].

Przeważająca liczba prac wykorzystuje ukryte modele Markowa. Jest to między innymi konsekwencją doświadczeń wynikających z szerokiego stosowania tego narzędzia w systemach rozpoznawania mowy, które potwierdzają, że ukryte modele Markowa dobrze sprawdzają się w zadaniach modelowania i rozpoznawania szeregów czasowych, wykazując właściwości automatycznego dokonywania nieliniowej transformacji czasowej i segmentacji. Skuteczności rozpoznawania uzyskane w układach rozpoznawania języków migowych zestawiono w tab. 1.1.

Na podstawie przeglądu literatury można sformułować następujące wnioski.

- Poza niektórymi intuicyjnymi znakami o uniwersalnym zastosowaniu języki migowe używane w różnych krajach są znacząco odmienne.

- W literaturze nie podaje się na ogół wyczerpujących informacji dotyczących szczegółów rozwiązań. Trudno też a priori ocenić uniwersalność rozwiązań publikowanych w odniesieniu do konkretnych języków.
- Zachodzi potrzeba badań nad rozpoznawaniem języka migowego używanego w Polsce, a w pierwszej kolejności zaproponowania metody pozwalającej na rozpoznawanie izolowanych słów i prostych zdań wykorzystywanych w typowej sytuacji życiowej i opracowania jej najistotniejszych elementów, popartego praktyczną weryfikacją w rzeczywistym systemie, najlepiej umożliwiającym bezpośrednią interakcję użytkownika z komputerem. Spośród istotnych zagadnień należy wskazać następujące:
  - wybór i sposób wyznaczania wektora cech,
  - syntezę klasyfikatora rozpoznającego pojedyncze wyrazy,
  - syntezę klasyfikatora zdań, z uwzględnieniem problemów przejść między sąsiednimi wyrazami i możliwości rozpoznawania nowych, nieznanych zdań wykorzystujących znane wyrazy,
  - przygotowanie reprezentatywnej bazy danych uwzględniającej z jednej strony praktyczną przydatność rozważanego słownika, z drugiej zaś możliwie szeroką gamę elementów rzutujących na trudność rozpoznawania, włącznie z wykonywaniem gestów przez osoby biegle posługujące się językiem migowym,
  - przygotowanie narzędzi programowych i uruchomienie stanowiska badawczego pozwalającego na weryfikację zaproponowanych rozwiązań oraz stanowiącego bazę do kontynuacji badań,
  - zasygnalizowanie kierunków tych badań.

1.3 Cel i zakres pracy

Celem pracy jest opracowanie i weryfikacja metody przybliżającej zbudowanie systemu wizyjnego do rozpoznawania słów i zdań Polskiego Języka Miganego (PJM), stanowiącego w Polsce podstawową formę komunikacji osób z uszkodzeniem narządu słuchu ze środowiskiem słyszących. Język ten powstał w wyniku ujednolicenia znaków migowych stosowanych w rożnych regionach kraju i dodania zasad gramatyki języka polskiego [24, 65, 66]. W pracy położono nacisk na wybór i wyznaczanie wektorów cech, konstrukcję klasyfikatorów opartych na teorii ukrytych modeli Markowa oraz eksperymentalną ocenę ich przydatności. Źródło informacji stanowiły obrazy ze stereowizyjnego układu kamer kolorowych. Przyjęto, że nie będą wykorzystywane żadne środki pomocnicze (np. rękawice z różnokolorowymi palcami). Założono, że osoba wykonująca gest stoi przodem do kamery w stałej odległości od niej, a w tle nie pojawiają się inne osoby. Skoncentrowano się na rozpoznawaniu zamkniętego słownika wyrazów i zdań występujących w wybranej sytuacji życiowej: u lekarza i na poczcie. W niniejszej pracy przyjęto, że gest utożsamiany jest z przedstawieniem

wyrazu, a sekwencja gestów z przedstawieniem zdania. Ograniczono się do manualnych środków wyrazu, pomijając informacje przekazywaną ruchami ust, głowy, torsu czy układem ciała. Założono także, że rozpatrywane wypowiedzi nie będą, wymagały użycia znaków Polskiego Alfabetu Palcowego. Integralnym elementem pracy było przygotowanie narzędzi programowych i bazy danych.

Słowa PJM są gestami dynamicznymi prawie zawsze wykonywanymi dwiema rekami. Występuje wtedy ruch jednej dłoni (tzw. dominującej), podczas gdy druga pozostaje nieruchoma, albo też obie ręce poruszają się. Podczas wykonywania gestów dłonie mogą się wzajemnie dotykać i przysłaniać oraz pojawiać na tle twarzy. Ruch może być jednokrotny albo powtarzany.

Tezę pracy można sformułować następująco: **Z użyciem ukrytych modeli Markowa możliwe jest rozpoznawanie pojedynczych słów i zdań Polskiego Języka Miganego na podstawie sekwencji wizyjnych. Jeżeli spełnione są ograniczenia odnośnie do ubioru i otoczenia osoby przedstawiającej, gesty ze zbioru wykorzystywanego w typowej sytuacji życiowej mogą być rozpoznawane z wysoką skutecznością w układzie pozwalającym na bezpośrednią interakcję z komputerem.**

Założony cel pracy i uzasadnienie tezy wyznaczają następujące zadania badawcze:

- zaproponowanie schematu przetwarzania obrazów pozyskiwanych w stereowizyjnym układzie kamer kolorowych w celu wyznaczenia wektorów cech,
- opracowanie i eksperymentalna weryfikacja metody identyfikacji dłoni i twarzy w obrazach kolorowych pod kątem jakości obrazów binarnych otrzymywanych w zmiennych warunkach oświetlenia oraz czasu przetwarzania umożliwiającego zastosowanie metody w trybie on-line,
- wybór metody wyznaczania mapy głębi na podstawie obrazów stereo z uwzględnieniem jakości otrzymywanych map oraz czasu przetwarzania pozwalającego na zastosowanie w trybie on-line,
- opracowanie algorytmu identyfikacji dłoni prawej, lewej i twarzy w otrzymanych obrazach binarnych,
- analiza wyników badań lingwistycznych dotyczących cech dystynktywnych znaków migowych, zaproponowanie wektora cech wykorzystywanego przez klasyfikator,
- opracowanie i eksperymentalna weryfikacja metody rozpoznawania wybranych wyrazów PJM z wykorzystaniem ukrytych modeli Markowa,
- opracowanie i weryfikacja metody rozpoznawania zdań PJM z wykorzystaniem ukrytych modeli Markowa.

Część badawcza pracy dotyczy przetwarzania i rozpoznawania obrazów cyfrowych i sekwencji wizyjnych, ze szczególnym uwzględnieniem zastosowania ukrytych modeli

Markowa w zadaniach interakcji człowiek-maszyna ukierunkowanych na interpretację przez komputer gestów wykonywanych rękami. Oczekiwanym praktycznym rezultatem jest zaś zbiór narzędzi programowych o ogólniejszym przeznaczeniu oraz powstała z ich wykorzystaniem aplikacja umożliwiająca rozpoznawanie wybranych wyrazów i zdań PJM w trybie on-line, a także obszerna baza sekwencji wizyjnych, która może być przydatna także innym badaczom.

1.4 Przegląd pracy

Podstawę pracy stanowią cztery dalsze rozdziały. Rozdział 2 zawiera zwięzłe omówienie kluczowych zagadnień dotyczących Polskiego Języka Miganego. Przedstawiono charakterystykę PJM zwracając szczególną uwagę na zagadnienia ważne ze względu na rozpoznawanie w układzie wizyjnym. Większość znaków PJM to gesty dynamiczne, w których integralnym elementem jest ruch. Wykonywane są na ogół w obrębie klatki piersiowej, twarzy i pasa. W przedstawieniu znaku bierze udział jedna ręka (niekiedy na podstawie utworzonej z drugiej ręki, która nie ma czynnego udziału w ruchu), albo obie ręce, poruszające się symetrycznie lub asymetrycznie. Bardziej precyzyjne zdefiniowanie sposobu wykonywania poszczególnych gestów ułatwia tzw. zapis gestograficzny, który w odniesieniu do PJM jest wynikiem stosunkowo nowych badań. Omówienie tego zapisu pozwoliło przedstawić przyjęte do eksperymentów gesty. Poszczególne wyrazy i zdania zostały starannie wybrane po konsultacji z osobami zajmującymi się tłumaczeniem PJM w Polskim Związku Głuchych w Rzeszowie. Wybór uwzględniał, oprócz przydatności w typowych sytuacjach życiowych (u lekarza i na poczcie) także reprezentatywność pod względem sposobu wykonywania i złożoności rzutującej na trudność rozpoznawania w układzie wizyjnym. Wzięto więc pod uwagę przypadki, gdy dochodzi do wzajemnego przysłaniania się dłoni lub pojawienia się dłoni na tle twarzy, gdy zachodzi podobieństwo niektórych gestów do innych, bądź utrata części informacji w wyniku projekcji z przestrzeni trójwymiarowej do płaszczyzny obrazu.

W rozdziale 3 przedstawiono schemat przetwarzania obrazów pozyskiwanych w układzie stereowizyjnym w celu wyznaczenia wektorów cech. Dokonano wyboru metody identyfikacji dłoni i twarzy w obrazie kolorowym, przyjmując jako kryterium jakość otrzymanych obrazów binarnych i czas przetwarzania. Opisano eksperymenty z wykorzystaniem kilku metod identyfikacji i wybranych przestrzeni barw. Przedstawiono także zaproponowany przez autora algorytm pozwalający na identyfikację dłoni prawej, lewej i twarzy w otrzymywanych obrazach binarnych oraz dokonano wyboru metody generowania mapy dysparycji. Wybór poprzedzono eksperymentami z różnymi metodami, biorąc pod uwagę jakość otrzymywanych map dysparycji oraz czas przetwarzania. Starano się tak dobrać poszczególne metody, aby możliwe było przetwarzanie sekwencji wizyjnych w trybie on-line z częstotliwością 25 klatek/s. W końcowej części rozdziału zaproponowano warianty wektorów cech, na których oparto później budowę klasyfikatorów. Wyboru cech dokonano wzorując się na wynikach badań lingwistycznych nad tzw. cechami dystynktywnymi znaków migowych.

Rozdział 4 opisuje ukryte modele Markowa (HMM), które zastosowano w niniejszej pracy do rozpoznawania wyrazów i zdań PJM. Przedstawiono tutaj matematyczny opis ukrytych modeli Markowa, omówiono w jaki sposób mogą być używane do rozpoznawania wyrazów i zdań oraz opisano równoległy ukryty model Markowa, którego zastosowanie okazuje się uzasadnione w przypadku rozpoznawania przekazu migowego, mającego charakter równoległy. Najmniejsze, niepodzielne jednostki różnicujące, będące odpowiednikami fonemów z języka mówionego, mogą w tym przypadku zmieniać się równolegle. Duża część gestów migowych wykonywana jest dwiema rękami, podczas ruchu dłoni jednocześnie może zmieniać się jej kształt i orientacja. Zwrócono uwagę na algorytm Viterbiego w wersji z przekazywaniem znaczników, który znacznie ułatwia rozpoznawanie z wykorzystaniem sieci równoległych modeli Markowa. Przedstawiono metodę uczenia z wbudowanym wyodrębnianiem elementów składowych w ciągu uczącym, np. wyrazów w zdaniu (*embedded training*), która jest szczególnie użyteczna w przypadku rozpoznawania sekwencji gestów, ponieważ nie wymaga precyzyjnej segmentacji zdań wykorzystanych w uczeniu, a jedynie informacji o kolejności występujących w nich wyrazów.

Rozdział 5 dotyczy rozpoznawania pojedynczych słów PJM. Opisano tutaj wyniki rozpoznawania podzbioru 101 wyrazów PJM występujących w typowych sytuacjach: u lekarza i na poczcie. Gesty wykonywane były przez lektorkę PJM oraz przez autora, który wyuczył się ich na użytek niniejszej pracy. Łącznie wykorzystano 6060 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przebadano 32 różne warianty wektora cech, uwzględniając przypadki, gdy uczenie i testowanie odbywało się na gestach tej samej osoby, uczenie i testowanie odbywało się na gestach różnych osób oraz gdy zbiór uczący został zbudowany z wykonań poszczególnych gestów przez obie osoby. W rozdziale przeanalizowano także kwestię doboru liczby stanów HMM oraz fuzji klasyfikatorów opartych na różnych wektorach cech.

W rozdziale 6 opisano wyniki rozpoznawania 35 zdań pokazanych w wariancie użytkowym PJM, występujących w typowych sytuacjach: u lekarza i na poczcie. Zdania wykonywane były przez lektorkę PJM oraz autora pracy. Łącznie wykorzystano 1400 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przebadano 15 różnych konfiguracji sieci ukrytych modeli Markowa uwzględniając statystyczny model lingwistyczny bigram oraz modele przejść między wyrazami składającymi się na rozpatrywaną wypowiedź. Przeprowadzono także eksperymenty z wykorzystaniem równoległych ukrytych modeli Markowa oraz eksperyment uogólniający polegający na odrzuceniu części lub wszystkich sekwencji ze zbioru uczącego i uczeniu modeli tylko na wykonaniach pojedynczych wyrazów.

Podsumowanie wyników oraz kierunki dalszych badań przedstawiono w rozdziale 7.

Uzupełnienie pracy stanowią dodatki: (A, B, C, D) omawiające oprogramowanie, które składa się na przygotowany przez autora system gromadzenia danych i rozpoznawania, (E) przedstawiający rozpoznawane wyrazy i zdania oraz (F) opisujący zawartość załączonej płyty DVD z wybranymi elementami bazy danych oraz wynikami eksperymentów.

Rozdział 2

Polski Język Migany

Podstawową formą porozumiewania się osób niesłyszących pomiędzy sobą jest klasyczny język migowy, zwany też naturalnym, swoistym lub tradycyjnym językiem migowym. Klasyczny język migowy wytworzony został przez środowisko niesłyszących do zaspokojenia naturalnej potrzeby porozumiewania się. Jest on więc dla tego środowiska językiem naturalnym, tak jak język mówiony dla osób słyszących. Bogaty zasób znaków migowych, obejmujący także pojęcia abstrakcyjne sprawia, że nie ma problemów w porozumiewaniu się między sobą osób niesłyszących. Bariera komunikacyjna pojawia się dopiero wtedy, gdy języki naturalne porozumiewających się osób są różne, a więc w sytuacji, gdy np. osoba z uszkodzeniem słuchu próbuje porozumieć się z osobą słyszącą. Dlatego w poszczególnych krajach opracowano sztuczne, bimodalne systemy komunikacji nazywane systemami językowo-migowymi [66]. W systemach tych ta sama treść przekazywana jest równolegle z wykorzystaniem kanału głosowego i znaków migowych. Dzięki dwukanałowej percepcji odbierane elementy wzajemnie się uzupełniają i wzmacniają. Część niefoniczna takiego przekazu nazywana jest językiem miganym. Język migany wykorzystuje znaki migowe z klasycznego języka migowego i zasady gramatyczne z ojczystego języka mówionego.

2.1 Ogólna charakterystyka Polskiego Języka Miganego

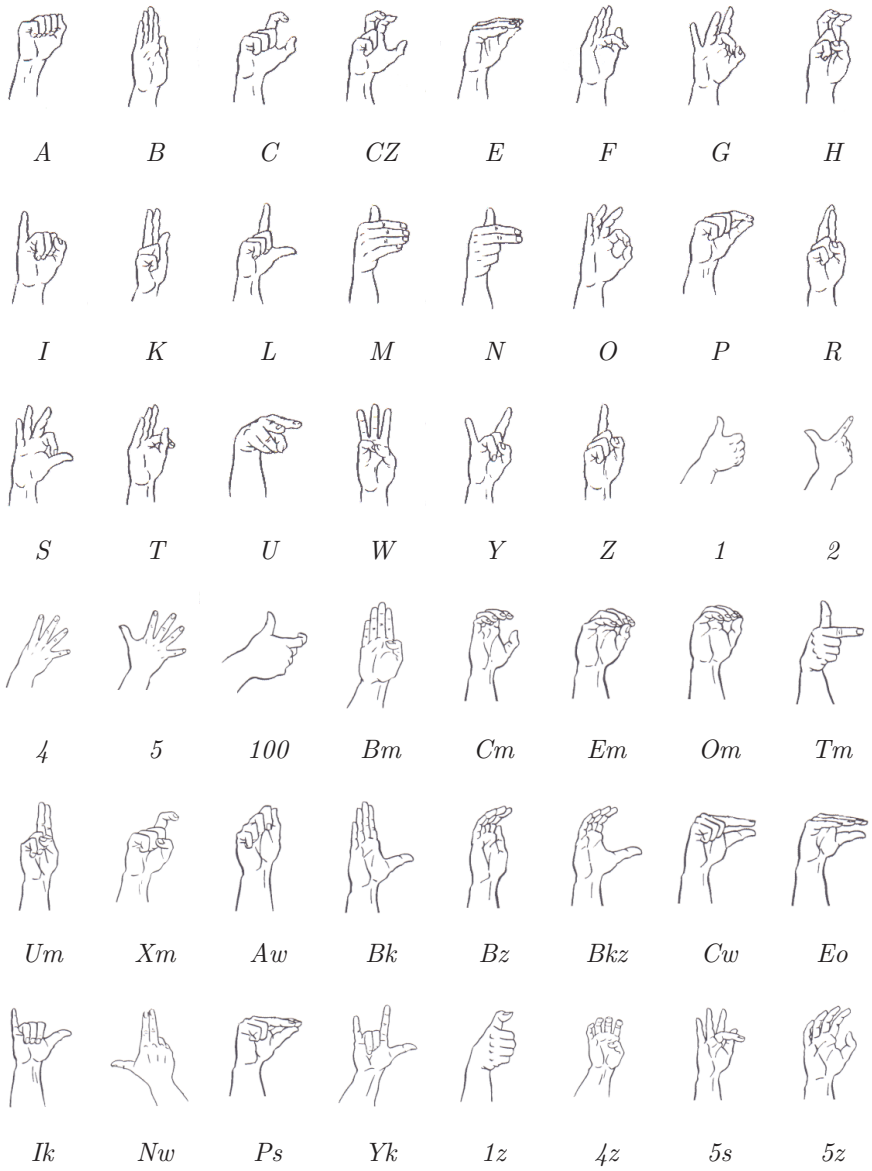
Polski Język Migany (PJM) został opracowany w latach 1964-65. Dokonano ujednolicenia znaków migowych stosowanych w różnych regionach kraju i dodano zasady gramatyki języka polskiego. PJM używany jest w sytuacjach formalnych, w tłumaczeniach programów telewizyjnych, na konferencjach i w nauczaniu dzieci niesłyszących. W PJM gest odpowiada jednemu wyrazowi. Zdania buduje się łącząc poszczególne gesty w sekwencje. Wyróżnia się dwa warianty przekazu: system językowo-migowy pełny i system językowo-migowy użytkowy. W systemie pełnym przekazuje się poszczególne wyrazy wraz z końcówkami fleksyjnymi. Zaletą tego przekazu jest dokładność i precyzja sprzyjające kształtowaniu prawidłowego myślenia językowego, wadą zaś tempo o około 30% wolniejsze od normalnego mówienia.
W wariancie użytkowym używa się tylko form podstawowych przekazywanych za po-

mocą znaku ideograficznego bez końcówek fleksyjnych. Dla prawidłowego funkcjonowania systemu językowo-migowego konieczna jest synchronizacja mowy i przekazu migowego. Dlatego w porozumiewaniu się osób niesłyszących znacznie większe zastosowanie ma wariant użytkowy, gdyż nie pozwala na opóźnienia toru miganego w stosunku do mowy. Wariant ten został przyjęty w niniejszej pracy.

W PJM nośnikiem informacji jest znak migowy. Wyróżniamy znaki migowe ideograficzne i daktylograficzne. Znaki ideograficzne odpowiadają poszczególnym słowom lub niekiedy krótkim zwrotom. Na znaki daktylograficzne składa się: alfabet palcowy, znaki liczebników głównych i porządkowych, znaki ułamków zwykłych i dziesiętnych oraz znaki interpunkcyjne [24]. Znaki ideograficzne stanowią główną część przekazu migowego, podczas gdy znaki daktylograficzne pełnią funkcję pomocniczą i uzupełniającą. Ponieważ znaki ideograficzne nie odmieniają się przez przypadki i osoby, struktury gramatyczne wyrażane są za pomocą odpowiedniego szyku wyrazów i luźnych gestów pomocniczych. Tłumaczenie wypowiedzi z klasycznego języka migowego na ojczysty język foniczny nie może zatem odbywać się słowo po słowie i wymaga uwzględnienia odmienności gramatyk obu języków. Nieco ponad 30% znaków Polskiego Języka Miganego to znaki ikoniczne czyli takie, w których układ dłoni i palców jest podobny do opisywanego kształtu albo ruch wykonywany dłońmi jest zbliżony do ruchu charakterystycznego dla opisywanej czynności. Około 40% stanowią znaki ezoteryczne, w których analogie do opisywanego przedmiotu albo czynności nie są łatwo dostrzegalne ale istnieją i mogą zostać wytłumaczone. Pozostałe 30% to znaki arbitralne, dla których nie można dopatrzeć się żadnych analogii do rzeczywistości. Gest migowy może zostać scharakteryzowany za pomocą tzw. cech dystynktywnych, do których zaliczymy: kształt dłoni, orientację dłoni i palców, położenie dłoni, charakter wykonywanego ruchu i udział drugiej ręki. Kształt, orientacja i położenie określają statyczną konfigurację dłoni.
Kształt dłoni określa położenie palców względem siebie. W PJM wyróżnia się 48 różnych układów palców dla każdej z rąk (rys. 2.1). 22 układy dłoni odpowiadają znakom Polskiego Alfabetu Palcowego (PAP). Są to znaki liter: *A*, *B*, *C*, *CZ*, *E*, *F*, *G*, *H*, *I*, *K*, *L*, *M*, *N*, *O*, *P*, *R*, *S*, *T*, *U*, *W*, *Y* i *Z*. Pięć konfiguracji odpowiada znakom liczebników głównych: 1, 2, 4, 5 i 100. Siedem układów dłoni to znaki międzynarodowego alfabetu palcowego oznaczone w literaturze dodatkową literą *m*. Są to znaki: *Bm*, *Cm*, *Em*, *Om*, *Tm*, *Um* i *Xm*. Dziesięć kształtów dłoni powstaje w wyniku modyfikacji liter polskiego alfabetu palcowego. Zmodyfikowane znaki oznacza się: *Aw*, *Bk*, *Bz*, *Bkz*, *Cw*, *Eo*, *Ik*, *Nw*, *Ps* i *Yk*. Pozostałe cztery układy to zmodyfikowane znaki liczebników głównych, oznaczane w literaturze odpowiednio: 1z, 4z, 5s i 5z. Teoretycznie, dla dwóch dłoni daje to więc $48^2 = 2304$ możliwości. W praktycznym przekazie migowym wykorzystuje się około 200 różnych kształtów dłoni. Przy rozpoznawaniu gestów migowych układy dłoni są często grupowane w pewne klasy kształtów, np.: piąstka, dłoń płaska, pojedynczy wysunięty palec, itp.

Do jednoznacznego określenia ułożenia dłoni w stosunku do pionu i poziomu konieczne jest określenie jej orientacji oraz orientacji poszczególnych palców. W PJM istnieją 32 różne ułożenia dłoni, co oznacza $32^2 = 1024$ różnych orientacji dla gestów dwuręcznych. W praktyce wykorzystuje się około 180 możliwości.

Położenie dłoni, zwane też miejscem artykulacji, określa się podając część ciała,



Rys. 2.1. *Kształty dłoni występujące w Polskim Języku Miganym*

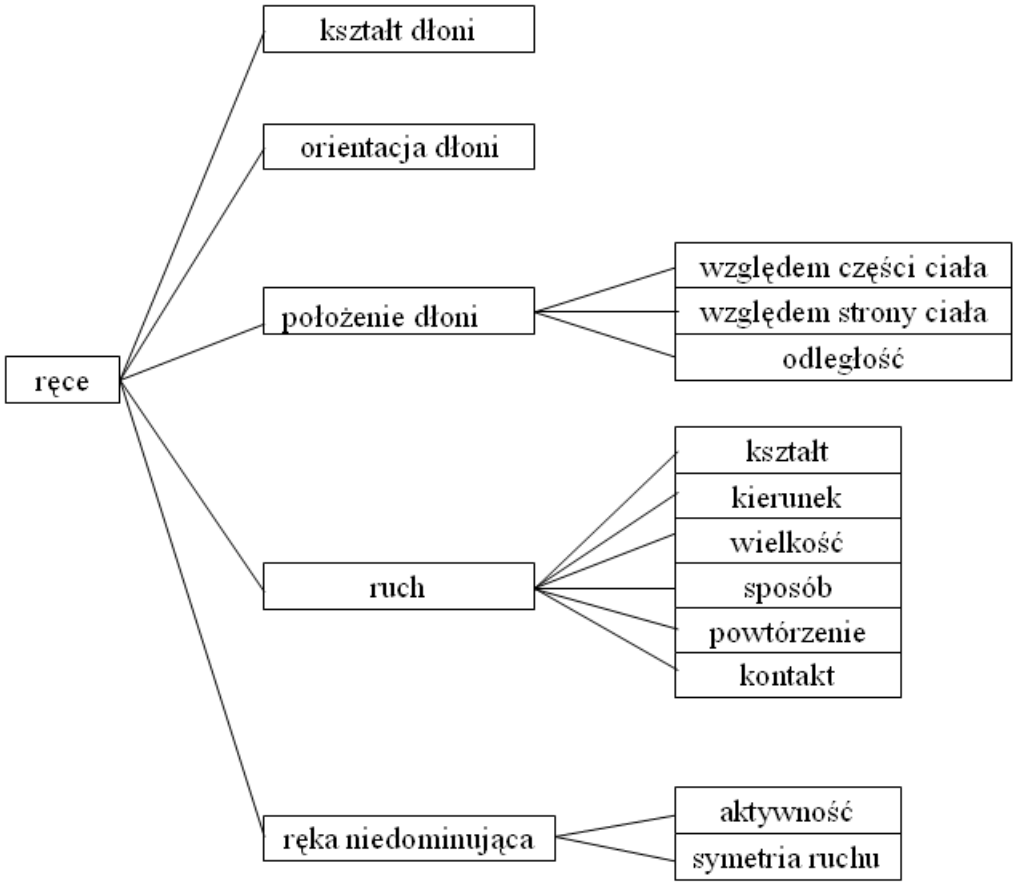
na tle której wykonywany jest gest. Gest może być wykonywany na tle klatki piersiowej, twarzy, brzucha. Informację tę uzupełnia się precyzując, po której stronie ciała odbywa się artykulacja (na prawo, na lewo, w środku). Podaje się także odległość dłoni od ciała. W PJM można wyróżnić 37 pozycji dla każdej z rąk: 17 przed lub obok twarzy, 18 przed klatką piersiową i dwie w dolnej części ciała. Każde z tych miejsc może być mniej lub bardziej wysunięte do przodu, co zwiększa liczbę możliwości do 74. Ponadto 26 miejsc może być w kontakcie dotykowym z ciałem. Łącznie daje to 100 możliwości i wszystkie one wykorzystywane są w praktyce. W PJM ponad 78% znaków migowych wykonywane jest w obrębie klatki piersiowej, 17% w okolicach twarzy, prawie 3% jednocześnie w obrębie twarzy i klatki piersiowej i niecałe 2% w innych miejscach.

Blisko 98% znaków PJM ma charakter dynamiczny. Występujący w nich ruch opisywany jest za pomocą przybliżonego kształtu trajektorii, kierunku, wielkości, sposobu wykonania, prędkości. Określa się także, czy podczas wykonywania ruchu występuje powtarzanie i kontakt dłoni z inną częścią ciała. Niekiedy podaje się także przegub, w którym ma miejsce ruch. Decydującymi parametrami są kształt i kierunek. Kształt najczęściej przybliża się za pomocą linii prostej, linii prostej z załamaniem pod kątem 45, 90 lub 135 stopni, fali, okręgu, łuku lub spirali. Kierunek ruchu definiuje się co 45 stopni. W znakach dynamicznych PJM wyróżnia się około 100 możliwych ruchów, które można opisać za pomocą 21 ruchów prostych i ich kombinacji. Sposób wykonania ruchu może być zróżnicowany. Ruch może być dłuższy lub krótszy od standardowego. Prędkość wykonania może być większa lub mniejsza od typowej. Podczas wykonywania ruchu dłoń może pozostawać w ciągłym kontakcie z drugą dłonią bądź z inną częścią ciała. Zetknięcie może mieć także charakter chwilowy i występować najczęściej w początkowej lub końcowej fazie ruchu. Po zakończeniu ruchu dłoń może zatrzymywać się w sposób energiczny lub łagodnie. Niekiedy złożona sekwencja ruchu może być kilkakrotnie powtarzana.

Gesty mogą być wykonywane jedną lub dwiema rękami. W przypadku gestów dwuręcznych wyróżnia się dłoń dominującą i niedominującą. Przy wykonywaniu gestu może obowiązywać reguła dominacji lub symetrii. W przypadku reguły dominacji obie dłonie mają różny kształt. Dłoń niedominująca przyjmuje wtedy jeden z kształtów z pewnego nielicznego zbioru i pozostaje nieruchoma, stanowiąc podstawę dla poruszającej się dłoni dominującej. W przypadku gestów symetrycznych obie dłonie przyjmują ten sam kształt i poruszają się w sposób symetryczny. W zależności od rodzaju symetrii rozróżniamy gesty równoległe, lustrzanie symetryczne, punktowo symetryczne i naprzemienne.

Zestaw parametrów opisujących manualną część przekazu migowego pokazano na rys. 2.2.

Po uwzględnieniu wszystkich możliwych kombinacji cech dystynktywnych otrzymujemy $2304 * 1024 * 100 * 100 = 23592960000$ możliwych konfiguracji znaków migowych. Dla porównania aparat foniczny człowieka może wytworzyć zaledwie około 90 różnych głosek.



Rys. 2.2. *Opis manualnej części przekazu migowego.*

2.2 Zapis gestograficzny

Język mówiony można łatwo zapisać za pomocą alfabetu fonetycznego. Znaki migowe nie mają odpowiedników w formie pisemnej a ich opis słowny musi być rozbudowany i nie zawsze jest jednoznaczny. Do precyzyjnego odwzorowania gestu migowego konieczne jest zatem wykorzystanie nagrania wideo albo animacji komputerowej. Dlatego opracowywane są metody opisu, w których znak określa się podając zapisane schematycznie wartości wszystkich cech dystynktywnych. Taki sposób opisu nazywamy zapisem gestograficznym, zaś opis pojedynczego znaku gestogramem.

Polski zapis gestograficzny opracowano w 1988 roku [65] i zmodyfikowano w 1998 roku, w trakcie prac na stworzeniem translatora tekstu pisanego na język migowy [63]. W zapisie tym, odpowiadający pojedynczemu znakowi gestogram opisany jest za pomocą jednej, dwóch, trzech lub czterech części rozdzielonych znakami '#'.

W części pierwszej zdefiniowana jest statyczna konfiguracja dłoni. Dla znaków dynamicznych jest to konfiguracja na początku gestu. Obejmuje ona kształt, orientację i położenie.

Opis kształtu składa się z dużej litery L lub P, oznaczającej odpowiednio dłoń

lewą bądź prawą, po której występuje jedno- lub dwuznakowe oznaczenie jednego z 48 możliwych kształtów (patrz rys. 2.1). Przed literą L lub P może pojawić się dodatkowo mała litera p, gdy następuje zetknięcie dłoni z przedramieniem lub r w przypadku, gdy dłoń dotyka ramienia.

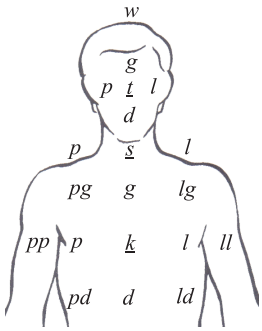
Orientację dłoni i palców zapisuje się po znaku ':' za pomocą liczby dwucyfrowej, w której pierwsza cyfra określa orientację zgrubnie, natomiast druga ją precyzuje. Znaczenie poszczególnych cyfr podano w tab. 2.1. Jeżeli drugą cyfrą zapisu jest 1,

Tab. 2.1. *Opis orientacji dłoni w zapisie gestograficznym*

pierwsza cyfra	znaczenie pierwszej cyfry
1	dłoń poziomo, wnętrzem w górę
2	dłoń poziomo, wnętrzem w dół
3	dłoń pionowo, końce palców skierowane w górę
4	dłoń pionowo, końce palców skierowane w dół
5	dłoń pionowo, kantem w dół
6	dłoń pionowo, kantem w górę
7	dłoń ukośnie w górę, oś dłoni pod kątem 45°
8	dłoń ukośnie w dół, oś dłoni pod kątem 45°
druga cyfra	**znaczenie drugiej cyfry**
1	końce palców skierowane do przodu
2	końce palców skierowane skośnie do przodu i w górę, wnętrzami dłoni do siebie
3	końce palców skierowane skośnie do przodu i do wewnątrz (przedramiona po kątem 90°)
4	końce palców skierowane do tyłu
5	kant dłoni skierowany do przodu
6	kant dłoni skierowany do tyłu
7	wnętrze dłoni skierowane do przodu
8	grzbiet dłoni skierowany do przodu
9	kanty dłoni skierowane do wewnątrz (lewej ręki w prawo, a prawej w lewo)
0	kanty dłoni skierowane na zewnątrz (prawej ręki w prawo, a lewej w lewo)

5, 7 lub 8, to może być przed nią umieszczony znak '/', który oznacza odchylenie dłoni w prawo lub znak '\' oznaczający odchylenie dłoni w lewo.

Położenie dłoni w stosunku do ciała opisuje się za pomocą jednej lub kliku małych liter umieszczonych bezpośrednio po opisie orientacji. Znaczenie poszczególnych liter przedstawiono na rys. 2.3. Pierwsza litera (na rysunku podkreślona) oznacza część ciała, przed którą znajduje się dłoń. Może to być litera t dla twarzy, s dla szyi lub k dla klatki piersiowej. Jeżeli dłoń ustawiona jest przed środkiem danej części ciała, to jednoliterowy opis pozycji jest wystarczający. W przeciwnym razie dodaje się następne litery, które precyzują przesunięcie dłoni. Litera p oznacza przesunięcie w prawo, l - w lewo, g - w górę i d - w dół. Jeżeli dłoń znajduje się poza obrębem



Rys. 2.3. *Opis pozycji dłoni względem ciała w zapisie gestograficznym*

danej części ciała, to literę określającą jej przesunięcie powtarza się. W niektórych znakach jako druga litera może wystąpić także *w* oznaczające wierzch głowy oraz *t* oznaczające tył głowy. W przypadku gdy występuje zetknięcie dłoni z drugą dłonią lub z inną częścią ciała, do opisu konfiguracji statycznej dodaje się znak '+'.

Dla gestów dwuręcznych należy jeszcze określić wzajemne położenie dłoni względem siebie. W tym celu wykorzystuje się znaki interpunkcyjne (tab. 2.2) umieszczone pomiędzy opisami dłoni prawej i lewej.

Tab. 2.2. *Opis wzajemnego położenia dłoni w zapisie gestograficznym*

zapis	wzajemne położenie dłoni
L)(*P*	obie dłonie obok siebie w bezpośrednim styku płaszczyznami lub kantami
L }{ *P*	obie dłonie obok siebie stykają się punktowo lub na małej powierzchni
L . *P*	obie dłonie obok siebie oddalone o ok. 15 cm
L .. *P*	obie dłonie obok siebie oddalone o ok. 30 cm
L ... *P*	obie dłonie daleko od siebie (40 cm lub więcej)
L X *P*	obie dłonie krzyżują się w przegubach pod kątem prostym
L x *P*	obie dłonie krzyżują się wyprostowanymi palcami pod kątem prostym
L ++ *P*	dłoń lub palce wsuwają się między palce drugiej dłoni z zetknięciem się
L +|+ *P*	dłoń lub palce wsuwają się między palce drugiej dłoni bez dotykania się
L)| *P*	lewa dłoń przed prawą w bezpośrednim styku płaszczyznami lub kantami
L }| *P*	lewa dłoń przed prawą w bezpośrednim styku punktowo lub na małej powierzchni
L | *P*	lewa dłoń przed prawą w odległości ok. 15 cm
L || *P*	lewa dłoń przed prawą w odległości ok. 30 cm
P)/ *L*	prawa dłoń na lewej w bezpośrednim styku płaszczyznami lub kantami
P }/ *L*	prawa dłoń na lewej w bezpośrednim styku punktowo lub na małej powierzchni
P X)/ *L*	prawa dłoń na lewej w bezpośrednim styku skrzyżowane w przegubach
P x}/ *L*	prawa dłoń na lewej w bezpośrednim styku skrzyżowane wyprostowanymi palcami
P / *L*	prawa dłoń nad lewą o ok. 15 cm
P // *L*	prawa dłoń nad lewą o ok. 30 cm

W tabeli 2.2 zapisy gestograficzne kształtu, orientacji i położenia obu dłoni zastąpiono dla uproszczenia literami *L* i *P*.

Dla znaków statycznych zapis gestograficzny złożony jest tylko z jednej części.

Na rys. 2.4 przedstawiono wybrane gesty statyczne PJM wraz z odpowiadającymi im gestogramami.

głowa: $PZ{:}25tppg+$ *ja*: $PZ{:}54k+$ *leżeć*: $PU{:}13k)/LBk{:}13k$ *w*: $LBk{:}58k{+}{+}PBk{:}23k$

Rys. 2.4. *Wybrane gesty statyczne PJM wraz z odpowiadającymi im gestogramami*

Druga część zapisu gestograficznego opisuje kierunek i sposób wykonania ruchu. Proste ruchy dłoni oznacza się cyframi rzymskimi (tab. 2.3). Ruchy złożone opisuje się łącząc cyfry z tab. 2.3 za pomocą znaku '\', gdy ruch jest wypadkową ruchów prostych, '\\', w przypadku gdy ruchy proste są wykonywane niezależnie od siebie i ';', gdy ruchy proste występują sekwencyjnie.

Tab. 2.3. *Opis prostych ruchów w zapisie gestograficznym*

zapis	kierunek ruchu
I	ruch w górę
II	ruch w dół
III	ruch do przodu
IV	ruch do tyłu
V	ruch w prawo
VI	ruch w lewo
VII	ruch na zewnątrz (rozchodzenie się rąk)
VIII	ruch do wewnątrz (schodzenie się rąk)
IX	ruch po okręgu poziomym
X	ruch po okręgu pionowym
XI	ruch po łuku poziomym
XII	ruch po łuku pionowym
XIII	ruch wahadłowy dłoni w przegubie
XIV	ruch wahadłowy palców (swobodne poruszanie palcami)
XV	ruch obu dłoni razem w tym samym kierunku
XVI	ruch obu dłoni na przemian (mijanie się)
XVII	ruch tam i z powrotem
XVIII	ruch obrotowy względem osi przedramienia
XIX	ruch falisty
XX	ruch pocierania palcami (kciuk o pozostałe)
XXI	nakreślenie krzyża w powietrzu (najpierw linia pionowa)

W przypadku gdy ruch wykonywany jest w sposób nietypowy, umieszcza się po jego opisie jeden lub kilka znaków z tab. 2.4.

Dla niektórych bardziej złożonych ruchów nie podaje się kierunku i sposobu wykonania ruchu, lecz kolejne konfiguracje statyczne oddzielone znakami '##'.

Tab. 2.4. *Opis nietypowych ruchów w zapisie gestograficznym*

zapis	sposób wykonania ruchu
$>$	ruch większy niż przeciętny
$\gg$	ruch znacznie większy niż przeciętny
$<$	ruch mniejszy niż przeciętny
$\ll$	ruch znacznie mniejszy niż przeciętny
$\geq$	ruch szybszy niż przeciętny
$\leq$	ruch wolniejszy niż przeciętny
!	ruch z energicznym zatrzymaniem ręki
+	ruch z dotknięciem drugiej ręki lub innej części ciała
-	ruch wzdłuż innej części ciała (przesuwanie)
=	ruch tam i z powrotem
”	ruch wykonany dwukrotnie w tym samym miejscu

Trzecia część zapisu gestograficznego zawiera opis końcowej konfiguracji statycznej dłoni. Część ta występuje tylko wtedy, gdy konfiguracja ta jest inna od konfiguracji początkowej.

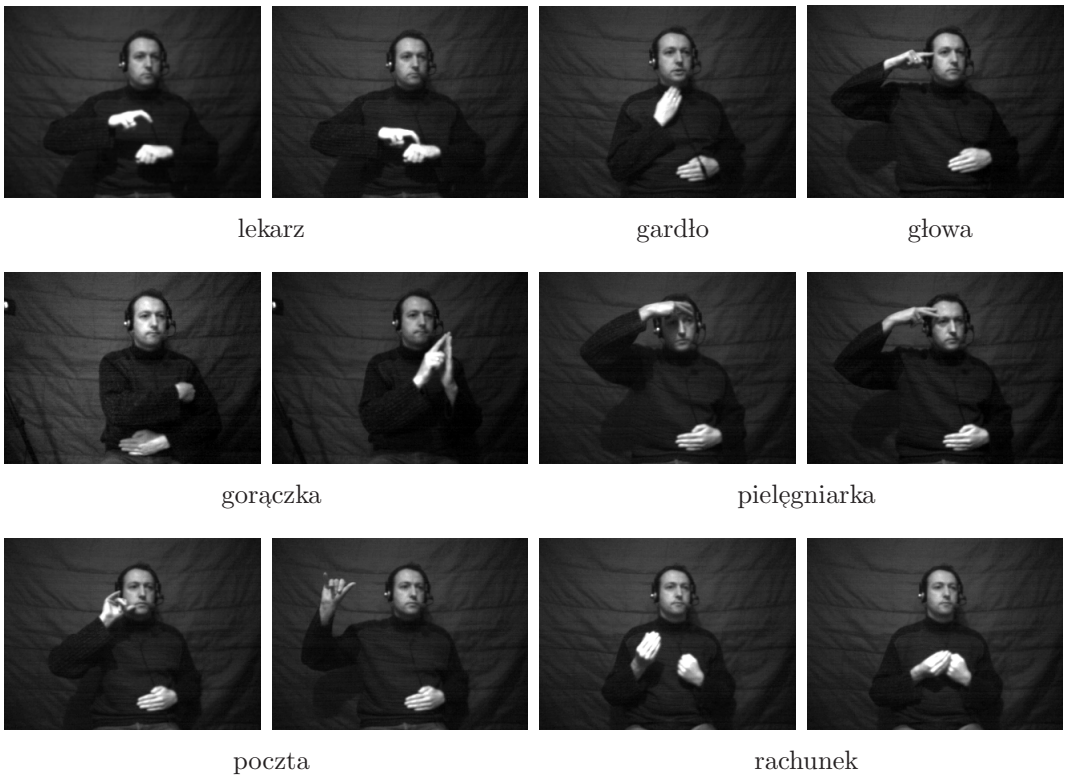
Jeżeli z wykonywanym gestem związana jest odpowiednia mimika twarzy, do opisu gestograficznego dodaje się czwartą część złożoną ze znaku '@'.

Zapis gestograficzny pozwala na jednoznaczne odwzorowanie około 98% znaków migowych. Tylko w nielicznych przypadkach ruch dłoni jest zbyt złożony, aby uzyskać wierny opis. Oznacza to, że użyte w nim cechy dystynktywne mogłyby być podstawą do zbudowania wektora cech wykorzystywanego w automatycznym rozpoznawaniu. Trudność polega jednak na tym, że w układzie wizyjnym następuje redukcja części informacji i wiązka cech, która dla człowieka jednoznacznie identyfikuje dany gest, może tutaj utracić swe właściwości dystynktywne. Ten sam przestrzenny kształt dłoni będzie w przetworzonym obrazie wyglądać zupełnie inaczej w zależności od orientacji dłoni i ustawienia osoby względem kamery. Dlatego bez przeprowadzenia dostatecznej liczby eksperymentów trudno wskazać wektor cech wystarczający do rozpoznawania znaków migowych.

2.3 Przykładowe wyrazy i zdania

Do rozpoznawania wybrano 101 wyrazów i 35 zdań używanych w typowych sytuacjach życiowych: u lekarza i na poczcie. Wyboru dokonano przy udziale konsultantki z Polskiego Związku Głuchych, która jest lektorem PJM i na co dzień uczestniczy we wspomnianych sytuacjach pełniąc rolę tłumacza. Przy wyborze wyrazów zwracano uwagę nie tylko na ich aspekt praktyczny i częstość występowania ale także na to, aby stanowiły one reprezentatywny podzbiór znaków migowych. 72 spośród wybranych znaków wykonywane jest w obrębie klatki piersiowej, 15 w obrębie twarzy, 11 w obrębie twarzy i klatki piersiowej oraz 3 w okolicach szyi. 88 znaków wiąże się z ruchem dłoni, zaś 13 ma charakter statyczny. Obie dłonie uczestniczą w wykonywaniu 62 znaków, pozostałe 39 wyrazów to gesty jednoręczne. Przykładowe słowa

PJM przedstawiono na rys. 2.5. Dla znaków dynamicznych pokazano początkową i końcową fazę gestu.



Rys. 2.5. *Wybrane słowa PJM*

Zdania tworzone były według zasady obowiązującej w wariancie użytkowym systemu językowo-migowego z wykorzystaniem tylko podstawowych form wyrazów bez końcówek fleksyjnych. Najkrótsze zdanie złożone było z dwóch a najdłuższe z 10 wyrazów. Przykładowe zdanie przedstawiono na rys. 2.6. Wykaz wszystkich rozpoznawanych wyrazów wraz z ich opisami gestograficznymi oraz listę wszystkich rozważanych zdań wraz z ich transkrypcjami do form użytych w układzie rozpoznającym zamieszczono w dodatku E.

2.4 Problemy związane z rozpoznawaniem PJM w układzie wizyjnym

Każdy obraz jest dwuwymiarowym odwzorowaniem trójwymiarowej sceny. Brakuje mu więc informacji o głębi. Na podstawie obrazu 2D nie jest możliwe jednoznaczne określenie położenia dłoni w przestrzeni oraz dokładne odwzorowanie trajektorii wykonywanej przez dłoń. Każde przemieszczenie się lub obrót osoby wykonującej gest prowadzi do odmiennego widoku na obrazie. Ponadto różne kształty dłoni mogą mieć to samo odwzorowanie 2D.

Rys. 2.6. *Zdanie "Pan jest chory."w wariancie użytkowym PJM*

Większość gestów PJM ma charakter dynamiczny i wykonywana jest dwiema rękami, dlatego bardzo częste są sytuacje, w których dłonie przysłaniają się wzajemnie, albo pojawiają się na tle twarzy. Utrudnia to zadanie identyfikacji dłoni i twarzy w obrazie.

Gesty wykonywane są w sposób subiektywny z charakterystycznym dla danej osoby "akcentem". Różnice w wykonaniu mogą objawiać się w wielkości i szybkości ruchu, sposobie realizacji kształtu dłoni, nieznacznych przesunięciach miejsca artykulacji, itp. Wykonania danego gestu przez tę samą osobę także mogą się różnić, zależnie od nastroju bądź kontekstu.

W przypadku rozpoznawania zdań zachodzi konieczność segmentacji ciągłej sekwencji gestów w celu wyodrębnienia poszczególnych wyrazów. Utrudnione jest to przez zjawisko koartykulacji polegające na tym, że w końcowej fazie danego gestu dłoń zaczyna już przygotowywać się do wykonania gestu następnego. Objawia się to zniekształceniami kształtu i ruchu w początkowych i końcowych fazach gestów wykonywanych w sekwencji. Dodatkowy ruch mający charakter spontaniczny pojawia się także wtedy, gdy miejsca artykulacji kolejnych gestów są od siebie odległe.

Układ dłoń-ramię ma wiele stopni swobody, co wpływa na złożoność możliwych ruchów. Dłoń musi być widziana jako ciało niesztywne z mniej lub bardziej silnymi zmianami kątów w stawach palców w zależności od gestu. Kształt dłoni może także zmieniać się w trakcie wykonywania gestu, gdy wynika to z jego specyfiki.

Spory problem w rozpoznawaniu gestów wykonywanych rękami wiąże się z tym, że badania nad językami migowymi znajdują się dopiero w fazie początkowej. Ogólnie akceptowany model fonologiczny na razie nie został opracowany. Struktura języka migowego nie jest szeroko zbadana a znaczenie niemanualnych środków wyrazu nie zostało dokładnie wyjaśnione.

Dokładna definicja poprawnego wykonania pojedynczych słów nie istnieje. Dodatkowe utrudnienie stanowi brak języka pisanego dla języka migowego. Słowa nie mogą więc być ujednolicone w takim zakresie, jak w odniesieniu do języka

mówionego.

Brakuje wystarczającego i ustandaryzowanego materiału badawczego. Niezbędne sekwencje gestów do trenowania i testowania układów automatycznego rozpoznawania muszą zostać przygotowane.

Duży wpływ na jakość przetwarzanych obrazów a tym samym także na skuteczność rozpoznawania mają zmieniające się warunki oświetlenia.

Rozpoznawanie gestów w układzie wizyjnym wymaga przetwarzania dużej ilości danych o charakterze wizyjnym i wykonywania złożonych obliczeniowo algorytmów.

2.5 Podsumowanie

W rozdziale dokonano charakterystyki Polskiego Języka Miganego, zwracając szczególną uwagę na te jego własności, które są istotne z punktu widzenia rozpoznawania w układzie wizyjnym. Opisano tzw. wiązki cech dystynktywnych, a więc grupy cech, które jednoznacznie identyfikują dany gest i mogą być pomocne przy wyborze wektorów cech do rozpoznawania. Przedstawiono, oparty na cechach dystynktywnych, gestograficzny zapis znaków migowych. W końcowej części rozdziału zasygnalizowano problemy istotne w przypadku rozpoznawania PJM w układzie wizyjnym.

Rozdział 3

Problemy przetwarzania obrazu

Pierwszym etapem w rozpoznawaniu wyrazów i zdań PJM w układzie wizyjnym jest przetwarzanie obrazów w celu wyznaczenia wektorów cech. W rozdziale opisano wiążące się z tym problemy dotyczące rozpoznawania koloru skóry, wyznaczania mapy głębi i identyfikacji dłoni i twarzy. Omówiono wyniki przeprowadzonych eksperymentów oraz przedstawiono warianty wektorów cech, które uwzględniane będą podczas klasyfikacji.

3.1 Rozpoznawanie koloru skóry

Duży wpływ na jakość otrzymywanych obrazów binarnych, a tym samym także na skuteczność rozpoznawania mają zmieniające się warunki oświetlenia. Pociąga to za sobą konieczność stosowania metod przetwarzania obrazów, które są mniej zależne od natężenia docierającego światła. Progowanie obrazów monochromatycznych nie daje niezawodnych rezultatów. W przypadku niejednorodnego tła i zmieniających się warunków oświetlenia nie można ustalić a priori, czy dłonie i twarz mają być jaśniejsze, czy ciemniejsze od innych obiektów. Inne rozwiązanie, polegające na wykorzystaniu informacji o gradiencie, wiąże się z koniecznością rozstrzygania, które z wyodrębnionych krawędzi należą do dłoni i twarzy. Nie jest to zadanie łatwe, ponieważ krawędzie wyglądają zupełnie inaczej, nawet dla tego samego obiektu, obserwowanego pod różnymi kątami. Dlatego w niniejszej pracy zdecydowano się na wykorzystanie obrazów kolorowych, w których poszukuje się obszarów o chrominancji zbliżonej do skóry ludzkiej [32]. Metoda umożliwia poprawną identyfikację pod warunkiem, że ubranie osoby wykonującej gesty i inne obiekty pojawiające się w tle mają barwy odmienne od koloru skóry ludzkiej. Spełnienie tych wymagań w warunkach praktycznych jest znacznie prostsze aniżeli zapewnienie, aby dłonie i twarz osoby były najjaśniejszymi obiektami w obrazie. Przyjęcie takich założeń wstępnych jest dodatkowo uzasadnione tym, że osoby uczestniczące w nagrywaniu tłumaczeń w języku miganym także są proszone o ubieranie ciemniejszych ubrań z długim rękawem i występują najczęściej na jednorodnym tle. W przypadku, gdy w tle mogą się pojawić obiekty o barwach zbliżonych do skóry ludzkiej, np. twarze i dłonie innych osób, można zaproponować metodę będącą połączeniem przetwarzania obrazów kolorowych i uzyskanych w układzie stereowizyjnym map głębi (podrozdział

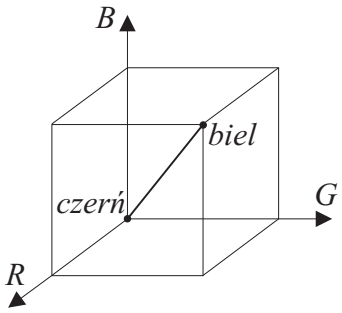
3.3). Mapy głębi wykorzystywane byłyby do wybierania tylko tych obiektów o kolorze skóry ludzkiej, które znajdują się najbliżej układu kamer.

Skóra ludzka ma charakterystyczną budowę, co wpływa na spektrum odbitego od niej światła. Należy ona do powierzchni dielektrycznych, dla których zjawisko odbicia światła może zostać zamodelowane za pomocą tzw. dwubarwnego modelu odbicia (*dichromatic reflection model*) [58]. Zgodnie z tą teorią światło odbite docierające do obserwatora (kamery) jest sumą światła odbitego od powierzchni obiektu (*surface reflection*) i światła, które wniknęło w głąb materiału, uległo tam wielokrotnemu rozszczepieniu i w części wydostało się z powrotem na zewnątrz obiektu (*body reflection*). Ponieważ skóra jest materiałem o dużej zawartości wody, można w stosunku do niej zastosować założenie, że odbicie od powierzchni ma rozkład spektralny w przybliżeniu odpowiadający rozkładowi źródła oświetlenia. Rzeczywisty kolor obiektu zależy wtedy od odbicia od wnętrza materiału. Odbicie powierzchniowe zachodzi w cienkiej warstwie naskórka i stanowi zaledwie 5% docierającego do kamery światła. Pozostałe 95% to odbicie od wnętrza materiału zachodzące już w warstwie skóry właściwej. Jest to dodatkowy argument przemawiający za tym, aby wykorzystać kolor właśnie w odniesieniu do skóry, ponieważ zawiera on głównie informacje o rzeczywistym kolorze obiektu.

Jakość segmentacji dłoni i twarzy zależy od wykorzystanej przestrzeni barw. Z analizy dostępnych w literaturze testów wynika, że nie ma przestrzeni barw uniwersalnej, dającej najlepsze rezultaty niezależnie od warunków oświetlenia, zastosowanej metody segmentacji i typu kamery. W pewnym stopniu uniezależnienie się od zmian warunków oświetlenia uzyskuje się w tych przestrzeniach, w których da się odseparować chrominancję od zależnej od poziomu oświetlenia luminancji. Dodatkową zaletą takiego podejścia jest redukcja wymiarowości uzyskiwana w wyniku liniowej bądź nieliniowej transformacji z trójwymiarowej przestrzeni RGB do dwuwymiarowej przestrzeni chrominancji. Wybór przestrzeni barw jest zadaniem ważnym, ponieważ kształt rozkładu barwy skóry ludzkiej zależy od przestrzeni chrominancji. Dlatego w niniejszej pracy przeprowadzono testy dla różnych przestrzeni barw, które wykorzystywane były w metodach opisywanych w literaturze i mają pewne własności istotne z punktu widzenia segmentacji. Do eksperymentów wybrano przestrzenie:

- znormalizowaną RGB,
- YUV,
- YIQ,
- barw przeciwstawnych OCS,
- barw przeciwstawnych w wersji logarytmicznej OCSL,
- I1I2I3,
- IHS,
- Lab.

W przestrzeni barw RGB kolor reprezentowany jest za pomocą trzech składowych: czerwonej R, zielonej G i niebieskiej B. W układzie współrzędnych prostokątnych, w którym na poszczególnych osiach odłożono wartości składowych R, G i B zbiór wszystkich możliwych barw tworzy sześcian. Na przekątnej tego sześcianu znajdują się odcienie szarości (rys. 3.1). Kolor punktu składa się z informacji ilościowej (lu-



Rys. 3.1. *Przestrzeń bar RGB*

minancji) określonej za pomocą sumy poszczególnych składowych $L = R + G + B$ i informacji jakościowej (chrominancji) zdefiniowanej za pomocą stosunku składowych podstawowych $R : G : B$. W celu zmniejszenia zależności od natężenia oświetlenia dokonuje się zalecanej przez Międzynarodową Komisję Oświetlenia CIE (*Commission International d'Eclairage*) normalizacji polegającej na podzieleniu każdej składowej przez wartość luminancji: $r = \frac{R}{L}$, $g = \frac{G}{L}$, $b = \frac{B}{L}$ [5]. Ponieważ otrzymane w ten sposób współrzędne trójchromatyczne spełniają równanie $r + g + b = 1$, do dalszego przetwarzania wybrano dwie z nich r i g. Otrzymane w wyniku normalizacji wartości współrzędnych trójchromatycznych r i g mogą być zakłócone przez szum w przypadku, gdy luminancja piksela jest niewielka. Dlatego zaprogramowano też drugi wariant segmentacji z wykorzystaniem znormalizowanej przestrzeni RGB, w którym piksele o jasnościach zbliżonych do zera nie zostały uwzględnione w procesie segmentacji.

Testy wykonano także dla dwóch przestrzeni wykorzystywanych w telewizji: YUV i YIQ. Przestrzeń YUV wykorzystywana jest w standardzie PAL. Kolor reprezentowany jest za pomocą jednego kanału achromatycznego Y i dwóch kanałów chromatycznych U, V. Wartości Y, U, V można otrzymać w wyniku liniowego przekształcenia wartości R, G i B [58].

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

W celu uniezależnienia się od zmieniającego się poziomu oświetlenia segmentację przeprowadzono z wykorzystaniem tylko składowych chromatycznych U i V.

Przestrzeń YIQ wykorzystywana jest w standardzie kodowania NTSC. Kolor punktu reprezentowany jest za pomocą jednego kanału achromatycznego Y i dwóch kanałów chromatycznych I i Q. Przekształcenie z przestrzeni RGB do przestrzeni

YIQ jest operacją liniową.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.273 & -0.322 \\ 0.212 & -0.522 & 0.315 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

Podobnie jak w poprzednim przypadku, do segmentacji wykorzystano składowe chromatyczne I i Q. Ponieważ niektórzy autorzy zwracali uwagę na to, że składowa I przestrzeni YIQ jest wrażliwa na kolor skóry, zaprogramowano też wersję algorytmu segmentacji dla przypadku jednowymiarowego z wykorzystaniem jedynie składowej I [12].

Przestrzeń barw przeciwstawnych zainspirowana została badaniami fizjologicznymi ludzkiego systemu wzrokowego. Obserwowane u człowieka zjawiska kontrastu barwy następczej pozwoliły na wysunięcie hipotezy, że kolor czerwony jest kolorem "przeciwnym" do koloru zielonego i analogicznie kolor niebieski jest "przeciwny" do koloru żółtego [52]. W przestrzeni tej kolor punktu określony jest za pomocą jednego kanału achromatycznego $WhBl$ (czarny-biały) i dwóch kanałów chromatycznych: RG (czerwony-zielony) i YeB (żółty-niebieski). Przekształcenie z przestrzeni RGB jest transformacją liniową.

$$\begin{bmatrix} WhBl \\ RG \\ YeB \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.3}$$

Do segmentacji wykorzystano kanały chromatyczne RG i YeB. W systemie wzrokowym człowieka sygnał przy wyjściu z receptorów na nerw optyczny poddawany jest przekształceniu, którego charakterystyka jest w przybliżeniu logarytmiczna. Dlatego do testów wykorzystano także logarytmiczną wersję przestrzeni barw przeciwstawnych.

$$WhBl_{log} = log\,(G) \tag{3.4}$$

$$RG_{log} = log\,(R) - log\,(G) \tag{3.5}$$

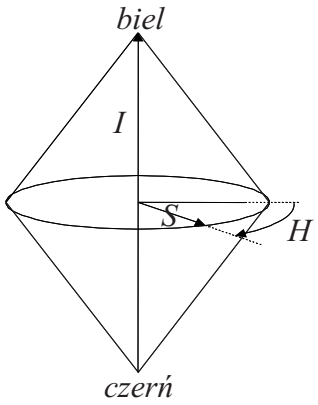
$$YeB_{log} = log\,(B) - \frac{log\,(R) + log\,(G)}{2} \tag{3.6}$$

W procesie segmentacji także uwzględniono tylko składowe chromatyczne RG_{log} i YeB_{log}.

Przestrzeń I1I2I3 wprowadzona została jako wynik badań statystycznych na dużym zbiorze obrazów. W przestrzeni tej kolor punktu opisany jest za pomocą składowej achromatycznej $I1$ i dwóch składowych chromatycznych $I2$ i $I3$. Przekształcenie z przestrzeni RGB jest transformacją liniową.

$$\begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.7}$$

Do segmentacji koloru skóry wykorzystano składowe chromatyczne $I2$ i $I3$.



Rys. 3.2. *Przestrzeń bar IHS*

Przestrzeń barw IHS wykorzystywana jest w komputerowych programach do edycji barw. Kolor punktu reprezentowany jest tutaj za pomocą intensywności I, która jest składową achromatyczną i dwóch składowych chromatycznych: barwy H i nasycenia S (rys. 3.2) [72]. Odpowiada to intuicyjnemu postrzeganiu koloru przez człowieka. Transformacja z przestrzeni RGB do przestrzeni IHS ma charakter nieliniowy.

$$I = \frac{1}{3}(R + G + B) \tag{3.8}$$

$$S = 1 - \frac{3}{R + G + B}\left[min\left(R, G, B\right)\right] \tag{3.9}$$

$$H = cos^{-1}\left[\frac{0.5\left[(R-G) + (R-B)\right]}{\sqrt{(R-G)^2 + (R-B)(G-B)}}\right], dlaB > G : H = 2\pi - H \tag{3.10}$$

Do segmentacji wykorzystano składowe chromatyczne: barwę i nasycenie.

W 1976 roku CIE wprowadziła przestrzeń barw Lab. W przestrzeni tej kolor reprezentowany jest za pomocą składowej achromatycznej L i dwóch składowych chromatycznych a^* i b^*. Cechą charakterystyczną tej przestrzeni jest to, że różnicom kolorów postrzeganym przez oko ludzkie jako jednakowe odpowiadają takie same odległości euklidesowe [14]. Uznając, że właściwość ta może mieć znaczenie w procesie segmentacji wykonano testy także dla tej przestrzeni wykorzystując składowe chromatyczne a^* i b^*. Przekształcenie z przestrzeni RGB jest transformacją nieliniową.

$$L = 116 * f_y - 16 \tag{3.11}$$

$$a^* = 500\left(f_x - f_y\right) \tag{3.12}$$

$$b^* = 200\left(f_y - f_z\right) \tag{3.13}$$

gdzie:

$$f_x = \begin{cases} X^{\frac{1}{3}} & X > 0.00865 \\ 7.787X + \frac{16}{116} & X \leq 0.00865 \end{cases} \tag{3.14}$$

$$f_y = \begin{cases} Y^{\frac{1}{3}} & Y > 0.00865 \\ 7.787Y + \frac{16}{116} & Y \leq 0.00865 \end{cases} \tag{3.15}$$

$$f_z = \begin{cases} Z^{\frac{1}{3}} & Z > 0.00865 \\ 7.787Z + \frac{16}{116} & Z \leq 0.00865 \end{cases} \tag{3.16}$$

zaś X, Y i Z są współrzędnymi opisującymi barwę w przestrzeni CIE XYZ i mogą być wyznaczone następująco:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.17}$$

W testowanych metodach przyjęto, że na początku użytkownik systemu "przedstawia się" prezentując do kamery wewnętrzną część dłoni z wyprostowanymi palcami skierowanymi ku górze (konfiguracja odpowiadająca znakowi daktylograficznemu dla litery B, rys. 2.1). Z dłoni tej wycinany jest ręcznie prostokątny obszar zawierający tylko piksele należące do skóry. Obszar ten wykorzystywany jest do budowania modelu rozkładu chrominancji skóry ludzkiej. W eksperymentach wykorzystano:

- metodę opartą o histogram kolorów z aproksymacją histogramu za pomocą rozkładu normalnego (G),
- metodę opartą o histogram kolorów z wygładzaniem histogramu filtrem Gaussa (H),
- metodę największej wiarygodności (ML, *Maximum Likelihood*),
- metodę maksimum prawdopodobieństwa a posteriori (MAP, *Maximum A Posteriori Probability*).

W metodach opartych o histogram kolorów, po transformacji wyciętego fragmentu obrazu do żądanej przestrzeni barw, wybierane są dwie składowe opisujące chrominancję i generowana jest dwuwymiarowa tablica histogramu. Tablica ta zawiera informację o rozkładzie cechy "chrominancja" w klasie "skóra ludzka". Histogram może nie być dostatecznie reprezentatywny, ponieważ wyznaczany jest na podstawie niewielkiego fragmentu obrazu. Pewne wartości chrominancji niewystępujące we wzorcu mogą pojawić się w innych obrazach zawierających dłoń i twarz. Dlatego zaprogramowano dwa warianty metody. Pierwszy z nich polegał na aproksymacji histogramu za pomocą rozkładu normalnego:

$$G(\xi) = \frac{1}{2\pi\sqrt{|C|}} e^{-\frac{1}{2}(\xi-\mu)^T C^{-1}(\xi-\mu)} \tag{3.18}$$

gdzie: $\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$, $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, $C = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$ oznaczają odpowiednio dwuelementowy wektor chrominancji, wartość oczekiwaną i kowariancję. Drugi wariant polegał na wygładzeniu histogramu filtrem Gaussa o masce 3×3.

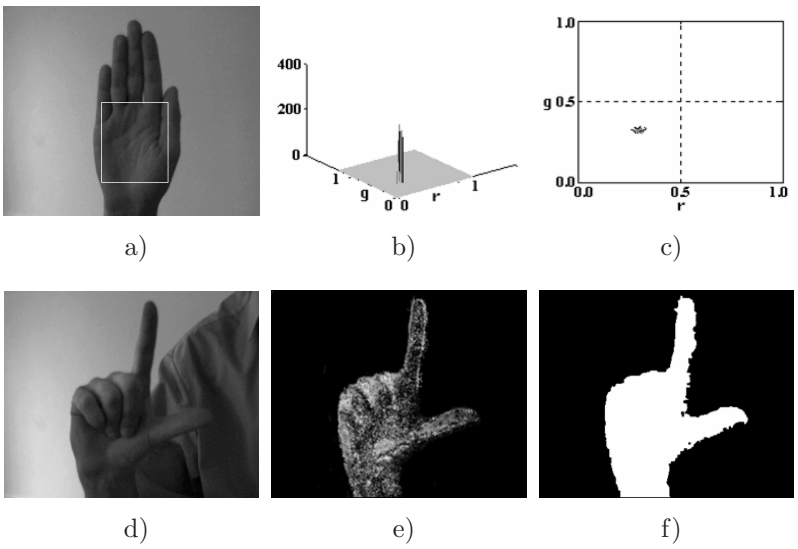
Po przygotowaniu modelu, wejściowy obraz kolorowy transformowany jest do żądanej przestrzeni barw i przekształcany w obraz prawdopodobieństwa wg reguły

(3.19) dla wariantu z aproksymacją histogramu i reguły (3.20) dla wariantu z wygładzaniem filtrem Gaussa.

$$z(i,j) = G(\xi) \tag{3.19}$$

$$z(i,j) = h(\xi) \tag{3.20}$$

gdzie: i, j - oznaczają współrzędne piksela, h - dwuwymiarowy wygładzony histogram, z - wynikowy obraz prawdopodobieństwa. Po przeskalowaniu wartości do zakresu 0 - 255 otrzymuje się obraz z poziomami szarości, w którym im piksel jaśniejszy, tym większe prawdopodobieństwo, że należy on do skóry ludzkiej. Obraz ten poddany został filtracji dolnoprzepustowej za pomocą maski Gaussa o wymiarach 3×3. Następnie wyznaczono jego histogram i ustalono adaptacyjnie wartość progu binaryzacji z wykorzystaniem metody opartej na aproksymacji histogramu za pomocą krzywych Gaussa [9, 45, 53]. Powstały po progowaniu obraz poddano morfologicznej filtracji OC polegającej na wykonaniu kolejno otwarcia i zamknięcia [51]. W ten sposób usunięto drobne dziury i wygładzano brzegi na obrazie dłoni i twarzy. Schemat przetwarzania dla metod opartych o histogram kolorów przedstawiono na rys. 3.3.



Rys. 3.3. *Segmentacja dłoni metodą opartą o histogram kolorów: a) fragment obrazu wykorzystywany do generowania modelu (przekształcony do poziomów szarości), b), c) model chrominancji skóry, d) obraz wejściowy (przekształcony do poziomów szarości), e) obraz prawdopodobieństwa f) obraz e) po binaryzacji i morfologicznej filtracji OC.*

W metodach opartych o twierdzenie Bayesa [16, 21, 43] zakłada się, że piksele obrazu mogą należeć do dwóch klas: klasa skóra - $\mathcal{S}$ i klasa tło - $\mathcal{T}$. Jako wyróżnik klas wykorzystuje się chrominancję ξ, jako kryterium klasyfikacji wynik porównania prawdopodobieństw warunkowych $P(\mathcal{S}|\xi)$ i $P(\mathcal{T}|\xi)$. Jeżeli

$$P(\mathcal{S}|\xi) > P(\mathcal{T}|\xi) \tag{3.21}$$

to wektor ξ jest klasyfikowany jako należący do skóry. Jeżeli zaś

$$P(\mathcal{S}|\xi) < P(\mathcal{T}|\xi) \tag{3.22}$$

wektor ξ jest klasyfikowany jako należący do tła. Zgodnie z regułą Bayesa [21] mamy:

$$P(\mathcal{S}|\xi) = \frac{p(\xi|\mathcal{S})\,P(\mathcal{S})}{p(\xi)}, P(\mathcal{T}|\xi) = \frac{p(\xi|\mathcal{T})\,P(\mathcal{T})}{p(\xi)} \tag{3.23}$$

gdzie $p(\xi)$ jest funkcją gęstości rozkładu wektora ξ, a $p(\xi|\mathcal{S})$ i $p(\xi|\mathcal{T})$, są funkcjami gęstości rozkładu prawdopodobieństw warunkowych. Można więc zapisać warunki równoważne z (3.21) i (3.22):

$$p(\xi|\mathcal{S})\,P(\mathcal{S}) > p(\xi|\mathcal{T})\,P(\mathcal{T}) \tag{3.24}$$

i

$$p(\xi|\mathcal{S})\,P(\mathcal{S}) < p(\xi|\mathcal{T})\,P(\mathcal{T}) \tag{3.25}$$

Rozkłady cechy chrominancja ξ w klasie skóra $\mathcal{S}$: $p(\xi|\mathcal{S})$ i w klasie $\mathcal{T}$: $p(\xi|\mathcal{T})$ przybliżono za pomocą znormalizowanych histogramów wygenerowanych dla obrazu wzorcowego, odpowiednio dla pikseli należących do skóry i dla pikseli należących do tła. Możliwe są dwa założenia dotyczące prawdopodobieństw wystąpienia klasy skóra $P(\mathcal{S})$ i klasy tło $P(\mathcal{T})$. Przyjęcie, że $P(\mathcal{S}) = P(\mathcal{T})$ prowadzi do metody ML. Inny sposób polega na wyznaczeniu wartości prawdopodobieństw $P(\mathcal{S})$ i $P(\mathcal{T})$ na podstawie analizy liczb pikseli należących do skóry i do tła w obrazie wzorcowym. Prowadzi to do metody MAP. W wyniku zastosowania metody ML i MAP otrzymujemy od razu obrazy binarne. Obrazy te podobnie jak w przypadku metod opartych na histogramie kolorów poddane zostały morfologicznej filtracji OC.

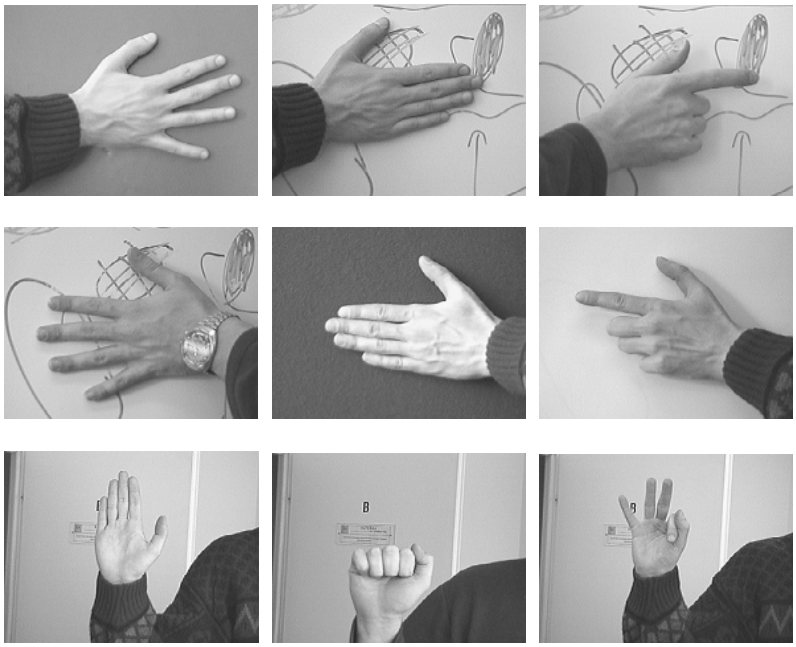
Analogiczne metody zastosowano także do przypadku wykorzystania tylko składowej I przestrzeni barw YIQ. Histogram był wtedy tablicą jednowymiarową i aproksymowano go za pomocą jednowymiarowej krzywej Gaussa.

Eksperymenty wykonano w pomieszczeniu zamkniętym. Przygotowano bazę danych dla sześciu różnych warunków: 1) dla oświetlenia dziennego w dzień pochmurny i jednorodnego tła, 2) dla oświetlenia dziennego w dzień słoneczny i jednorodnego tła, 3) dla oświetlenia sztucznego otrzymanego z neonówek i jednorodnego tła, 4) oświetlenia dziennego w dzień pochmurny i niejednorodnego tła, 5) oświetlenia dziennego w dzień słoneczny i niejednorodnego tła i 6) oświetlenia sztucznego z neonówek i niejednorodnego tła. Przez tło jednorodne rozumie się tutaj tło o jednolitej barwie. Wykorzystano w tym celu: białą ścianę laboratorium, białą tablicę szkolną, niebieską tablicę szkolną i zielona tablicę ogłoszeniową. Jako tło niejednorodne wykorzystano białą tablicę szkolną zapisaną za pomocą różnokolorowych markerów oraz typową scenerię w laboratorium. W eksperymencie uczestniczyły dwie osoby o różnej karnacji skóry. Przykładowe obrazy wykorzystane w eksperymentach przedstawiono na rysunku 3.4.

Przeprowadzono 8 testów (tab. 3.1). Modele rozkładu chrominancji generowano oddzielnie dla każdych warunków oświetlenia z wyjątkiem testu 7 i 8, gdzie obowiązywał jeden model rozkładu.

Dwa ważne kryteria dla detektora pikseli skóry to:

- dokładność z jaką dany model chrominancji opisuje złożony rzeczywisty rozkład w danej przestrzeni,



Rys. 3.4. *Przykładowe obrazy wykorzystane w testach różnych przestrzeni barw (przekształcone do poziomów szarości).*

Tab. 3.1. *Wykaz przeprowadzonych testów różnych przestrzeni barw*

Nr testu	Oświetlenie	Tło	Liczba obrazów testowych
1	Dzienne (pochmurno)	Jednorodne	36
2	Dzienne (słonecznie)	Jednorodne	36
3	Sztuczne (świetlówki)	Jednorodne	36
4	Dzienne (pochmurno)	Niejednorodne	18
5	Dzienne (słonecznie)	Niejednorodne	18
6	Sztuczne (świetlówki)	Niejednorodne	18
7	Wszystkie obrazy z testów 1 - 6 model wzorcowy skóry jak dla testu 2		162
8	Wszystkie obrazy z testów 1 - 6 model skóry otrzymany na podstawie danych wzorcowych z testów 1, 2, 3		162

- wielkość obszaru nakładania się rozkładów barw dla skóry i tła.

Wygenerowane obrazy binarne porównywane były z obrazami wzorcowymi otrzymanymi w wyniku ręcznej segmentacji obrazów testowych. Obszary błędów zostały naniesione na obrazy wejściowe, dzięki czemu możliwe było także wyciągnięcie wniosków na temat selektywności segmentacji i niewrażliwości na cienie i odblaski. W celu oceny jakości segmentacji wyznaczono następujące wskaźniki: SE - skin error, liczba pikseli należących do skóry, które zostały mylnie zaliczone do tła, NSE - non-skin error, liczba pikseli należących do tła, które zostały mylnie zaliczone do skóry,

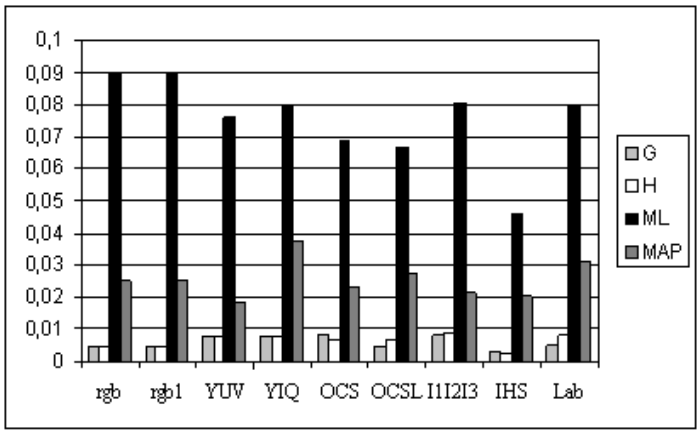
E - liczba pikseli sklasyfikowanych niepoprawnie. Wymienione wskaźniki rozpatrywano w odniesieniu do łącznej liczby pikseli. Na rys. 3.5, 3.6, 3.7 przedstawiono wyniki dotyczące kolejno względnych wartości wskaźników SE, NSE i E dla testu 8. Omawiając wyniki wykorzystano następujące oznaczenia (w nawiasach podano współrzędne chromatyczne zastosowane do modelowania koloru skóry):

rgb - znormalizowana przestrzeń RGB; (r,g),
rgb1 - znormalizowana przestrzeń RGB z pominięciem pikseli o małych jasnościach; (r,g),
YUV - przestrzeń YUV; (U,V),
YIQ - przestrzeń YIQ; (I,Q),
YIQ1 - przestrzeń YIQ z wykorzystaniem tylko składowej I,
OCS - przestrzeń barw przeciwstawnych; (R-G , Ye-B),
OCSL - logarytmiczna wersja przestrzeni barw przeciwstawnych; (log R-log G, log B-(log R+log G)/2),
I1I2I3 - przestrzeń I1I2I3; (I2,I3),
IHS - przestrzeń IHS; (H, S),
Lab - przestrzeń CIELab; (a*, b*),
G - metoda z aproksymacją histogramu za pomocą rozkładu normalnego,
H - metoda z wygładzaniem histogramu,
ML - metoda największej wiarogodności,
MAP - metoda maksimum prawdopodobieństwa a posteriori.

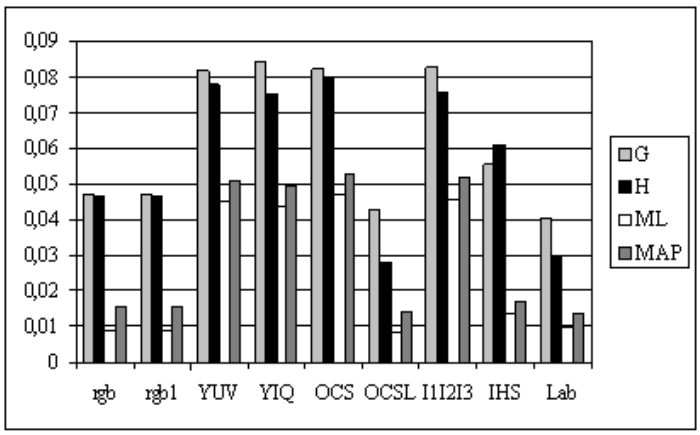
Dla przestrzeni YIQ z wykorzystaniem tylko składowej I otrzymano dużo gorsze wyniki, dlatego pominięto ją na rys. 3.5 - 3.7.

Podsumowanie wyników wszystkich eksperymentów zawiera tab. 3.2. Syntetyczne omówienie poszczególnych testów zamieszczono w [35]. Tabelę sporządzono wybierając kryteria $E > 5\%$, NSE, $SE > 2.5\%$. Stwierdzono, że rezultat $E > 2.5\%$ dotyczy zdecydowanej większości przypadków. Analizując białe pola można wskazać przestrzenie barw i metody detekcji, które okazały się najlepsze. Są to przestrzenie rgb, rgb1, IHS z metodami detekcji skóry MAP, rgb, rgb1 dla metody ML, OCSL dla metod H, ML, MAP oraz I1I2I3 z metodą G i Lab z metodą MAP. W tabeli zauważa się przewagę błędów typu SE tzn. tendencję do klasyfikowania fragmentów skóry jako tła. Błędy te objawiają się w formie dziur w obszarze dłoni lub jako efekt erozji morfologicznej. Błędy typu NSE z kolei objawiają się jako niewielkie obiekty w obszarze tła oraz podobnie jak efekt dylatacji morfologicznej powodują np. sklejanie się palców.

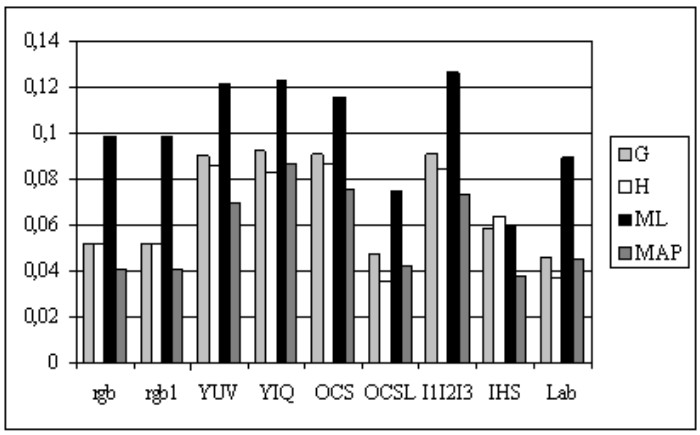
Testy wykonano na komputerze PC z procesorem AMD Athlon 1,81GHz i 1 GB pamięci RAM pracującym pod kontrolą systemu operacyjnego Windows XP. Oprogramowanie napisano w C++ z wykorzystaniem platformy Visual C++ .NET. Czasy przetwarzania otrzymane dla obrazów o wymiarach 320×240 zamieszczono w tab. 3.3. Na podstawie przeprowadzonych testów można wyróżnić znormalizowaną przestrzeń RGB, logarytmiczną przestrzeń barw przeciwstawnych OCSL, przestrzenie IHS, Lab i I1I2I3, dla których otrzymywano najmniejsze błędy. Wśród metod segmentacji nieco lepsze okazały się metody bayesowskie. Metody te jednak wyma-



Rys. 3.5. *Względna wartość wskaźnika NSE dla testu 8*



Rys. 3.6. *Względna wartość wskaźnika SE dla testu 8*



Rys. 3.7. *Względna wartość wskaźnika E dla testu 8*

Tab. 3.2. *Podsumowanie wyników eksperymentów dla różnych przestrzeni barw*

	RGB				RGB1				YUV				YIQ				YIQ1		OCS				OCSL				I1I2I3				IHS				Lab			
Test	G	H	L	A	G	H	L	A	G	H	L	A	G	H	L	A	G	H	G	H	L	A	G	H	L	A	G	H	L	A	G	H	L	A	G	H	L	A
1	n				n				2	s	s	s	s	s	2	s	s	s	n	s	s	s	n					s	2	s	s				s				
2		s				s			s	s	s	s	s	s	s	s	s	s		s	s	s						s	s	s	s	s			s				
3		s				s				s	s	s	s	s	s	s	s	s	s	s	s	s	n					s	s	s	s	s			s				
4	n				n				2	s	2	s	s	s	2	s	2	2	n	s	2	s	n					s	2	s	s		n		s	n	n		
5		s				s			s	s	s	s	s	s	s	s	n		n	s	s	s	n					s	s	s	s				s				
6	n	s			n	s			n				s				n	n	n	s			2					s			s				s	n			
7	s	s	2	2	s	s	2	2	s	s	2	2	s	s	2	2	2	2	s	s	2	2	s	s	n	n	s	s	2	2	s	s	2	2	s	s	2	2
8	s	s	n		s	s	n		s	s	2	s	s	s	2	2	n	2	s	s	2	2	s	s	n	n	s	s	2	2	s	s	n		s	s	n	n

Kolor szary dotyczy wyników, w których wartość wskaźnika E przekroczyła 0.05, n - NSE $>$ 0.025, s - SE $>$ 0.025, 2 = n & s, L = ML, A = MAP.

Tab. 3.3. *Czasy przetwarzania w ms*

-	rgb	rgb1	YUV	YIQ	YIQ1	OCS	OCSL	I1I2I3	IHS	Lab
G	10	11	16	16	21	16	21	33	11	11
H	10	10	15	14	23	15	21	33	11	11
ML	12	14	14	14	-	14	20	31	10	10
MAP	12	14	14	14	-	14	20	31	10	10

gają znajomości rozkładu chrominancji dla tła w obrazie wzorcowym i zakładają, że rozkład ten nie ulega później zmianom. W przypadku gestów dynamicznych, poruszają się nie tylko dłonie ale także przedramiona i ramiona co wpływa na zmiany rozkładu chrominancji dla tła. Dlatego do dalszych eksperymentów z rozpoznawaniem wyrazów i zdań PJM zdecydowano się wybrać metody oparte o aproksymację histogramu za pomocą rozkładu Gaussa (G). Analizując otrzymywane błędy i biorąc pod uwagę czasy wykonania do dalszych eksperymentów wybrano znormalizowaną przestrzeń RGB.

Podobne eksperymenty przeprowadzono także dla przypadku detekcji twarzy w obrazie kolorowym. Wykorzystano bazę 108 obrazów twarzy, zarejestrowanych przez dwie osoby w różnorodnych warunkach oświetleniowych oraz przy różnych tłach. Otrzymano zbliżone wyniki.

3.2 Segmentacja dłoni i twarzy

W wyniku zastosowania metod opisanych w podrozdziale 3.1 otrzymujemy obrazy binarne zawierające od jednego do trzech obiektów o dominujących rozmiarach, odpowiadających twarzy i dłoniom osoby wykonującej gest. Trzy obiekty występują wtedy, gdy dłonie nie dotykają się, nie przysłaniają się wzajemnie, nie dotykają twarzy i nie pojawiają się na tle twarzy. Jeżeli obie dłonie dotykają się lub przysłaniają się wzajemnie, albo jedna z dłoni dotyka twarzy, bądź pojawia się na tle twarzy, to otrzymujemy dwa obiekty. Jeden obiekt występuje wtedy, gdy obie dłonie równocześnie dotykają twarzy lub pojawiają się na tle twarzy. W celu rozróżnienia,

który z obiektów odpowiada dłoni prawej, dłoni lewej i twarzy, zastosowano następujący heurystyczny algorytm, wykorzystujący informację o pozycjach i polach powierzchni obiektów w bieżącej i poprzedniej ramce:

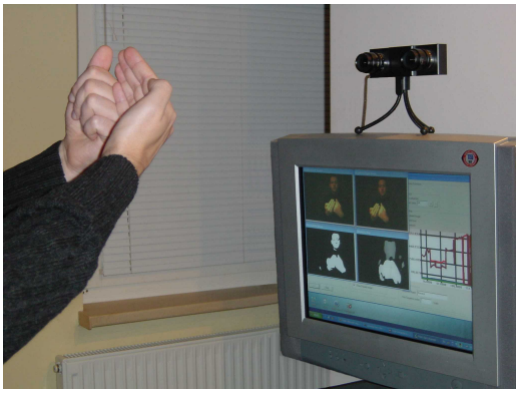
Algorytm 3.1: Rozróżnienie dłoni prawej, lewej i twarzy w obrazie binarnym

1. Wykonaj segmentację i wyznacz pola powierzchni i środki ciężkości otrzymanych obiektów.
2. Odrzuć obiekty o polach powierzchni < MIN_HAND_SURFACE (usunięcie zakłóceń nieodfiltrowanych przez filtrację OC).
3. Jeżeli liczba pozostałych obiektów jest równa trzy, idź do 4. Jeżeli liczba pozostałych obiektów wynosi dwa, idź do 5. Jeżeli pozostał jeden obiekt, idź do 6. W innych przypadkach idź do 7.
4. Dłonie i twarz nie dotykają się. Podstaw: TWARZ := obiekt o największej powierzchni. Jeżeli jest to pierwsza klatka w sekwencji, idź do 4.1. W przeciwnym razie idź do 4.2.
   - 4.1 Podstaw: DŁOŃ_PRAWA := obiekt leżący bardziej na prawo z pozostałych dwóch obiektów, DŁOŃ_LEWA := obiekt leżący bardziej na lewo z pozostałych dwóch obiektów.
   - 4.2 Podstaw: DŁOŃ_PRAWA := ten z dwóch pozostałych obiektów, który leży bliżej pozycji dłoni prawej w poprzedniej ramce, DŁOŃ_LEWA := ten z dwóch pozostałych obiektów, który leży bliżej pozycji dłoni lewej w poprzedniej ramce.
5. Nastąpiło zetknięcie dłoni dominującej i twarzy bądź dłoni dominującej z dłonią niedominującą. Jeżeli powierzchnia większego obiektu > MAX_FACE_SURFACE, idź do 5.1; w przeciwnym razie idź do 5.2.
   - 5.1 Dłoń dominująca dotyka twarzy. Podstaw: TWARZ := obiekt odpowiadający twarzy z poprzedniej klatki, DŁOŃ_PRAWA := obiekt odpowiadający dłoni prawej z poprzedniej klatki, DŁOŃ_LEWA := obiekt o mniejszej powierzchni.
   - 5.2 Dłoń dominująca i niedominująca dotykają się. Podstaw: TWARZ := obiekt o większej powierzchni, DŁOŃ_PRAWA := obiekt odpowiadający dłoni prawej z poprzedniej klatki, DŁOŃ_LEWA := obiekt odpowiadający dłoni prawej z poprzedniej klatki
6. Nastąpiło zetknięcie obu dłoni i twarzy. Podstaw: TWARZ := obiekt odpowiadający twarzy z poprzedniej klatki, DŁOŃ_PRAWA := obiekt odpowiadający dłoni prawej z poprzedniej klatki, DŁOŃ_LEWA := obiekt odpowiadający dłoni lewej z poprzedniej klatki.
7. Wykryto niewłaściwą liczbę obiektów. Zasygnalizuj błąd.

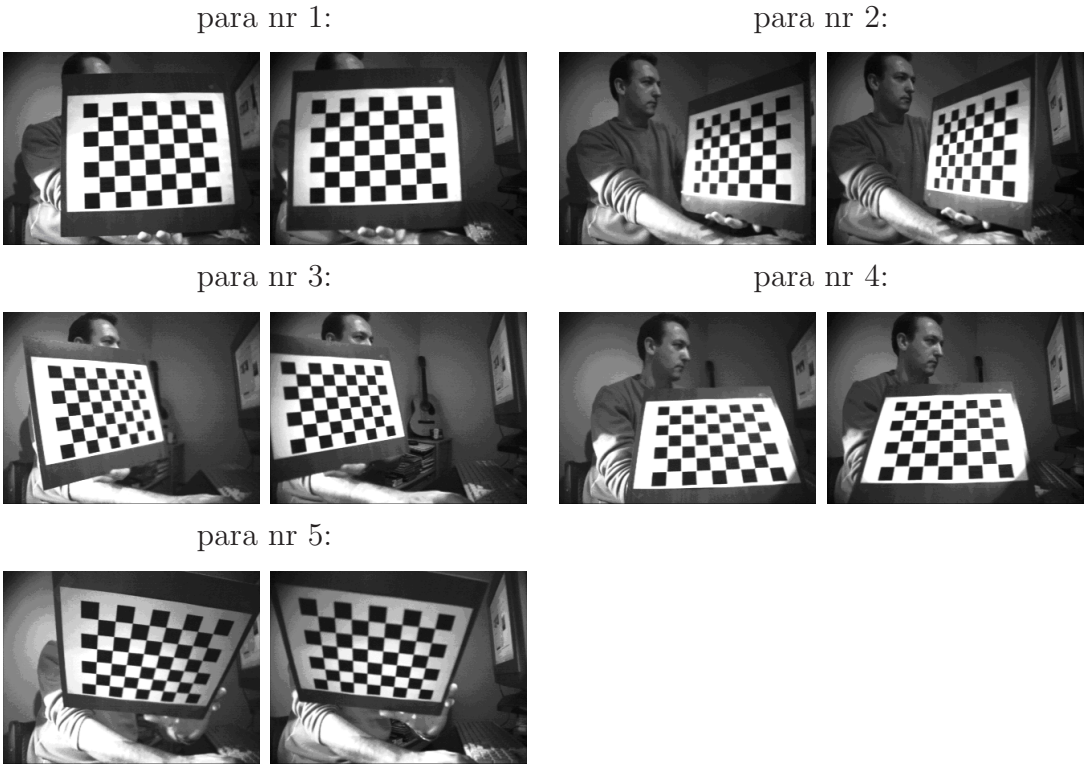
Algorytm wymaga ustalenia wartości dla dwóch zmiennych: MIN_HAND_SURFACE i MAX_FACE_SURFACE. Pierwsza oznacza minimalną powierzchnię binarnego obiektu odpowiadającego dłoni. Wartość tę ustalono eksperymentalnie, obserwując binarne obrazy odpowiadające dłoniom dla wszystkich rozważanych gestów. Przyjęto, że MIN_HAND_SURFACE będzie równe powierzchni najmniejszego obrazu dłoni, pomniejszonej jeszcze o 5%. Wszystkie obiekty o powierzchniach mniejszych od tej stałej traktowane są jako zakłócenia. MAX_FACE_SURFACE jest maksymalną powierzchnią binarnego obiektu odpowiadającego twarzy. Wartość ta została ustalona podczas "przedstawiania się" użytkownika w fazie generowania modelu (podrozdział 3.1). Po obliczeniu parametrów modelu wykorzystano go do wygenerowania binarnego obrazu twarzy użytkownika i przyjęto, że MAX_FACE_SURFACE będzie równa powierzchni otrzymanego obiektu binarnego powiększonej jeszcze o 5%, aby uwzględnić ewentualne wahania rozmiaru w kolejnych klatkach. Wszystkie obiekty o powierzchniach większych od MAX_FACE_SURFACE traktowane są jako zetknięcie twarzy i dłoni dominującej.
W przypadku gestów rozpoczynających się od zetknięcia lub skrzyżowania dłoni, algorytm rozpoczyna swe działanie jeszcze przed rozpoczęciem wykonywania właściwego gestu, gdy dłonie i twarz nie stykają się, dłoń prawa jest po prawej a lewa po lewej stronie ciała. Przeprowadzone eksperymenty wykazały, że dla rozważanego słownika gestów PJM, przy typowym tempie wykonywania i prędkości przetwarzania 25 ramek/s, opisywany algorytm poprawnie identyfikuje obiekty w obrazie binarnym.

3.3 Stereowizja

Statyczna konfiguracja dłoni i jej ruch mają charakter przestrzenny. W wyniku dokonywanej przez układ obrazujący projekcji do płaszczyzny obrazu spora część informacji 3D zostaje utracona. Pomimo, iż częściowa informacja o składowej ruchu zgodnej z osią optyczną kamery przekazywana jest w postaci zmieniających się rozmiarów dłoni, zdecydowano się na wzbogacenie wektora cech o informację 3D. Do wyznaczenia mapy dysparycji wykorzystywano monochromatyczne obrazy o wymiarach 320×240 pikseli, zarejestrowane z wykorzystaniem stereowizyjnego układu kamer firmy Videre Design (rys. 3.8).
Gwarantuje on, że obrazy z obu kamer są rejestrowane równocześnie, co ma duże znaczenie w przypadku szybko zmieniającej się sceny. Kalibrację układu kamer przeprowadzono prezentując przygotowaną specjalnie planszę w pięciu różnych pozycjach (rys. 3.9). Wykorzystano procedury z biblioteki Small Vision System, dostarczanej wraz z kamerą stereowizyjną [42]. Parametry wewnętrzne i zewnętrzne układu stereowizyjnego wyznaczane były za pomocą metody opartej o algorytm Tsai [48, 73]. Znajomość tych parametrów potrzebna jest do dokonania tzw. rektyfikacji obrazów stereo, czyli sprowadzenia ich do takiej formy, jakby otrzymane były w idealnym kanonicznym układzie stereowizyjnym, w którym osie optyczne kamer są równoległe a obrazy leżą na tej samej płaszczyźnie. Dla obrazów otrzymanych



Rys. 3.8. *Stanowisko laboratoryjne ze sterowizyjnym układem kamer.*



Rys. 3.9. *Ujęcia planszy kalibracyjnej (z lewej i prawej kamery) wykorzystywane w procesie kalibracji.*

w idealnym kanonicznym układzie stereowizyjnym odpowiadające sobie linie epipolarne są współliniowe, co znacznie upraszcza proces poszukiwania odpowiedników. Zrektyfikowane obrazy wykorzystano do wyznaczania map dysparycji, zawierających informację o głębi.

W tym celu przetestowano korelacyjną metodę generowania zwartej mapy dysparycji dla 10 miar dopasowania ujętych w tab. 3.4 [7, 11].
Przyjęto następujące oznaczenia:

Tab. 3.4. *Miary dopasowania uwzględnione w testach*

nazwa	formuła
SAD	$\sum_{(i,j)\in U} \left\lvert I_1\left(x+i,y+j\right) - I_2\left(x+d_x+i,y+d_y+j\right)\right\rvert$
ZSAD	$\sum_{(i,j)\in U} \left\lvert \left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right) - I_2\left(\left(x+d_x+i,y+d_y+j\right) - \overline{I_2\left(x+d_x,y+d_y\right)}\right)\right\rvert$
SSD	$\sum_{(i,j)\in U} \left(I_1\left(x+i,y+j\right) - I_2\left(x+d_x+i,y+d_y+j\right)\right)^2$
ZSSD	$\sum_{(i,j)\in U} \left[\left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right) - I_2\left(\left(x+d_x+i,y+d_y+j\right) - \overline{I_2\left(x+d_x,y+d_y\right)}\right)\right]^2$
SSD-N	$\frac{\sum_{(i,j)\in U} \left[I_1\left(x+i,y+j\right) - I_2\left(x+d_x+i,y+d_y+j\right)\right]^2}{\sqrt{\sum_{(i,j)\in U} I_1\left(x+i,y+j\right)^2 \cdot \sum_{(i,j)\in U} I_2\left(x+d_x+i,y+d_y+j\right)^2}}$
ZSSD-N	$\frac{\sum_{(i,j)\in U} \left[\left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right) - \left(I_2\left(x+d_x+i,y+d_y+j\right) - \overline{I_2\left(x+d_x,y+d_y\right)}\right)\right]^2}{\sqrt{\sum_{(i,j)\in U} \left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right)^2 \cdot \sum_{(i,j)\in U} \left(I_2\left(x+d_x+i,y+d_y+j\right) - \overline{I_2\left(x+d_x,y+d_y\right)}\right)^2}}$
SCP	$\sum_{(i,j)\in U} I_1\left(x+i,y+j\right) \cdot I_2\left(x+d_x+i,y+d_y+j\right)$
SCP-N	$\frac{\sum_{(i,j)\in U} I_1\left(x+i,y+j\right) \cdot I_2\left(x+d_x+i,y+d_y+j\right)}{\sqrt{\sum_{(i,j)\in U} I_1\left(x+i,y+j\right)^2 \cdot \sum_{(i,j)\in U} I_2\left(x+d_x+i,y+d_y+j\right)^2}}$
CoVar	$\frac{\sum_{(i,j)\in U} \left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right) \cdot \left(I_2\left(x+d_x+i,y+d_y+j\right) - \overline{I2\left(x+d_x,y+d_y\right)}\right)}{\sqrt{\sum_{(i,j)\in U} \left(I_1\left(x+i,y+j\right) - \overline{I_1\left(x,y\right)}\right)^2 \cdot \sum_{(i,j)\in U} \left(I_2\left(x+d_x+i,y+d_y+j\right) - \overline{I_2\left(x+d_x,y+d_y\right)}\right)^2}}$
Census	$\sum_{(i,j)\in U} IC_1\left(x+i,y+j\right) \otimes IC_2\left(x+d_x+i,y+d_y+j\right)$

- $I_1\left(x,y\right)$ i $I_2\left(x,y\right)$ - wartości intensywności piksela o współrzędnych (x,y), odpowiednio dla obrazu z kamery lewej i prawej,
- $\overline{I_1\left(x,y\right)}$ i $\overline{I_2\left(x,y\right)}$ - średnie wartości intensywności w pewnym otoczeniu punktu o współrzędnych (x,y), odpowiednio dla obrazu z kamery lewej i prawej,
- $IC_1\left(x,y\right)$ i $IC_2\left(x,y\right)$ - wartości miary Census wyznaczone w punkcie o współrzędnych (x,y), odpowiednio dla obrazu z kamery lewej i prawej,
- i, j - indeksy przyjmujące wartości ze zbioru liczb całkowitych,
- d_x, d_y - wartości dysparycji (mogą także być ujemne),
- U - zbiór par liczb całkowitych wyznaczających lokalne sąsiedztwo pewnego punktu o współrzędnych (x,y),

- $\otimes$ - operator Hamminga (liczba różnych bitów w porównywanych słowach binarnych).

Wartość miary Census $IC(i,j)$ dla danego wewnętrznego piksela obrazu o współrzędnych lokalnych (i,j) oraz jego najbliższego n-sąsiedztwa wyraża się ciągiem bitów:

$$IC(i,j) = b_{n^2-1}...b_k...b_3b_2b_1b_0 \tag{3.26}$$

dla $k \in [0,...,n^2-1] / \left\{ \lfloor \frac{n^2}{2} \rfloor \right\}$, gdzie $\lfloor \frac{k}{n} \rfloor$ oznacza dzielenie całkowitoliczbowe k przez n. Wartość b_k wyznacza się następująco:

$$b_k = \begin{cases} 1 & \Leftrightarrow \quad I\left(i - \lfloor \frac{n}{2} \rfloor + \lfloor \frac{k}{n} \rfloor, j - \lfloor \frac{n}{2} \rfloor + k \ mod \ n\right) \geq I(i,j) \\ 0 & \quad w \ przeciwnym \ razie \end{cases} \tag{3.27}$$

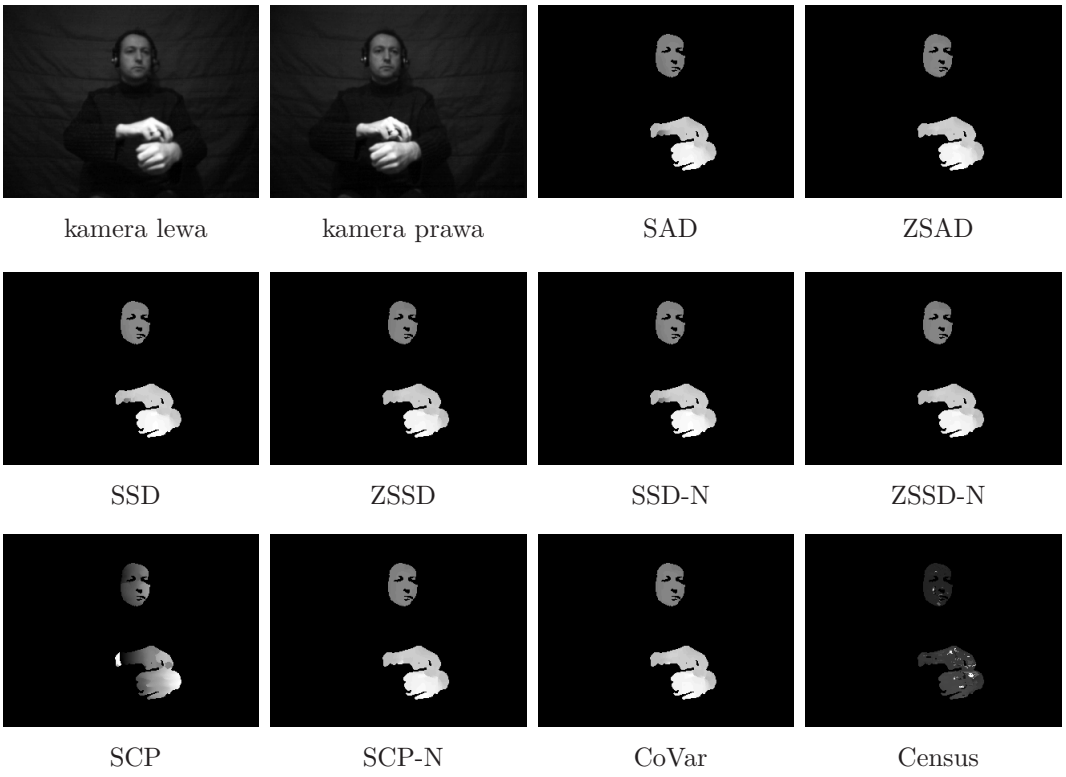
gdzie $i(i,j)$ jest wartością intensywności obrazu w punkcie o współrzędnych (i,j), zaś $k \ mod \ n$ oznacza resztę z dzielenia całkowitoliczbowego k przez n.

Dopasowanie pikseli wyznaczano dla okien o wymiarach 3x3, 5x5, ..., 31x31. Dla rozmiarów mniejszych od 17x17 otrzymywano błędne wartości dysparycji dla niektórych punktów należących do dłoni, twarzy oraz tła. Błędy te pojawiały się dla każdej z testowanych miar dopasowania. Przy większych rozmiarach okien wciąż pojawiały się niepoprawne wartości w charakteryzujących się słabszą teksturą obszarach tła, ale błędy w obszarach należących do dłoni i twarzy występowały tylko dla miar SCP i Census. Dlatego w kolejnych eksperymentach zmodyfikowano procedurę wyznaczania zwartej mapy dysparycji, tak aby dopasowania odbywały się jedynie dla pikseli należących do dłoni i twarzy, wyznaczonych na podstawie wejściowych obrazów kolorowych (podrozdział 3.1). Na rys. 3.10 przedstawiono zrektyfikowane obrazy wejściowe oraz wygenerowane na ich podstawie mapy dysparycji dla każdej z testowanych miar dopasowania. Okno korelacji miało rozmiar 17. Przedstawione obrazy pochodzą z sekwencji odpowiadającej słowu *skierowanie*. Wyznaczanie dysparycji tylko dla pikseli należących do dłoni i twarzy znacznie przyspieszyło obliczenia ale nadal czasy przetwarzania otrzymywane na typowym komputerze PC (AMD Athlon 1,81GHz i 1 GB RAM) nie pozwalały na zastosowanie metody w trybie on-line z częstotliwością akwizycji 25 klatek/s. W tab. 3.5 przedstawiono czasy generowania zwartych map dysparycji dla różnych miar dopasowania.

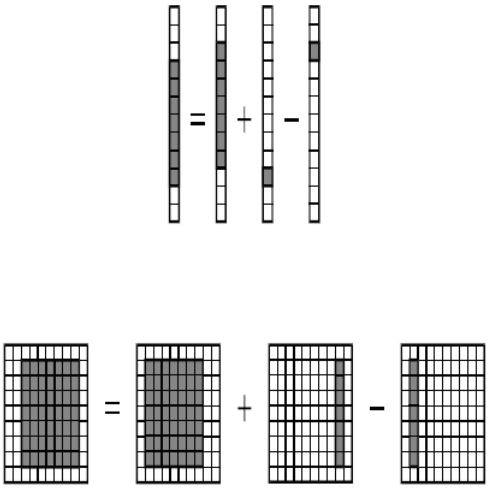
Tab. 3.5. *Czasy generowania zwartych map dysparycji na podstawie różnych miar dopasowania [s]; wielkość obrazu* 320×240*, wielkość okna korelacji:* 17×17

SAD	ZSAD	SSD	ZSSD	SSD-N	ZSSD-N	SCP	SCP-N	CoVar	Census
1.3	4.3	1.3	4.4	3.7	9.7	1.2	3.6	7.0	3.4

Wraz z przesuwaniem okna dopasowania dochodzi do powtarzania tych samych obliczeń. Dlatego w pracy [19] zaproponowano optymalizację algorytmu, polegającą na zapamiętaniu pośrednich sum, występujących we wzorach na miary dopasowań (patrz tab. 3.4), a następnie dodawaniu do tych sum wartości obliczonych dla pikseli, które w wyniku przesunięcia okna znalazły się w obszarze dopasowania i odjęciu



Rys. 3.10. *Obrazy oryginalne oraz zwarte mapy dysparycji dla różnych miar dopasowania.*



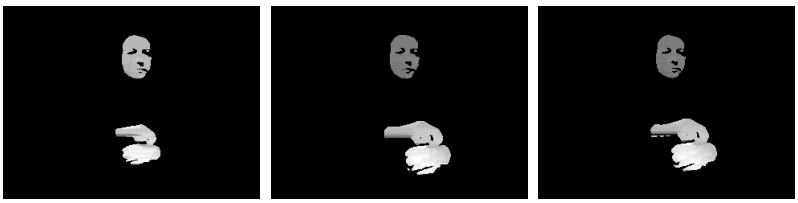
Rys. 3.11. *Graficzne przedstawienie założeń uproszczenia algorytmu wyznaczania miar dopasowań.*

wartości dla pikseli, które przestały należeć do tego obszaru (rys. 3.11).
W tab. 3.6 przedstawiono czasy obliczeń otrzymane po zastosowaniu opisywanej metody.
Otrzymane czasy wyznaczania map dysparycji nadal nie pozwalały na zastosowanie metody w trybie on-line.

Tab. 3.6. *Czasy generowania zwartych map dysparycji dla różnych miar dopasowań po optymalizacji algorytmu [s]; wielkość obrazu* 320×240*, wielkość okna korelacji:* 17×17

SAD	ZSAD	SSD	ZSSD	SSD-N	ZSSD-N	SCP	SCP-N	CoVar	Census
0.4	1.3	0.4	1.4	1.1	2.5	0.4	1.1	2.2	1.0

W związku z tym przetestowano także metodę wyznaczania mapy dysparycji z wykorzystaniem algorytmu opracowanego przez Stana Birchfielda [4]. W metodzie tej poszukiwanie odpowiadających sobie pikseli odbywa się niezależnie dla poszczególnych linii epipolarnych z wykorzystaniem programowania dynamicznego i dodatkowego przetwarzania otrzymanych map w celu wyeliminowania błędów. Dla metody tej otrzymano czasy przetwarzania rzędu 800 ms. Wadą algorytmu jest jednak konieczność doboru kilku parametrów sterujących procesem dodatkowego przetwarzania, otrzymywanych w pierwszej fazie map dysparycji. Okazuje się, że jakość otrzymywanych map dysparycji jest zależna od wartości tych parametrów. Dynamiczny charakter wykonywanych gestów i zmienność kątów nachylenia powierzchni dłoni powoduje, że trudno jest dobrać jeden zestaw parametrów, który dawałby zadawalające rezultaty dla wszystkich obrazów w rozpatrywanej sekwencji. Na rys. 3.12 przedstawiono trzy przykładowe mapy dysparycji pochodzące z sekwencji odpowiadającej słowu *skierowanie*, wyznaczone dla tych samych wartości parametrów.



Rys. 3.12. *Przykładowe mapy dysparycji dla obrazów z sekwencji odpowiadającej słowu skierowanie wyznaczone z wykorzystaniem algorytmu S. Birchfielda.*

Widoczne na pierwszym i trzecim obrazie zakłócenia w postaci poziomych pasów wynikają z niewłaściwego doboru parametrów.

Dlatego przetestowano także metodę generowania rzadkiej mapy dysparycji, w której miarę SAD zastosowano do obrazu krawędzi otrzymanego w wyniku filtracji LOG [42]. Metoda ta przy czasie przetwarzania 15 ms dla obrazów o rozdzielczości 320×240 i okna korelacji o wymiarach 17×17 dała mapy dysparycji o zadowalającej jakości (rys. 3.13) i została ostatecznie wybrana do dalszych eksperymentów.

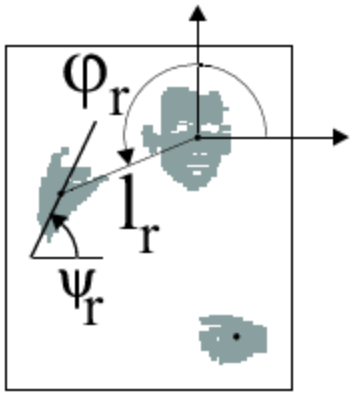
3.4 Wyznaczanie cech

Obrazy binarne zawierające obiekty odpowiadające dłoniom i twarzy oraz mapy dysparycji wykorzystano do budowy wektorów cech. Przyjęte składowe wektora cech można podzielić na cztery grupy: (1) opisującą położenie dłoni, (2) opisującą

Rys. 3.13. *Przykładowe rzadkie mapy dysparycji otrzymane z wykorzystaniem pakietu SVS.*

kształt dłoni, (3) opisującą orientację dłoni, (4) zawierającą informację przestrzenną. Położenie dłoni zostało określone z wykorzystaniem następujących parametrów (rys. 3.14):

- l_r - odległość środka ciężkości prawej dłoni od środka ciężkości twarzy,
- φ_r - orientacja odcinka łączącego środki ciężkości prawej dłoni i twarzy,
- l_l, φ_l - analogicznie dla dłoni lewej.



Rys. 3.14. *Składowe wektora cech opisujące położenie i orientację dłoni.*

Do opisu kształtu dłoni wykorzystano następujące parametry:

- S_r - pole powierzchni prawej dłoni,
- γ_r - współczynnik zwartości prawej dłoni,
- ε_r - ekscentryczność (niecentryczność) prawej dłoni,
- S_l, γ_l, ε_l - analogicznie dla dłoni lewej.

Współczynniki zwartości i ekscentryczności wyznaczano na podstawie wzorów [45, 71]:

$$\gamma = \frac{P^2}{4\pi S},\ \varepsilon = \frac{\left(m_{20} - m_{02}\right)^2 + 4m_{11}^2}{S^4} \tag{3.28}$$

gdzie: S - pole powierzchni, P - obwód, m_{11}, m_{20}, m_{02} - momenty centralne, które dla danego obiektu O w obrazie binarnym określa zależność [67]:

$$m_{pq} = \sum_{i \in O} \sum_{j \in O} (i - \bar{i})^p (j - \bar{j})^q \tag{3.29}$$

a $\bar{i}$, $\bar{j}$ to współrzędne środka ciężkości obiektu O. Dla koła parametry γ i ε są równe odpowiednio: 1 i 0. Dla elipsy są one funkcjami stosunku długości jej osi głównych. Orientację dłoni wyznaczono jako orientację osi głównej binarnego obiektu odpowiadającego dłoni, korzystając ze wzoru [28, 71]:

$$\psi = 0,5 tan^{-1} \frac{2m_{11}}{m_{20} - m_{02}} \tag{3.30}$$

Składowe wektora cech zawierające informację przestrzenną zostały określone jako:

$$\Delta Z_r = \bar{Z}_f - \bar{Z}_r, \ \Delta Z_l = \bar{Z}_f - \bar{Z}_l \tag{3.31}$$

gdzie: $\bar{Z}_f$, $\bar{Z}_r$, $\bar{Z}_l$ oznaczają, kolejno, średnią wartość głębi dla obiektów odpowiadających twarzy, dłoni prawej i dłoni lewej. Ponieważ wykorzystywano rzadkie mapy dysparycji, przy obliczaniu średniej wartości głębi twarzy i dłoni uwzględniano jedynie te piksele, dla których wartość dysparycji była określona.

Dla każdej dłoni mamy zatem po 7 parametrów. Teoretycznie daje to więc 2^{14} możliwych kombinacji cech. Jeżeli założymy, że dla obu dłoni uwzględniamy te same cechy (większość gestów PJM ma charakter dwuręczny i często obie dłonie są równorzędne przy wykonywaniu danego znaku), to otrzymamy 2^7 możliwości. Gdy następnie uznamy, że informacja o położeniu dłoni (l i φ) jest niezbędna otrzymamy $2^5 = 32$ warianty wektora cech (tab. 3.7). Na potrzeby niniejszej pracy oznaczono je numerami 1-32. Każdy z wektorów zawiera informację o położeniu. Wektory 2-8 wykorzystują informację o kształcie, wektory 9-15 utworzono, odpowiednio, z wektorów 2-8 przez dodanie informacji o głębi, wektory 16 - 29 utworzono z wektorów 2 - 15 przez dodanie informacji o orientacji. Pozostałe 3 wektory zawierają informację o: 30 - położeniu i głębi, 31 - położeniu i orientacji i 32 położeniu, głębi i orientacji.

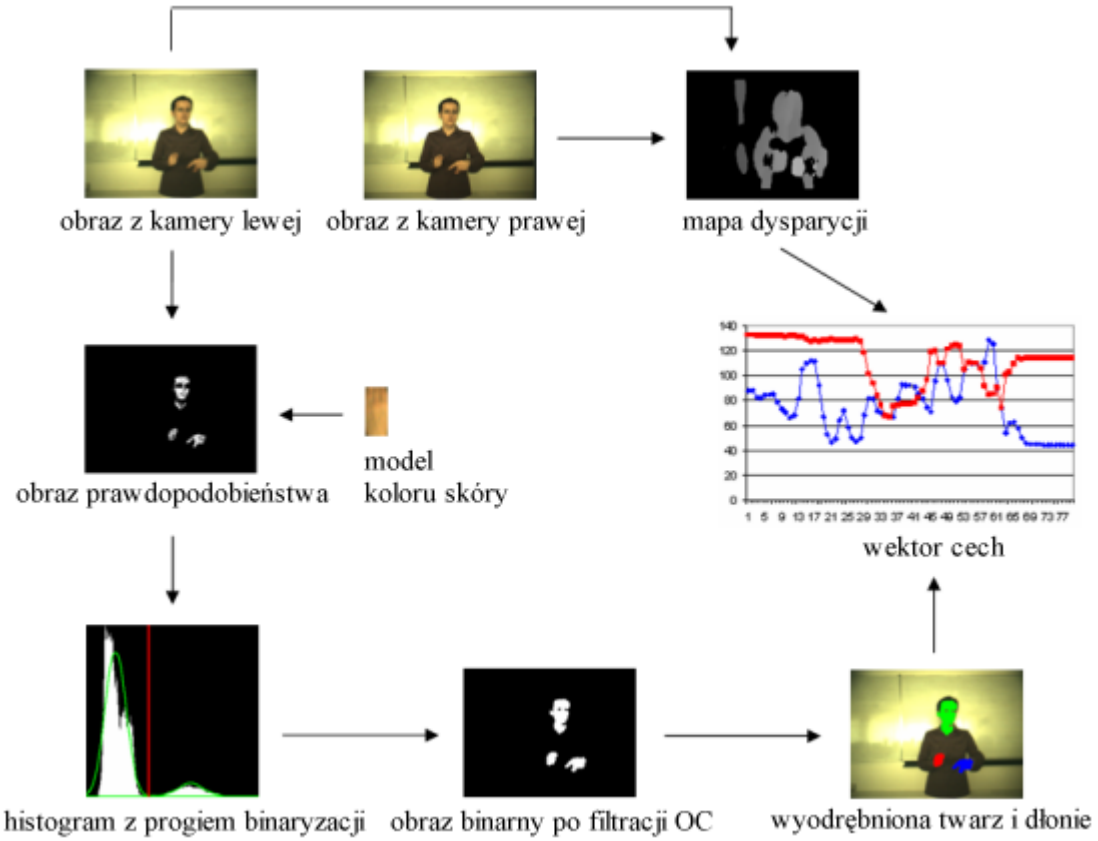
3.5 Podsumowanie

W rozdziale zaproponowano schemat przetwarzania obrazów w celu wyznaczania wektorów cech. Jego elementy uwidacznia przedstawiony dalej rys. 3.15. Wykorzystano obrazy kolorowe pozyskiwane w układzie stereowizyjnym. Do identyfikacji dłoni i twarzy osoby wykonującej gest wykorzystano metodę opartą o zbudowany uprzednio model rozkładu chrominancji skóry ludzkiej w znormalizowanej przestrzeni barw RGB. Wybór metody poprzedzony był studiami literaturowymi oraz opisanymi w rozdziale eksperymentami dla czterech różnych metod identyfikacji i dziewięciu wybranych przestrzeni barw o właściwościach istotnych z punktu widzenia segmentacji. Testy przeprowadzono z wykorzystaniem, przygotowanej w tym celu bazy danych, zawierającej 162 obrazy testowe, zarejestrowane przez dwie osoby

Tab. 3.7. *Warianty wektora cech*

	Wykorzystywane cechy						opis
	położenie	kształt			głębia	orientacja	
Nr	l_r, φ_r, l_l, φ_l	S_r, S_l	γ_r, γ_l	ε_r, ε_l	ΔZ_r, ΔZ_l	ψ_r, ψ_l	
1	+	-	-	-	-	-	położenie
2	+	+	-	-	-	-	
3	+	-	+	-	-	-	
4	+	-	-	+	-	-	położenie
5	+	+	+	-	-	-	+
6	+	+	-	+	-	-	kształt
7	+	-	+	+	-	-	
8	+	+	+	+	-	-	
9	+	+	-	-	+	-	
10	+	-	+	-	+	-	położenie
11	+	-	-	+	+	-	+
12	+	+	+	-	+	-	kształt
13	+	+	-	+	+	-	+
14	+	-	+	+	+	-	głębia
15	+	+	+	+	+	-	
16	+	+	-	-	-	+	
17	+	-	+	-	-	+	położenie
18	+	-	-	+	-	+	+
19	+	+	+	-	-	+	kształt
20	+	+	-	+	-	+	+
21	+	-	+	+	-	+	orientacja
22	+	+	+	+	-	+	
23	+	+	-	-	+	+	położenie
24	+	-	+	-	+	+	+
25	+	-	-	+	+	+	kształt
26	+	+	+	-	+	+	+
27	+	+	-	+	+	+	głębia
28	+	-	+	+	+	+	+
29	+	+	+	+	+	+	orientacja
30	+	-	-	-	-	+	położenie + orientacja
31	+	-	-	-	+	-	położenie + głębia
32	+	-	-	-	+	+	położenie + głębia + orientacja

o odmiennej karnacji skóry, w pomieszczeniu zamkniętym, w dzień słoneczny i pochmurny oraz w nocy przy oświetleniu sztucznym. Przetestowano metodę z aproksymacją histogramu kolorów za pomocą rozkładu normalnego, metodę z wygładzaniem histogramu kolorów filtrem Gaussa, metodę największej wiarygodności i metodę maksimum prawdopodobieństwa a posteriori. Rozważano znormalizowaną przestrzeń RGB, przestrzeń YUV, przestrzeń YIQ, przestrzeń barw przeciwstawnych OCS, przestrzeń barw przeciwstawnych w wersji logarytmicznej OCSL, przestrzeń



Rys. 3.15. *Schemat przetwarzania obrazów*

I1I2I3, przestrzeń IHS oraz przestrzeń Lab. Jakość otrzymywanych obrazów binarnych oceniano porównując je z obrazami otrzymanymi w wyniku ręcznej segmentacji wszystkich obrazów testowych. Przy wyborze metody i przestrzeni barw uwzględniono także czasy wykonania poszczególnych metod na typowym komputerze PC.

W rozdziale zaproponowano algorytm identyfikacji dłoni prawej, lewej i twarzy w otrzymanym obrazie binarnym. Algorytm ten wykorzystuje informację o polach powierzchni i środkach ciężkości otrzymanych obiektów binarnych w bieżącej i poprzedniej klatce. Przeprowadzone eksperymenty wykazały, że dla rozważanego słownika gestów, typowego tempa wykonywania oraz częstotliwości przetwarzania 25 klatek/s obiekty identyfikowane są poprawnie przy założeniu, że jeżeli gest rozpoczyna się od zetknięcia lub przysłonięcia obiektów, algorytm rozpoczyna działanie wcześniej, w sytuacji gdy dłonie i twarz dają w obrazie rozłączne obiekty i dłoń prawa znajduje się po prawej, a lewa po lewej stronie osi ciała (dokładniej linii pionowej przechodzącej przez środek ciężkości twarzy).

Ponieważ kształty przyjmowane przez dłonie oraz trajektorie ruchu dłoni mają charakter przestrzenny, dodano do wektora cech informację 3D. W tym celu wykorzystano, otrzymane w wyniku przetworzenia obrazów stereo, rzadkie mapy dysparycji. Wybór odpowiedniej metody poprzedzony był, opisanymi w niniejszym

rozdziale, testami różnych algorytmów. Przetestowano: (1) dające zwarte mapy dysparycji, korelacyjne metody poszukiwania odpowiedników dla okien korelacji o rozmiarach 3x3, 5x5, ..., 31x31 i dziesięciu różnych miar dopasowania scharakteryzowanych w tab. 3.4, (2) metody korelacyjne zmodyfikowane w ten sposób aby mapy dysparycji generowane były tylko dla obszarów dłoni i twarzy, (3) metodę zaproponowaną przez S. Birchfielda [4], polegającą na poszukiwaniu odpowiedników niezależnie dla poszczególnych linii epipolarnych z wykorzystaniem programowania dynamicznego i dodatkowego przetwarzania otrzymywanych map dysparycji i (4) metodę generowania rzadkiej mapy dysparycji na podstawie obrazów krawędzi otrzymanych w wyniku filtracji LOG. Oceniano wizualnie jakości otrzymywanych map dysparycji oraz czasy przetwarzania. Metody korelacyjne dla okien o rozmiarach 17x17 i większych umożliwiały otrzymywanie map dobrej jakości, ale czasy przetwarzania, nawet po zastosowaniu zaproponowanej w pracy [19] optymalizacji algorytmu lub ograniczeniu poszukiwania odpowiedników tylko do obszarów o barwie skóry, były dłuższe niż 400 ms, co uniemożliwiło zastosowanie tych metod do przetwarzania w trybie on-line. W przypadku metody S. Birchfielda problematyczny okazał się dobór takich parametrów algorytmu, które pozwoliłyby uzyskiwać mapy dysparycji dobrej jakości dla wszystkich klatek w danej sekwencji wideo. Czas przetwarzania 800 ms także nie pozwalał na zastosowanie tej metody w czasie rzeczywistym. Dopiero metoda generowania rzadkiej mapy dysparycji na podstawie obrazów krawędzi dała mapy zadowalającej jakości przy czasie przetwarzania rzędu 15 ms na typowym komputerze PC.

W rozdziale zaproponowano wektory cech, na podstawie których będzie dokonywane rozpoznawanie. Przesłanką do ich wyboru były dostępne w literaturze wyniki badań lingwistycznych nad tzw. wiązkami cech dystynktywnych, pozwalającymi jednoznacznie opisać znak migowy. Cechy podzielono na cztery grupy opisujące miejsce artykulacji, kształt dłoni, głębię i orientacje dłoni. Miejsce artykulacji, analogicznie jak w przypadku gestogramów (patrz podrozdział 2.2), określono względem innej części ciała. Jako odniesienie wybrano twarz, ponieważ podczas konwersacji przeprowadzanej z wykorzystaniem języka migowego zazwyczaj jest ona statyczna i musi być zwrócona w kierunku odbiorcy, tak aby możliwa była obserwacja wykonywanych gestów i ewentualnie czytanie z ruchu ust. Dodatkowym uzasadnieniem takiego wyboru, z punktu widzenia przetwarzania, jest fakt, że chrominancja twarzy jest zbliżona do chrominancji dłoni, co pozwala na zastosowanie tych samych metod identyfikacji. Przyjęto, że położenie dłoni względem twarzy będzie określone za pomocą długości odcinka łączącego środki ciężkości dłoni i twarzy i orientacji przechodzącej przez nie prostej. W odróżnieniu od rozpoznawania Polskiego Alfabetu Palcowego [47] rozpoznawanie wyrazów i zdań PJM wymaga, aby pole widzenia kamer obejmowało sylwetkę osoby wykonującej gest co najmniej od pasa w górę. W takim przypadku rozmiary dłoni w obrazie są zbyt małe, aby możliwa była dokładna analiza ich kształtu z uwzględnieniem układu i orientacji poszczególnych palców, tak jak ma to miejsce w zapisie gestograficznym. Dlatego przy wyborze cech opisujących kształt dłoni zdecydowano się na opis zgrubny z wykorzystaniem jedynie przybliżonego opisu danego kształtu. Zastosowano trzy miary: pole powierzchni, współczynnik zwartości i ekscentryczność i rozważono różne ich kombinacje. Po-

nieważ kształt dłoni i wykonywane ruchy mają charakter przestrzenny, dodano do wektora cech informację o wzajemnym usytuowaniu dłoni i twarzy osoby wykonującej gesty. Opis dłoni uzupełniono o orientację osi głównej odpowiadającego jej obiektu binarnego.

Wektor cech zawiera więc maksymalnie 14 elementów, po 7 na każdą dłoń. Cechy, które z lingwistycznego punktu widzenia są dystynktywne, mogą w przypadku przetwarzania obrazów te właściwości utracić w wyniku wystąpienia problemów, które opisano wcześniej. Ponadto w zaproponowanym wektorze cech opis kształtu jest znacznie uboższy aniżeli w gestograficznym zapisie statycznej konfiguracji dłoni. Dlatego nawet przy założeniu, że proces wyznaczania wartości cech będzie zawsze bezbłędny, nie można określić, w jakim stopniu wystarczają one do rozpoznawania przyjętego słownika gestów, bez przeprowadzenia eksperymentów. Eksperymenty związane z wyborem wektora cech opisano w podrozdziale 5.1.

Rozdział 4

Ukryte modele Markowa

Ukryty model Markowa (HMM - Hidden Markov Model) jest automatem o skończonej liczbie stanów, generujących łańcuchy obserwacji [13, 55]. Ukryte modele Markowa stanowią narzędzie wykorzystywane szeroko do modelowania szeregów czasowych. Spotyka się je w większości obecnych systemów rozpoznawania mowy, w licznych zastosowaniach komputerowej biologii molekularnej, w kompresji danych oraz innych obszarach sztucznej inteligencji i rozpoznawania obrazów. Ostatnio HMM znajdują zastosowanie w wizji komputerowej do modelowania sekwencji obrazów i śledzenia obiektów.

HMM opisuje dwa stochastyczne procesy: jednym jest nieobserwowalny łańcuch Markowa ze skończoną liczbą stanów, scharakteryzowany rozkładem prawdopodobieństwa stanu początkowego i macierzą prawdopodobieństw przejść pomiędzy stanami, drugim jest ciąg obserwacji generowany przez stany zgodnie z danymi rozkładami prawdopodobieństwa. Wykonanie gestu przez człowieka także wiąże się z dwoma procesami, z których pierwszy skojarzony jest z ukrytymi stanami mentalnymi powstającymi w mózgu, drugi zaś z obserwowanymi na zewnątrz akcjami. Dlatego obecnie coraz częściej HMM wykorzystywane są do rozpoznawania gestów [23, 54, 61].

4.1 Podstawy matematyczne

HMM jest narzędziem pozwalającym reprezentować rozkłady prawdopodobieństwa w ciągach obserwacji. Oznaczmy obserwację w chwili t przez y_t. Obserwacją może być symbol z dyskretnego alfabetu, zmienna rzeczywista lub całkowita, albo inny obiekt, nad którym można zdefiniować rozkład prawdopodobieństwa. Zakładamy, że obserwacje są dokonywane w dyskretnych, jednakowo odległych chwilach, więc t jest indeksem przyjmującym całkowite wartości, $t \in \{1, 2, ..., T\}$.
Nazwa HMM wynika z dwóch podstawowych założeń:

1. Obserwacja w chwili t została wygenerowana przez pewien proces, którego stan x_t jest ukryty dla obserwatora.

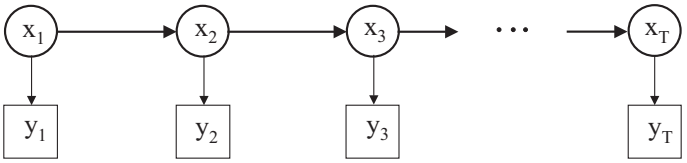
2. Stan x_t spełnia własność Markowa, tzn. zależy wyłącznie od x_{t-1} a ponadto obserwacja y_t zależy wyłącznie od x_t , tzn. nie zależy od stanów i obserwa-

cji w innych momentach. Tak więc stan x_t reprezentuje wszystko, co musimy wiedzieć o historii procesu, by móc przewidzieć przebieg tego procesu w przyszłości.

Wykorzystując wymienione własności można przedstawić łączny rozkład prawdopodobieństwa stanów i obserwacji w formie iloczynu:

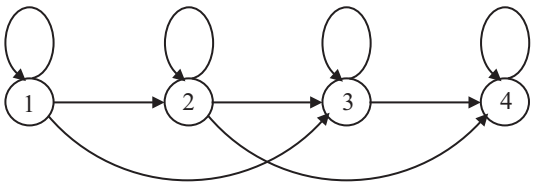
$$p\left(y_{1:T}, x_{1:T}\right) = p\left(x_1\right) p\left(y_1|x_1\right) \Pi_{t=2}^{T} p\left(x_t|x_{t-1}\right) p\left(y_t|x_t\right) \tag{4.1}$$

gdzie przez $\eta_{p:k}$ rozumie się ciąg η_p, η_{p+1}, ..., η_k, $1 \leq p < k \leq T$. Zapisowi (4.1) odpowiada interpretacja w formie sieci bayesowskiej [22, 31] przedstawionej na rys. 4.1, gdzie stany oznaczono kółkami, obserwacje kwadratami. Sieć ukazuje relacje niezależności warunkowej pozwalające na dokonanie faktoryzacji łącznego rozkładu. W iloczynie (4.1) występuje tyle rozkładów warunkowych, ile jest węzłów. Zmienna



Rys. 4.1. *Sieć bayesowska przedstawiająca relacje warunkowych niezależności w HMM.*

reprezentowana przez dany węzeł jest zależna od zmiennych z węzłów, z których do niego dochodzi łuk. W odniesieniu do HMM zakłada się, że zmienna stanu x_t przyjmuje wartości dyskretne ze zbioru $\{1, 2, ..., N\}$. Na rys. 4.2 przedstawiono przykładowy schemat topologiczny HMM z czterema stanami ($N = 4$).



Rys. 4.2. *Przykładowy HMM z czterema stanami.*

Tutaj łuki wskazują możliwe przejścia między stanami. Ustalenia liczby stanów i struktury modelu dokonuje się na ogół eksperymentalnie. Pokazany układ jednokierunkowy (tzw. model Bakisa [55]) jest typowy w modelowaniu szeregów czasowych. Dzięki istnieniu dodatkowych przejść z pominięciem pewnych stanów możliwe jest modelowanie sekwencji o różnej długości.

Praktyczne wykorzystanie HMM o zadanej strukturze wymaga znajomości:

- rozkładu prawdopodobieństwa stanu początkowego,
- prawdopodobieństw przejść między stanami,
- modelu obserwacji.

Rozkład prawdopodobieństwa stanu początkowego scharakteryzowany jest przez N - elementowy wektor Π, którego i - ty element π_i oznacza prawdopodobieństwo $P(x_1 = i)$. Prawdopodobieństwa przejść określa macierz $A = [a_{ij}]$ o wymiarach $N \times N$, gdzie $a_{ij} = P(x_{t+1} = j | x_t = i)$, $i, j = 1, 2, ..., N$.
Model obserwacji opisuje prawdopodobieństwo $P(y_t|x_t)$. Jeżeli obserwacje są dyskretnymi symbolami, którym można przypisać wartości 1, 2, ..., K, model ten jest w pełni opisany przez macierz prawdopodobieństw obserwacji $B = [b_{ik}]$ o wymiarach $N \times K$, gdzie

$$b_{ik} = P(y_t = k | x_t = i) \tag{4.2}$$

Gdy obserwacja jest wektorem liczb rzeczywistych, $B = [b_i(y_t)]$ staje się N wymiarowym wektorem funkcji gęstości rozkładów prawdopodobieństwa, przyjmowanych zazwyczaj jako suma rozkładów normalnych

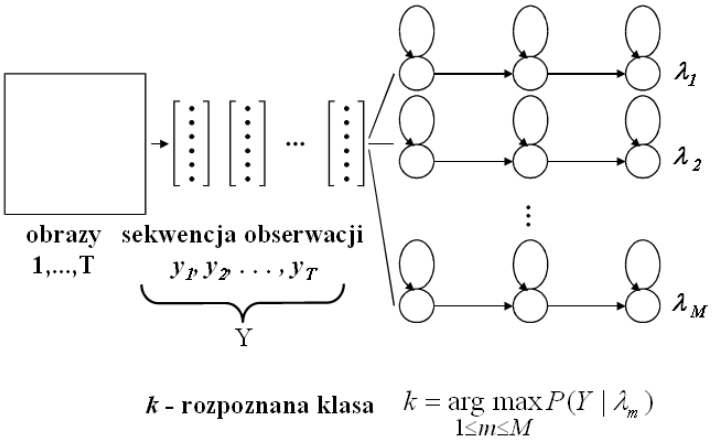
$$b_i(y_t) = p(y_t | x_t = i) = \sum_{s=1}^{S} c_{is} \mathcal{N}(y_t, \mu_{is}, \Sigma_{is}) \tag{4.3}$$

z wektorami wartości oczekiwanych μ_{is} i macierzami kowariancji Σ_{is}. Nieujemne współczynniki wagowe c_{is} dają w sumie 1, dla każdego i. Na ogół zakłada się, że macierz przejścia A oraz model obserwacji, który będziemy dalej oznaczać przez B, nie zależą od czasu.

Traktując HMM jako trójkę $\lambda = (A, B, \Pi)$ można zdefiniować następujące ważne problemy:

1. Uczenie: mając ciąg obserwacji $Y = y_{1:T}$ oraz strukturę modelu λ należy wyznaczyć parametry modelu λ maksymalizujące prawdopodobieństwo $P(Y|\lambda)$.

2. Klasyfikacja: dany ciąg obserwacji Y należy przypisać do klasy k reprezentowanej przez model λ_k ze zbioru znanych modeli λ_1, λ_2, ... λ_M, dla którego prawdopodobieństwo $P(Y|\lambda_k)$ przyjmuje wartość maksymalną w tym zbiorze. Ideę rozpoznawania z wykorzystaniem ukrytych modeli Markowa ilustruje rys. 4.3



Rys. 4.3. *Rozpoznawanie z wykorzystaniem ukrytych modeli Markowa.*

3. Dekodowanie: na podstawie ciągu obserwacji Y należy dla znanego modelu λ wyznaczyć najbardziej prawdopodobną sekwencję stanów $X^* = x^*_{1:T}$, tzn. maksymalizującą prawdopodobieństwo $P(Y, X|\lambda)$.

Rozwiązanie zadania 3 jest związane ze znanym z telekomunikacji algorytmem Viterbiego [20, 55], wykorzystującym metodę programowania dynamicznego. Rozważmy sekwencję $X = x_{1:T}$ ukrytych stanów. Miarą prawdopodobieństwa realizacji ciągu X w ukrytym modelu Markowa λ jest prawdopodobieństwo warunkowe $P(Y, X|\lambda)$, które określone jest wzorami (4.4), (4.5) i (4.6):

$$P(Y, X|\lambda) = \frac{P(Y, X, \lambda)}{P(\lambda)} = \frac{P(Y|X, \lambda) P(X, \lambda)}{P(\lambda)} = P(Y|X, \lambda) P(X|\lambda) \tag{4.4}$$

$$P(Y|X, \lambda) = P(y_1|x_1) P(y_2|x_2) ...P(y_T|x_T) \tag{4.5}$$

$$P(X|\lambda) = P(x_1) P(x_2|x_1) P(x_3|x_2) ...P(x_T|x_{T-1}) \tag{4.6}$$

Mamy zatem:

$$\begin{aligned} P(Y, X|\lambda) &= \Pi_{t=1}^{T} P(x_t|x_{t-1}) P(y_t|x_t) \\ P(x_1|x_0) &= P(x_1) \end{aligned} \tag{4.7}$$

Interesuje nas wyznaczenie takiej sekwencji X^*, dla której zachodzi:

$$P(Y, X^*|\lambda) = max_X P(Y, X|\lambda) \tag{4.8}$$

Zadanie (4.8) można zapisać w dwóch równoważnych wersjach:

$$(i)\ max_X \left\{ P(Y, X|\lambda) = \Pi_{t=1}^{T} \tilde{d}(t, x_t|t-1, x_{t-1}) \right\} \tag{4.9}$$

$$(ii)\ min_X \left\{ [-\log\ P(Y, X|\lambda)] = \sum_{t=1}^{T} d(t, x_t|t-1, x_{t-1}) \right\} \tag{4.10}$$

gdzie:

$$\tilde{d}(t, x_t|t-1, x_{t-1}) = P(x_t|x_{t-1}) P(y_t|x_t) \tag{4.11}$$

$$d(t, x_t|t-1|x_{t-1}) = -\log\ \tilde{d}(t, x_t|t-1, x_{t-1}) \tag{4.12}$$

Zarówno zadanie (4.9) jak i (4.10) można rozwiązać metodą programowania dynamicznego wykorzystującego zasadę optymalności Bellmana.

Niech $(t_p, s_p) \rightarrow (t_k, s_k)$ oznacza optymalną ścieżkę łączącą węzły (t_p, s_p) i (t_k, s_k) prostokątnej siatki, w której na osi odciętych zaznaczono momenty czasu, na osi rzędnych zaś numery identyfikujące stan (zob. też rys. 4.4), a $(t_p, s_p) \overset{(t,s)}{\rightarrow} (t_k, s_k)$ oznacza optymalną ścieżkę między węzłami (t_p, s_p) i (t_k, s_k) przechodzącą przez węzeł pośredni (t, s). Zasada optymalności Bellmana mówi, że:

$$(t_p, s_p) \overset{(t,s)}{\rightarrow} (t_k, s_k) = (t_p, s_p) \rightarrow (t, s) \oplus (t, s) \rightarrow (t_k, s_k) \tag{4.13}$$

gdzie: $t_p, t, t_k \in \{1, 2, ..., T\}$, $t_p < t < t_k$, $s_p, s, s_k \in \{1, 2, ..., N\}$ a znak $\oplus$ oznacza konkatenację ścieżek. Wprowadzamy następujące oznaczenia:

$$\tilde{D}(\tau, x_\tau) = max_{X_\tau} \Pi_{t=1}^{\tau} \tilde{d}(t, x_t|t-1, x_{t-1}) \tag{4.14}$$

$$D(\tau, x_\tau) = min_{X_\tau} \sum_{t=1}^{\tau} d(t, x_t | t-1, x_{t-1}) \tag{4.15}$$

gdzie $1 \leq \tau \leq T$, X_τ jest początkowym fragmentem sekwencji stanów kończącym się w stanie x_τ w chwili τ, tzn. $X_\tau = x_{1:\tau}$. Wartości początkowe dla funkcji $\tilde{D}$ i D są następujące:

$$\tilde{D}(1, x_1) = P(x_1) P(y_1 | x_1) \tag{4.16}$$

$$D(1, x_1) = -\log \left(\tilde{D}(1, x_1) \right) \tag{4.17}$$

Na podstawie (4.16) i (4.17) można zapisać następujące związki rekurencyne:

$$\tilde{D}(t+1, x_{t+1}) = max_{x_t} \tilde{D}(t, x_t) P(x_{t+1} | x_t) P(y_{t+1} | x_{t+1}) \tag{4.18}$$

$$D(t+1, x_{t+1}) = min_{x_t} \{ D(t, x_t) - \log (P(x_{t+1} | x_t)) - \log (P(y_{t+1} | x_{t+1})) \} \tag{4.19}$$

Optymalny stan końcowy x_T^* otrzymuje się z zależności:

$$x_T^* = arg\ max_{x_T} \tilde{D}(T, x_T) = arg\ min_{x_T} D(T, x_T) \tag{4.20}$$

a optymalną długość ścieżki:

$$\tilde{D}^* = \tilde{D}(T, x_T^*)\ lub\ D^* = D(T, x_T^*) \tag{4.21}$$

Oznaczając przez $\Psi(t+1, x_{t+1})$ przedostatni stan na optymalnej ścieżce kończącej się w chwili $t+1$ w stanie x_{t+1} otrzymuje się następującą zależność rekurencyjną:

$$\begin{aligned} \Psi(t+1, x_{t+1}) &= arg\ max_{x_t} \tilde{D}(t, x_t) P(x_{t+1} | x_t) P(y_{t+1} | x_{t+1}) \\ &= arg\ min_{x_t} \{ D(t, x_t) - \log (P(x_{t+1} | x_t)) - \log (P(y_{t+1} | x_{t+1})) \} \end{aligned} \tag{4.22}$$

Z przedstawionych rozważań wynika algorytm Viterbiego:

Algorytm 4.1: Algorytm Viterbiego

Ustalenie wartości początkowych:

For i=1, 2, ..., N
  $D(1, i) = -log\ P(x_1 = i) - log\ P(y_1 | x_i)$
  $\Psi(1, i) = 0$
end i

Pętla główna:

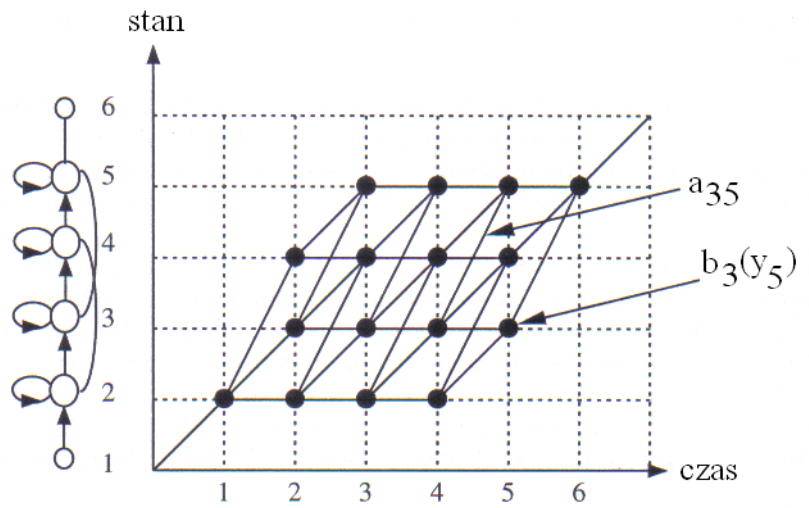
For t=1, 2, ..., $T-1$
  For x_t=1, 2, ..., N
    Wyznacz $D(t+1, x_{t+1})$ według (4.19)
    Zapamiętaj $\Psi(t+1, x_{t+1})$ wyliczone według (4.22)
  end x_t
end t

Zakończenie:

Długość optymalnej ścieżki D^* jest określona przez (4.21)
Najlepsza sekwencja stanów X^* jest wyznaczona następująco:
Wyznacz x_T^* na podstawie (4.20)
For t=$T-1$, $T-2$, ... 1
$x_t^* = \Psi\left(t+1, x_{t+1}^*\right)$
end t

Zabieg polegający na logarytmowaniu pozwala uniknąć problemów numerycznych wynikających z mnożenia małych liczb.

Algorytm ten może zostać zobrazowany jako znajdowanie najlepszej ścieżki na siatce, której wymiar pionowy reprezentuje stany ukrytego modelu Markowa, zaś wymiar poziomy momenty wystąpienia kolejnych wektorów obserwacji (rys. 4.4).



Rys. 4.4. *Graficzna interpretacja algorytmu Viterbiego.*

Czarny węzeł na rysunku reprezentuje logarytm prawdopodobieństwa pojawienia się danego wektora obserwacji w danym stanie, zaś łącząca węzły linia odpowiada logarytmowi prawdopodobieństwa przejścia pomiędzy stanami. Logarytm prawdopodobieństwa dla danej ścieżki jest wyznaczany poprzez zsumowanie logarytmów prawdopodobieństw przejść i logarytmów prawdopodobieństw obserwacji wzdłuż tej ścieżki. Ścieżka jest rozszerzana od lewej do prawej, kolumna po kolumnie. W odniesieniu do HMM algorytm ma złożoność $O\left(TN^2\right)$. Metodę Viterbiego stosuje się często bezpośrednio w zadaniach 1 i 2. W pierwszym, potrzebne prawdopodobieństwa wyznacza się wtedy zliczając przejścia i obserwacje dotyczące tylko stanów z optymalnej sekwencji X^* i modyfikując model aż do uzyskania zbieżności (uczenie metodą Viterbiego [13]). W zadaniu 2 natomiast rozpoznanie opiera się na prawdopodobieństwie $P\left(Y, X^*|\lambda\right)$, które ma dominujący udział przy wyznaczaniu brzego-

wej wartości $P(Y|\lambda)$. Bardziej zaawansowany sposób uczenia zaproponowany przez Bauma i Welcha stanowi wersję metody EM. Sprowadza się on do wyznaczania prawdopodobieństw $P(y_{1:t}, x_t = i|\lambda)$ oraz $P(y_{t+1:T}|x_t = i, \lambda)$ metodami, odpowiednio, w przód, wstecz - pokrewnymi do metody Viterbiego [55]. Parametry modelu poprawia się iteracyjnie, bazując na ostatnio zaktualizowanych. W kolejnych krokach następuje wzrost prawdopodobieństwa $P(Y|\lambda)$ aż do zbieżności w maksimum lokalnym. Metoda nie gwarantuje uzyskania rozwiązania globalnego. Prawdopodobieństwo $P(Y|\lambda)$, występujące także w zadaniu klasyfikacji wyznacza się metodą w przód. Bliższe informacje zamieszczono w następnym podrozdziale.

4.2 Uczenie ukrytych modeli Markowa

Jak już wspomniano, do uczenia ukrytych modeli Markowa wykorzystuje się najczęściej metodę Bauma Welcha lub metodę Viterbiego. W niniejszej pracy zastosowano przybornik HTK, w którym uczenie odbywa się w dwóch etapach z wykorzystaniem obu tych metod. W dalszej części podrozdziału opisano krótko ideę uczenia zastosowaną w HTK. Bardziej szczegółowe informacje wraz z formułami matematycznymi można znaleźć w pracach [55, 81].

Na podstawie ciągu obserwacji $Y = y_{1:T}$ dokonuje się reestymacji założonego modelu λ uzyskując nowy model $\hat{\lambda}$, dla którego zachodzi:

$$P\left(Y|\hat{\lambda}\right) > P(Y|\lambda) \tag{4.23}$$

Procedurę powtarza się do uzyskania maksimum. W HTK w początkowej fazie uczenia parametry modelu wyznaczane są metodą Viterbiego. Na początku oblicza się prawdopodobieństwo $P(Y, X^*|\lambda)$ z wykorzystaniem algorytmu Viterbiego (patrz podrozdział 4.1). W trakcie, dla każdych dwóch stanów i, j wyznaczana jest liczba przejść ze stanu i do stanu j: n_{ij}. Prawdopodobieństwa przejść obliczane są na podstawie zależności:

$$\hat{a}_{ij} = \frac{n_{ij}}{\sum_{k=1}^{N} n_{ik}} \tag{4.24}$$

Znajomość optymalnej sekwencji stanów X^* umożliwia przyporządkowanie obserwacji do stanów. Po tym przyporządkowaniu w przypadku modelu obserwacji ciągłej w obrębie każdego stanu dokonywana jest klasteryzacja metodą k-średnich [16] w celu przypisania obserwacji do S grup, z których każda modelowana będzie za pomocą pojedynczego rozkładu Gaussa, jak przyjęto w (4.3). Wartość średnia $\hat{\mu}_{is}$ i wariancja $\hat{\Sigma}_{is}$ dla rozkładu s w stanie i wyznaczane są na podstawie znanych zależności, z wykorzystaniem tylko tych obserwacji, które zostały przyporządkowane do danej grupy:

$$\hat{\mu}_{is} = \frac{1}{T_{is}} \sum_{y \in Y_{is}} y \tag{4.25}$$

$$\hat{\Sigma}_{is} = \frac{1}{T_{is}} \sum_{y \in Y_{is}} (y - \hat{\mu}_{is})(y - \hat{\mu}_{is})^T \tag{4.26}$$

gdzie Y_{is} jest zbiorem obserwacji przyporządkowanych do rozkładu s w stanie i natomiast T_{is} jest liczbą elementów w zbiorze Y_{is}. Wagi poszczególnych rozkładów

Gaussa wyznaczane są na podstawie liczby obserwacji przyporządkowanych do danej grupy:

$$\hat{c}_{is} = \frac{T_{is}}{T_i} \tag{4.27}$$

gdzie T_i jest liczbą obserwacji przyporządkowanych do stanu i. Metodę Viterbiego można łatwo rozszerzyć na przypadek wielu ciągów obserwacji, kumulując w licznikach i mianownikach wzorów (4.24) - (4.27) odpowiednie liczby przejść i obserwacji otrzymane dla każdego ciągu z osobna.

W HTK po wstępnej estymacji parametrów metodą Viterbiego modele douczane są metodą Bauma-Welcha. Niech $\alpha_t(i)$ oznacza tzw. prawdopodobieństwo w przód:

$$\alpha_t(t) = P(y_{1:t}, x_t = i|\lambda) \tag{4.28}$$

Wyznacza się je na podstawie zależności rekurencyjnej:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(y_{t+1}) \tag{4.29}$$

z warunkiem początkowym:

$$\alpha_1(i) = \pi_i b_i(y_1) \tag{4.30}$$

Niech z kolei $\beta_t(i)$ określa tzw. prawdopodobieństwo wstecz:

$$\beta_t(i) = P(y_{t+1:T}|x_t = i, \lambda) \tag{4.31}$$

Prawdopodobieństwo to może zostać wyznaczone na podstawie zależności rekurencyjnej:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1})\, \beta_{t+1}(j) \tag{4.32}$$

przy czym:

$$\beta_T(i) = 1 \tag{4.33}$$

Oznaczmy przez $\gamma_t(i)$ prawdopodobieństwo przebywania w stanie i w chwili t dla modelu λ i sekwencji obserwacji Y:

$$\gamma_t(i) = P(x_t = i|Y, \lambda) \tag{4.34}$$

które może zostać wyznaczone na podstawie prawdopodobieństw w przód i wstecz:

$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{P(Y|\lambda)} = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)} \tag{4.35}$$

Niech $\xi_t(i,j)$ oznacza prawdopodobieństwo przebywania w stanie i w chwili t i w stanie j w chwili $t+1$ dla modelu λ i sekwencji obserwacji Y:

$$\xi_t(i,j) = P(x_t = i, x_{t+1} = j|Y, \lambda) \tag{4.36}$$

Można je wyznaczyć z wykorzystaniem prawdopodobieństw w przód i wstecz:

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(y_{t+1})\, \beta_{t+1}(j)}{P(Y|\lambda)} = \frac{\alpha_t(i)\, a_{ij} b_j(y_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(y_{t+1})\, \beta_{t+1}(j)} \tag{4.37}$$

Prawdopodobieństwa $\gamma_t(i)$ i $\xi_t(i,j)$, powiązane zależnością

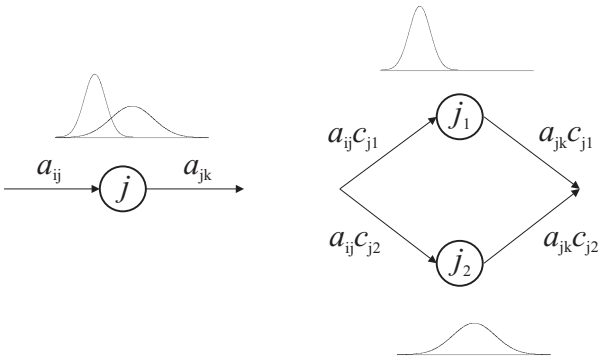
$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{4.38}$$

wykorzystuje się do estymacji wektora stanu początkowego Π i macierzy prawdopodobieństw przejść A:

$$\hat{\pi}_i = \gamma_1(i) \tag{4.39}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.40}$$

W celu wyznaczenia parametrów modelu obserwacji (4.3) przyjmuje się, że stan dla którego funkcja gęstości rozkładu prawdopodobieństwa obserwacji ma postać sumy rozkładów normalnych może zostać potraktowany jako grupa podstanów z rozkładami pojedynczymi. Prawdopodobieństwa przejść do tych podstanów powstają wskutek przemnożenia prawdopodobieństwa przejścia do danego stanu przez wartości wag poszczególnych rozkładów normalnych. Na rys. 4.5 pokazano to dla przykładu, gdy funkcja gęstości rozkładu prawdopodobieństwa jest sumą dwóch rozkładów normalnych.



Rys. 4.5. *Zastąpienie stanu z funkcją gęstości rozkładu prawdopodobieństwa w postaci sumy dwóch rozkładów Gaussa dwoma podstanami z rozkładami pojedynczymi.*

Problem wyznaczenia parametrów modelu obserwacji sprowadza się zatem do znalezienia wartości średnich i wariancji dla każdego podstanu.

Prawdopodobieństwo wystąpienia danej sekwencji obserwacji w danym modelu jest sumą prawdopodobieństw po wszystkich możliwych sekwencjach stanów X:

$$P(Y|\lambda) = \sum_{X} (Y, X|\lambda) \tag{4.41}$$

Każdy wektor obserwacji y_t uczestniczy w wyznaczaniu najbardziej prawdopodobnych parametrów dla każdego stanu. Zatem zamiast przyporządkowywać poszczególne wektory do określonych podstanów, każda obserwacja może zostać przyporządkowana do podstanu z pewną wagą równą prawdopodobieństwu, że model jest w danym podstanie, gdy interesujący nas wektor został zaobserwowany. Prowadzi to do następujących wzorów na parametry modelu obserwacji:

$$\hat{\mu}_{is} = \frac{\sum_{t=1}^{T} \gamma_t(i)\, y_t w_{is}(t)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{4.42}$$

$$\hat{\Sigma}_{is} = \frac{\sum_{t=1}^{T} \gamma_t(i)\,(y_t - \hat{\mu}_i)\,(y_t - \hat{\mu}_i)^T\, w_{is}(t)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{4.43}$$

$$\hat{c_{is}} = \frac{\sum_{t=1}^{T} \gamma_t(i)\, w_{is}(t)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{4.44}$$

gdzie:

$$w_{is}(t) = \frac{c_{is}\mathcal{N}(y_t, \mu_{is}, \Sigma_{is})}{\sum_{s=1}^{S} c_{is}\mathcal{N}(y_t, \mu_{is}, \Sigma_{is})} \tag{4.45}$$

a S oznacza liczbę rozkładów normalnych w modelu obserwacji.

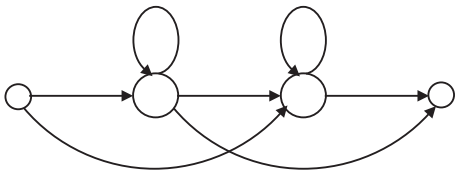
Algorytm uczenia metodą Bauma Welcha jest następujący:

Algorytm 4.2: Algorytm Bauma Welcha

1. Dla każdego parametru danego modelu zaalokuj pamięć na wartości sum z licznika i mianownika wzorów (4.39), (4.40), (4.42), (4.43) i (4.44). Przydzielone komórki pamięci będą dalej nazywane akumulatorami.
2. Pobierz sekwencję ze zbioru uczącego.
3. Dla każdej chwili czasowej t i dla każdego stanu i:
   - 3.1 Wyznacz prawdopodobieństwa $\alpha_t(i)$ (4.29, 4.30) i $\beta_t(i)$ (4.32, 4.33).
   - 3.2 Wyznacz prawdopodobieństwa $\gamma_t(i)$ (4.35) i $\xi_t(i,j)$ (4.37).
   - 3.3 Uaktualnij wartości akumulatorów.
4. Powtarzaj punkty 2 i 3 dla każdej sekwencji ze zbioru uczącego odpowiadającej danej klasie.
5. Wyznacz nowe wartości parametrów wykorzystując wartości akumulatorów.
6. Jeżeli wartość $P(Y|\lambda)$ wzrasta dostatecznie mało w stosunku do iteracji poprzedniej zakończ algorytm; w przeciwnym razie powtarzaj powyższe kroki, wykorzystując reestymowane wartości parametrów.

4.3 Modelowanie złożonych procesów

Złożone procesy modeluje się zwykle wyodrębniając prostsze elementy i łącząc ich modele w odpowiednie struktury (sieci HMM). Dotyczy to np. modelowania całych zdań złożonych z sekwencji pojedynczych słów (zob. rozdz. 6). W tym celu bardzo często na początku i na końcu modelu Markowa o topologii Bakisa dodaje się tzw. stany nieemitujące odpowiadające odpowiednio chwilom czasowym $1 - \delta t$ i $T + \delta t$ (rys. 4.6). Stany te nie generują obserwacji a jedynie są pomocne podczas budowania sieci modeli.



Rys. 4.6. *Model HMM z dwoma stanami emitującymi i dwoma nieemitującymi.*

Wykonanie gestu wyizolowanego może znacznie odbiegać od jego wykonania w sekwencji. Podobnie jak w przypadku sygnału mowy, zachodzi tutaj zjawisko koartykulacji, polegające na tym, że dłoń kończąc wykonanie jednego gestu mimowolnie przygotowuje się do wykonania gestu następnego [1]. Zjawisko to występuje szczególnie wtedy, gdy gesty wykonywane są spontanicznie w sposób naturalny i objawia się zniekształceniami układu i pozycji dłoni w początkowych i końcowych fazach gestu. Dlatego ważne jest, aby poszczególne modele wchodzące w skład złożonej struktury uczone były na wykonaniach gestów pochodzących z rzeczywistych sekwencji. Przygotowanie danych uczących wymaga więc ręcznego wyodrębniania poszczególnych gestów, co dla dużych systemów rozpoznających jest zadaniem pracochłonnym. Ponadto wskutek zjawiska koartykulacji ręczne wskazanie granic pomiędzy poszczególnymi słowami w zdaniu nie zawsze jest oczywiste i jednoznaczne. Podobna sytuacja występuje w przypadku mowy. Dlatego opracowano specjalną strategię uczenia z wbudowanym wyodrębnianiem elementów składowych w ciągu uczącym, a więc np. wyrazów w zdaniach (*embedded training*) [81]. Opiera się ono na algorytmie Bauma-Welcha, ale dostrajanie poszczególnych modeli odbywa się równolegle z wykorzystaniem tworzonych dynamicznie struktur złożonych z sekwencji modeli odpowiadających sekwencjom np. wyrazów w zdaniach ze zbioru uczącego.

Algorytm 4.3: Uczenie z wbudowanym podziałem zdań uczących na wyrazy

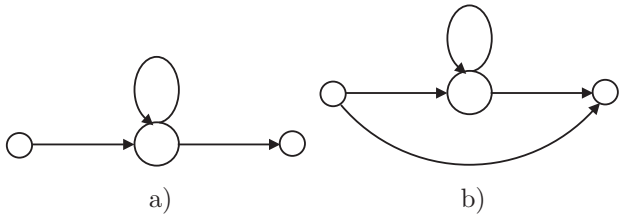
1. Dla każdego parametru każdego z modeli HMM przydziel pamięć na wartości sum z licznika i mianownika.

[1] Jak zaznaczono w podrozdziale 1.3, w niniejszej pracy gest jest utożsamiany z przedstawieniem wyrazu, a sekwencja gestów z przedstawieniem zdania.

2. Pobierz sekwencję ze zbioru uczącego.
3. Zbuduj złożony model HMM składający się z połączonych odpowiednio modeli wyrazów wchodzących w skład sekwencji.
4. Traktując otrzymany w punkcie 3 układ jako jeden duży model HMM, dla każdej chwili czasowej t i dla każdego stanu j:
   4.1 Wyznacz prawdopodobieństwa $\alpha_t(i)$ (4.29, 4.30) i $\beta_t(i)$ (4.32, 4.33).
   4.2 Wyznacz prawdopodobieństwa $\gamma_t(i)$ (4.35) i $\xi_t(i,j)$ (4.37).
   4.3 Uaktualnij wartości akumulatorów.
5. Powtarzaj punkty 3 i 4 dla każdej sekwencji ze zbioru uczącego.
6. Wykorzystując wartości akumulatorów wyznacz nowe estymaty parametrów wszystkich modeli.

W tym przypadku nie jest wymagana dokładna segmentacja danej sekwencji, a jedynie informacja o kolejności gestów, z których jest ona zbudowana. Taka strategia uczenia ułatwia proces gromadzenia danych uczących, co ma duże znaczenie, zwłaszcza w przypadkach, gdy zbiór rozpoznawanych zdań jest liczny.

Miejsca artykulacji i konfiguracje dłoni następujących po sobie słów mogą znacznie się różnić. W takim przypadku przejście pomiędzy tymi gestami wymaga wykonania dodatkowego ruchu. Dlatego często wprowadza się do sieci rozpoznającej dodatkowe modele odwzorowujące ten ruch, nazywane modelami przejścia. Na rys. 4.7 przedstawiono jednostanowe modele przejścia wykorzystywane w niniejszej pracy. Model z rys. 4.7b, nazywany modelem typu 'Tee', ma dodatkowe połączenie pomiędzy pierwszym i ostatnim stanem nieemitującym.



Rys. 4.7. *Modele przejścia z jednym stanem emitującym.*

Przy rozpoznawaniu sekwencji gestów wykorzystuje się często model języka bigram uwzględniający statystyczną informację o następstwie gestów [13, 40, 56]. Prawdopodobieństwa bigram opisują wtedy przejścia pomiędzy stanami nieemitującymi modeli odpowiadających następującym po sobie gestom. Prawdopodobieństwa te można wyznaczyć na podstawie zbioru uczącego z wykorzystaniem następującej formuły [81]:

$$P(j,i) = \begin{cases} \frac{R(j,i)-D}{R(j)} & dla \quad R(j,i) > t \\ b(j)\,p(i) & dla \quad R(j,i) \leq t \end{cases} \tag{4.46}$$

gdzie: $R(j,i)$ określa ile razy w zbiorze sekwencji uczących gest j wystąpił po geście i, zaś $R(j)$ jest liczbą wystąpień gestu j. Jeżeli częstość wystąpień pewnej pary gestów jest mniejsza niż przyjęty próg t, zamiast prawdopodobieństwa bigram wykorzystuje się prawdopodobieństwo unigram $p(i)$ określone na podstawie częstości wystąpień danego gestu w zbiorze uczącym. Współczynnik $b(j)$ dobiera się tak, aby odpowiednie prawdopodobieństwa dawały w sumie 1. Zazwyczaj przyjmuje się $D = 0.5$. Zabieg ten jest konieczny, ponieważ w przeciwnym razie dla niektórych par następujących po sobie gestów prawdopodobieństwa bigram byłyby bardzo małe, bądź wręcz zerowe. W konsekwencji prawdopodobieństwa tranzycji pomiędzy odpowiadającymi im modelami byłyby znikome, co praktycznie uniemożliwiłoby rozpoznawanie zawierających je sekwencji.

Do rozpoznawania sekwencji gestów często stosuje się algorytm Viterbiego w wersji z przekazywaniem znaczników. W algorytmie tym, każdy stan i posiada w każdej chwili t znacznik, z którym skojarzone jest prawdopodobieństwo $P(y_{1:t}, x_t = i|\lambda)$. Dla chwili $t+1$ wykonywane są następujące działania:

Algorytm 4.4: Algorytm Viterbiego z przekazywaniem znaczników

1. Dla każdego stanu i kopia jego znacznika jest przekazywana do wszystkich połączonych z nim stanów j. W trakcie przejścia znacznika skojarzone z nim prawdopodobieństwo, wyrażone w skali logarytmicznej, jest korygowane o wartość $log\,[a_{ij}] + log\,[b_j\,(y_{t+1})]$.

2. Dla każdego stanu j zgromadzone w nim znaczniki są porównywane i pozostawiany jest tylko jeden, z najwyższą wartością prawdopodobieństwa $P(y_{1:t+1}, x_{t+1} = j|\lambda)$.

Zaletą wersji algorytmu Viterbiego z przekazywaniem znaczników jest to, że z danym znacznikiem można powiązać jeszcze dodatkowe informacje o sekwencji przebytych stanów i słów. Po wykonaniu algorytmu sekwencja stanów przebytych przez umieszczony w ostatnim stanie znacznik pozwala na odtworzenie rozpoznawanego zdania i wyznaczenie granic pomiędzy wyrazami.

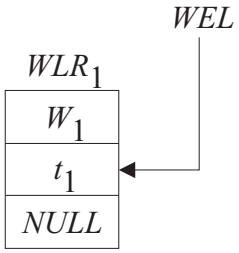
Każdy znacznik posiada wskaźnik o nazwie Word End Link (WEL). Jeżeli znacznik nie przebył jeszcze żadnego wyrazu, jego wskaźnik WEL wskazuje na miejsce puste: $WEL = NULL$. Jeżeli znacznik przechodzi przez nieemitujący stan kończący wyraz W_1 w chwili t_1, tworzona jest dynamicznie struktura o nazwie Word Link Record (WLR_1), której pola zawierają odpowiednio:

1. Identyfikator wyrazu, który znacznik właśnie przebył (W_1),

2. Chwilę czasową, w której znacznik opuszcza wyraz (t_1),

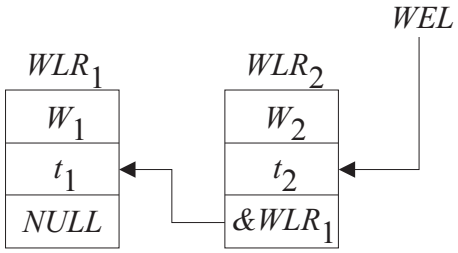
3. Bieżącą wartość wskaźnika WEL ($NULL$ - jeżeli jest to pierwszy przebywany wyraz).

Następnie uaktualniana jest wartość wskaźnika WEL tak, aby wskazywał na nowo utworzoną strukturę WLR_1: $WEL = \&WLR_1$ (rys. 4.8). Symbol & oznacza tu-



Rys. 4.8. *Wskaźnik WEL po przejściu przez znacznik wyrazu W_1 w chwili t_1.*

taj operator adresowy z języka C, który zwraca adres zmiennej w pamięci. Jeżeli następnie znacznik opuszcza wyraz W_2 w chwili t_2, tworzona jest kolejna struktura WLR_2 o wartościach pól, odpowiednio: W_2, t_2, $\&WLR_1$ oraz uaktualniana jest wartość wskaźnika WEL: $WEL = \&WLR_2$ (rys. 4.9). Po wykonaniu algorytmu

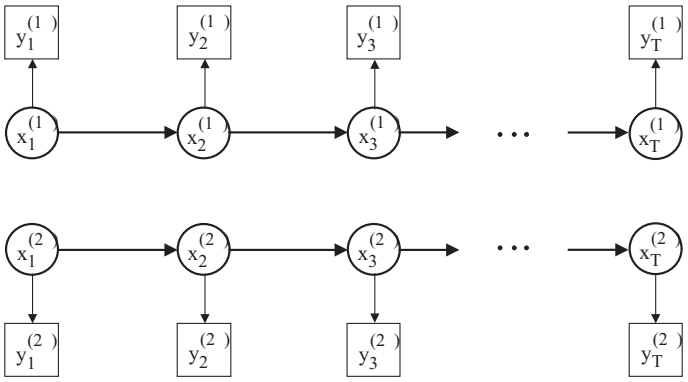


Rys. 4.9. *Wskaźnik WEL po przejściu przez znacznik wyrazu W_1 w chwili t_1 i wyrazu W_2 w chwili t_2.*

wskaźnik WEL zwycięskiego znacznika jest początkiem jednokierunkowej listy dynamicznej, której elementy opisują przebyte przez znacznik modele i chwile czasowe, w których to nastąpiło. Umożliwia to odtworzenie rozpoznanego zdania oraz przyporządkowanie obserwacji do stanów.

4.4 Równoległy ukryty model Markowa

Ważnym zagadnieniem związanym z HMM jest wybór modelu. Mogą się zdarzyć sytuacje, kiedy obserwacje powstają w wyniku równoczesnego działania kilku procesów o odmiennej złożoności i dynamice. Rozważmy np. O obiektów w sekwencji obrazów, zakładając, że każdy obiekt może przyjmować L kombinacji położeń i orientacji. Do zbudowania odpowiedniego HMM należałoby użyć O^L stanów. Ustalenie powiązań między nimi nie jest jednak łatwe. Ponadto duża liczba stanów nie tylko zwiększa złożoność obliczeniową, ale przede wszystkim utrudnia uczenie modelu. Przy niedostatecznie reprezentatywnym zbiorze uczącym może być powodem przeuczenia, ujawniającego się dobrym dopasowaniem do tych danych przy słabej jednocześnie zdolności uogólniania. Rozwiązaniem może się okazać zastosowanie innych modeli Markowa. Najprostszym od strony obliczeniowej jest model równoległy PaHMM, składający się z niezależnych zwykłych HMM. Na rys. 4.10 przedstawiono sieć Bayesowską odpowiadającą PaHMM uwzględniającemu dwa procesy równoległe. Każdy



Rys. 4.10. *PaHMM odpowiadający dwu procesom równoległym.*

z modeli składowych ma własną strukturę geometryczną i generuje własny ciąg obserwacji. Ciągi obserwacji są jednakowo liczne, jednak generujące je zmienne stanu są niezależne. Pozwala to traktować modele odrębnie w fazie uczenia a klasyfikację opierać na iloczynie prawdopodobieństw. Modele PaHMM stanowią krok w kierunku uwzględnienia równoległych procesów. Znane są również inne propozycje [22], które jednak wymagają bardziej złożonych algorytmów obliczeniowych. Model PaHMM jest np. szczególnym przypadkiem tzw. CHMM (coupled HMM), gdzie stan w danym kanale w chwili t może zależeć od stanów z wszystkich kanałów w chwili $t-1$. W innej wersji - FHMM (factorial HMM) - stany w chwili t ze wszystkich wzajemnie niezależnych kanałów mają wpływ na tę samą obserwację y_t.

PaHMM został wprowadzony przy rozpoznawaniu mowy przez Bourlarda i Duponta w pracy [6]. Sygnał mowy podzielono tam na pasma, które były modelowane niezależnie w celu wyeliminowania zakłóconych, niewiarygodnych fragmentów. Do rozpoznawania gestów po raz pierwszy model ten został zastosowany przez Voglera i Metaxasa [75]. Równoległy model Markowa modeluje M_p procesów za pomocą M_p niezależnych modeli z oddzielnymi wyjściami. Modele dla poszczególnych procesów są uczone niezależnie. Do rozpoznawania stosowana jest wersja algorytmu Viterbiego z przekazywaniem znaczników [81], którą wyjaśniono w podrozdziale 4.3. Algorytm ten musi zostać zmodyfikowany dla stanów nieemitujących, w których następuje połączenie modeli poszczególnych procesów niezależnych. W stanie takim znaczniki przychodzące z kanałów modelu równoległego są zastępowane jednym znacznikiem o wartości prawdopodobieństwa będącej iloczynem prawdopodobieństw znaczników zastępowanych. Ponieważ z danego kanału może przyjść w danej chwili czasowej więcej znaczników, rozważane są wszystkie możliwe kombinacje, przy czym każdorazowo bierze się po jednym znaczniku dla każdego z kanałów. Następnie z grupy tak otrzymanych znaczników wybierany jest jeden lub więcej znaczników wygrywających, które przekazane mogą być do kolejnych modeli PaHMM w sieci. Ponieważ w przypadku sieci modeli PaHMM te same stany nieemitujące łączą także poszczególne modele w sieci, konieczne jest wprowadzenie dodatkowego ograniczenia. Łączone mogą być jedynie znaczniki, dla których sekwencje przebytych modeli są identyczne.

Sekwencje obserwacji dla poszczególnych kanałów modelu równoległego są inne. W rezultacie optymalne ścieżki wyznaczone dla każdego z kanałów mogą przechodzić przez stany kończące wyrazy w różnych chwilach czasowych. Dla sieci modeli PaHMM informacje pochodzące z poszczególnych kanałów muszą być synchronizowane w stanach oznaczających końce wyrazów. Sekwencje stanów łączonych tam znaczników nie zawsze będą odpowiadać sekwencjom optymalnym. Należy się jednak spodziewać, że będą to sekwencje z grupy tych, dających najwyższe prawdopodobieństwa. Dlatego w przypadku modeli PaHMM konieczne jest zapamiętywanie dla każdego stanu i propagowanie w sieci kilku znaczników, z którymi skojarzone są najwyższe wartości prawdopodobieństw. W niniejszej pracy przyjęto, że w każdym stanie zapamiętywane są 4 znaczniki. Zwiększenie liczby znaczników nie spowodowało bowiem poprawy skuteczności rozpoznawania (zob. podrozdział 6.2).

W przypadku równoległego modelu Markowa poszczególne stany nie muszą zmieniać się w tych samych dyskretnych chwilach czasowych. Wykorzystywane modele mogą mieć różną topologię a zakłócenia w jednym strumieniu danych nie degradują wyniku tak silnie jak w przypadku pojedynczego ukrytego modelu Markowa.

4.5 Podsumowanie

W rozdziale opisano podstawy matematyczne ukrytych modeli Markowa, wykorzystywanych w niniejszej pracy do rozpoznawania wyrazów i zdań PJM. Rozpoczęto od typowego w modelowaniu szeregów czasowych układu jednokierunkowego (model Bakisa), w którym dzięki istnieniu dodatkowych przejść z pominięciem pewnych stanów możliwe jest modelowanie sekwencji o różnej długości. Ma to duże znaczenie w przypadku rozpoznawania wyrazów i zdań PJM, których wykonania nawet przez jedną osobę mogą znacznie odbiegać od siebie, zależnie od chwilowego nastroju mówcy albo kontekstu, w którym dany gest został użyty. Skalowanie sekwencji wektorów cech do jednej ustalonej długości nie jest w tym przypadku takie proste i oczywiste ponieważ w trakcie wykonywania gestu prędkości dłoni nie są jednostajne. Zazwyczaj dłoń przyśpiesza w początkowej fazie ruchu w danym kierunku i zwalnia nieznacznie tuż przed gwałtowną zmianą kierunku. Opisano także równoległe ukryte modele Markowa. Przesłanką do ich zastosowania w rozpoznawaniu języka miganego może być fakt, że w większości znaków migowych uczestniczą dwie dłonie, których kształty i pozycje, zwłaszcza w przypadku spontanicznego wykonania gestu, niekoniecznie muszą zmieniać się synchronicznie. Ponadto nawet jeśli ruch wykonywany jest jedną dłonią, to poszczególne wartości cech mogą zmieniać się równocześnie, np. dłoń zmienia swoją konfigurację w trakcie wykonywania ruchu. W przypadku sygnału mowy wyróżnia się tzw. fonemy czyli najmniejsze, niepodzielne jednostki różnicujące dane słowo. Niektórzy badacze wskazują odpowiedniki fonemów dla przekazu migowego, nazywając je cheremami. Zasadnicza różnica pomiędzy strukturą sygnału mowy i przekazu w języku miganym polega na tym, że o ile fonemy mogą zmieniać się tylko sekwencyjnie, to zmiany cheremów mogą także zachodzić równolegle. Można oczywiście modelować procesy zachodzące równolegle z wykorzystaniem regularnych modeli Markowa, grupując w jednym wektorze cechy

z poszczególnych kanałów. W takim przypadku powstaje jednak ograniczenie, że zmiany w kanałach muszą zachodzić w tych samych dyskretnych chwilach czasowych. PaHMM ma więcej stopni swobody i można przypuszczać, że będzie lepiej radził sobie z rozpoznawaniem gestów wykonywanych w sposób spontaniczny przez wprawnego mówcę.

Omawiając podstawy matematyczne ukrytych modeli Markowa zwrócono szczególną uwagę na wersję algorytmu Viterbiego z przekazywaniem znaczników, która znajduje zastosowanie przy rozpoznawaniu zdań z wykorzystaniem sieci zbudowanych z ukrytych modeli Markowa odpowiadających poszczególnym wyrazom, zarówno w przypadku zwykłych jak i równoległych ukrytych modeli Markowa. W wersji tej każdy stan modelu, w każdej chwili czasowej posiada znacznik, który w następnej chwili czasowej zostaje przekazany do wszystkich połączonych dozwolonym przejściem stanów. Ze znacznikiem tym oprócz wartości prawdopodobieństwa mogą być związane pewne dodatkowe przydatne informacje, np. o sekwencji przebytych stanów i modeli.

Zwrócono także uwagę na specjalną strategię uczenia z wbudowanym podziałem zdań uczących na słowa, która znajduje zastosowanie przy rozpoznawaniu sekwencji gestów z wykorzystaniem sieci ukrytych modeli Markowa. Uczenie poszczególnych modeli odbywa się wtedy równolegle z wykorzystaniem tworzonych dynamicznie struktur odpowiadających sekwencjom ze zbioru uczącego. Zaletą metody jest to, że nie jest wymagana precyzyjna segmentacja danej sekwencji, a jedynie informacja o kolejności tworzących ją gestów. Ułatwia to przygotowywanie danych do uczenia.

W rozdziale opisano także dwa rodzaje modeli przejścia, które zastosowano w niniejszej pracy do modelowania przejść pomiędzy gestami wyrażającymi wyrazy podczas przedstawiania zdań.

Rozdział 5

Rozpoznawanie pojedynczych słów

Do badań wybrano 101 wyrazów występujących w typowych sytuacjach u lekarza i na poczcie (patrz dodatek E). Sekwencje obrazów 320*240 pikseli otrzymane w układzie stereowizyjnym z kamerami kolorowymi rejestrowano z częstotliwością 25 klatek/s. Przygotowano trzy zbiory danych, liczące po 20 wykonań każdego wyrazu przez dwie osoby (A i B) w dobrych warunkach oświetlenia, oraz przez osobę B w niekorzystnych warunkach oświetlenia. Osoba B jest lektorką PJM, osoba A, autor niniejszej pracy, wyuczyła się języka migowego na potrzeby tych badań. Zbiory nazwane, odpowiednio, A, B, B' podzielono na równe, rozłączne podzbiory, przeznaczone do uczenia (A_{tr}, B_{tr}, B'_{tr}) i do testowania (A_{te}, B_{te}, B'_{te}) ukrytych modeli Markowa. Na zbiorach A i B wykonano również czterokrotną walidację skrośną, wyodrębniając w każdym cztery rozłączne, równoliczne podzbiory, z których trzy wykorzystywano do uczenia, a czwarty do testowania.

5.1 Wybór wektora cech

Rozważono 32 warianty wektora cech, scharakteryzowane w podrozdziale 3.4. Tab. 5.1 zawiera wyniki eksperymentów dotyczących zbiorów testowych. Przyjęte dodatkowe oznaczenia mają następującą interpretację:

- A_{V4}, B_{V4} - dotyczy uśrednienia wyników czterokrotnej walidacji skrośnej dla zbioru A, B,
- A_{te}, B_{te}, B'_{te} - dotyczy wyników rozpoznawania zbiorów testowych przez modele wyuczone na podstawie odpowiadających zbiorów uczących,
- A_{tr}/B_{te}, B_{tr}/A_{te} - dotyczy testowania modeli wyuczonych na gestach innej osoby,
- AB_{tr}/A_{te}, AB_{tr}/B_{te} - dotyczy testowania modeli wyuczonych na gestach obu osób, wtedy do uczenia wzięto po pięć pierwszych elementów z A_{tr} i B_{tr} .

Modele odpowiadające poszczególnym wyrazom miały po 4 stany, przy czym stany 1 i 4 były nieemitujące (rys. 4.6). Stany nieemitujące (początkowy i końcowy)

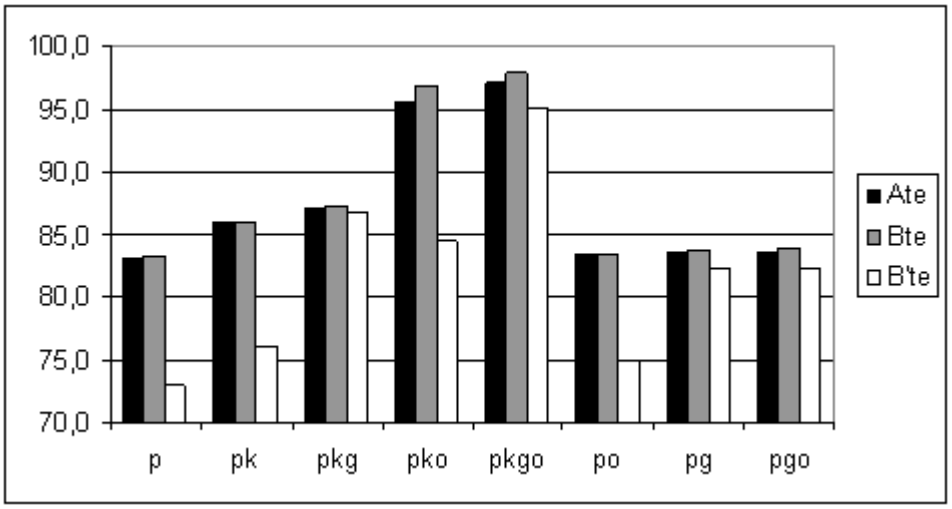
Tab. 5.1. *Skuteczność rozpoznawania na zbiorach testowych* [%] *(Oznaczenia wektorów cech są zgodne z przyjętymi w tab. 3.7)*

wektor cech	opis	A_{V4}	A_{te}	B_{V4}	B_{te}	B'_{te}	A_{tr}/B_{te}	AB_{tr}/B_{te}	B_{tr}/A_{te}	AB_{tr}/A_{te}
1	położenie	90.1	83.1	90.7	83.2	73.0	81.8	83.9	82.7	83.6
2		91.6	84.7	92.2	84.5	74.6	82.9	85.0	84.7	85.3
3		91.8	86.0	92.4	86.1	76.0	85.3	86.6	85.1	86.5
4	położenie	91.9	84.5	92.5	84.4	74.5	83.3	84.8	84.9	85.1
5	+	92.4	86.5	93.0	86.9	76.4	85.5	87.0	85.6	86.9
6	kształt	93.0	87.2	93.6	87.2	77.2	85.8	87.7	86.0	87.3
7		92.9	85.4	93.5	84.9	75.4	82.8	85.3	83.3	85.7
8		93.2	87.5	93.8	87.6	77.4	85.5	88.1	86.6	87.6
9		92.9	86.8	93.5	87.6	86.7	84.8	87.8	85.5	87.2
10	położenie	93.2	85.6	93.8	85.9	85.5	84.6	86.6	84.8	86.2
11	+	92.2	87.1	92.8	87.4	85.9	85.2	87.7	86.2	87.6
12	kształt	93.1	87.5	93.7	87.5	86.9	85.0	87.8	86.7	87.5
13	+	93.4	87.5	94.0	87.6	87.8	86.2	88.2	87.0	87.9
14	głębia	93.3	87.3	93.9	87.3	86.9	86.3	88.1	86.3	87.7
15		93.5	87.2	94.1	87.1	87.0	85.4	87.3	86.2	86.7
16		93.0	93.9	92.6	95.2	83.2	92.1	94.7	94.5	94.7
17	położenie	93.3	96.7	92.8	96.4	84.5	94.8	96.1	93.3	94.9
18	+	92.7	93.9	93.8	95.2	83.8	92.3	93.3	93.5	93.3
19	kształt	94.2	96.3	93.6	98.4	85.4	94.9	96.1	94.3	94.9
20	+	94.2	95.8	94.3	97.4	84.8	95.1	95.7	95.6	95.4
21	orientacja	94.7	94.9	94.8	95.5	83.9	92.5	94.6	92.3	93.9
22		95.3	97.6	94.5	99.4	85.7	94.5	97.7	96.7	96.0
23	położenie	94.3	97.4	94.3	98.4	95.1	94.1	97.4	94.7	95.5
24	+	94.2	93.9	94.8	95.8	93.7	93.5	95.2	92.9	95.4
25	kształt	94.2	95.8	93.9	97.4	94.6	93.2	96.3	95.4	96.3
26	+	95.0	98.5	94.8	97.6	95.5	94.5	96.7	95.3	95.5
27	głębia	94.0	98.6	94.1	99.3	97.0	95.5	97.7	95.7	97.3
28	+	94.7	97.4	95.7	98.6	95.5	95.5	97.5	95.7	96.4
29	orientacja	94.7	97.4	94.9	97.4	94.5	93.6	96.2	95.3	95.3
30	położenie + orientacja	90.2	83.4	90.7	83.4	75.0	82.1	83.4	82.3	83.3
31	położenie + głębia	90.3	83.6	90.8	83.7	82.3	82.4	83.6	82.4	83.4
32 32 32	położenie + głębia + orientacja	90.2	83.6	90.8	83.8	82.3	82.5	83.6	82.4	83.6

są pomocne podczas budowania sieci modeli wykorzystywanych w rozpoznawaniu zdań (zob. podrozdział 4.3). Funkcja gęstości prawdopodobieństwa obserwacji dla każdego ze stanów emitujących miała postać sumy dwóch rozkładów Gaussa.

Rozkład wielomodalny daje możliwość uwzględnienia różnych reprezentacji tego samego gestu. Suma dwóch Gaussianów odpowiada dwóm wariantom wykonania. Nawet ta sama osoba może np. raz rozpocząć wykonanie słowa na wysokości biodra, podczas gdy innym razem na wysokości brzucha. Jako narzędzie obliczeniowe wykorzystano pakiet HTK [81].

Na rys. 5.1 przedstawiono średnie wartości skuteczności rozpoznawania w poszczególnych grupach wektorów cech dla zbiorów A_{te}, B_{te} i B'_{te}.

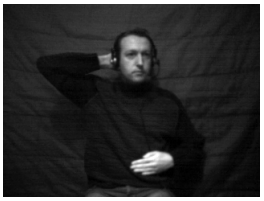
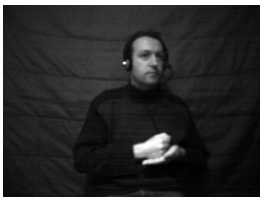
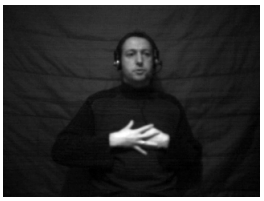


Rys. 5.1. *Średnie skuteczności rozpoznawania wyrazów w poszczególnych grupach wektorów cech dla zbiorów A_{te}, B_{te}, B'_{te}; p - położenie, k - kształt, g - głębia, o - orientacja.*

Analizując bliżej rezultaty rozpoznawania stwierdzono, że uwzględnienie w wektorze cech elementów związanych z kształtem dłoni poprawiło rozpoznawanie tych wyrazów, w wykonywaniu których dłonie przyjmują podobne pozycje, lecz różne kształty. W tab. 5.2 pokazano przykłady pomyłek popełnianych podczas rozpoznawania takich wyrazów z wykorzystaniem wektora cech zawierającego tylko informację o położeniu.

Informacja o położeniu przestrzennym dłoni jest częściowo zawarta w jej powierzchni. Przy zbliżaniu dłoni do kamery powierzchnia odpowiadającego jej obiektu w obrazie wzrasta. Jednak gdy w wyniku niedokładności segmentacji otrzymywane kształty dłoni zostaną zdeformowane, np. wskutek dołączenia do obszaru dłoni mylnie zakwalifikowanych pikseli tła, zmiany powierzchni związane ze składową ruchu równoległą do osi optycznej kamery mogą być niezauważalne. Wtedy informacja o głębi stanowi uzupełnienie, pozwalające na dokonanie poprawnej klasyfikacji gestu. Na rys. 5.2 przedstawiono początkową i końcową fazę gestu skierowanie nagranego w trudniejszych warunkach, przy jasnym tle i silnym oświetleniu sztucznym, mającym charakter kierunkowy. Środki ciężkości obiektów odpowiadających dłoniom na obrazach binarnych 5.2b) i 5.2e) są nieznacznie przesunięte, co świadczy o istnieniu składowej ruchu w płaszczyźnie równoległej do płaszczyzny obrazu. Jednak powierzchnie i kształty tych obiektów są podobne i trudno na tej podstawie wnioskować o składowej ruchu równoległej do osi optycznej kamery. Widać jednak, że obszar

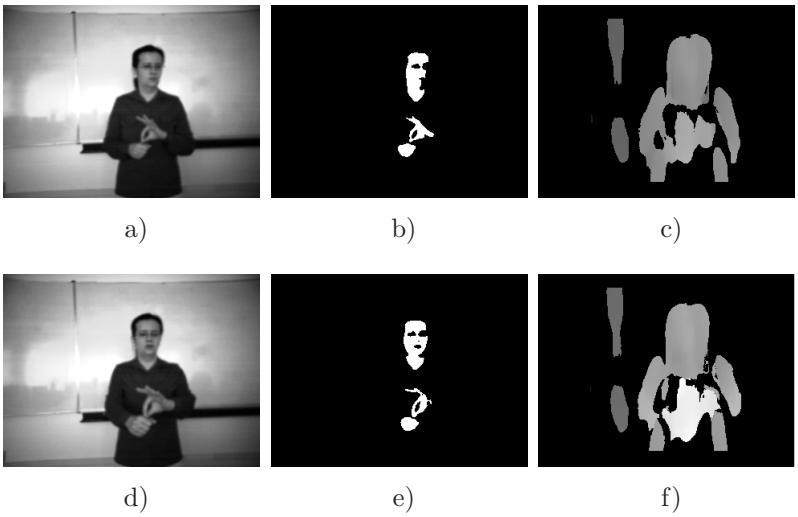
Tab. 5.2. *Wybrane pomyłki w trakcie rozpoznawania z wykorzystaniem wektora cech zawierającego tylko informację o położeniu dłoni*

wyraz rozpoznany	wyraz rozpoznawany
 *pogotowie*	 *opony mózgowe*
 *przyjmować*	 *otrzymać*
 *list*	 *prześwietlenie*

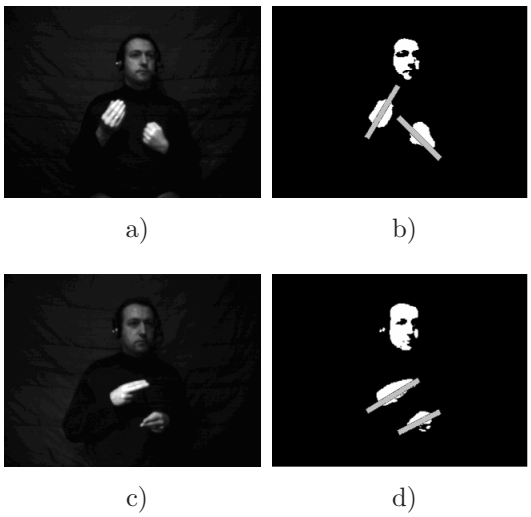
odpowiadający dłoniom na obrazie mapy dysparycji 5.2f) jest wyraźnie jaśniejszy, niż na obrazie 5.2c). Oznacza to, że w końcowej fazie gestu dłonie znajdowały się bliżej kamery, a zatem wystąpił ruch w kierunku od osoby wykonującej gest. Podobne sytuacje spowodowały wyraźnie gorszą skuteczność rozpoznania w zbiorze B', jeśli wektor cech nie uwzględniał informacji o głębi. Dodanie tej informacji dało jednocześnie znacznie wyraźniejszy skutek niż w przypadkach, kiedy lepsze warunki oświetlenia pozwalały na dokładniejszą detekcję obszaru dłoni (zob. np. kolumny dotyczące zbiorów A_{te}, B_{te}).

Wpływ dodania do wektora cech orientacji ψ pokazano na rys. 5.3. Kształty i pozycje binarnych obiektów odpowiadających dłoniom w obrazach 5.3b) i 5.3d) są zbliżone. W takim przypadku dodanie do wektora cech orientacji osi głównej dla obu dłoni ułatwia rozpoznanie tych dwóch gestów.

W tab. 5.3 zestawiono wyrazy, które były najczęściej rozpoznawane niepoprawnie z wykorzystaniem wektora cech (27), dla którego otrzymano najlepszy wynik i dla wektora (13) zawierającego te same elementy składowe z wyjątkiem orientacji osi głównej.



Rys. 5.2. *Obrazy zarejestrowane podczas rozpoznawania gestu skierowanie nagranego w niekorzystnych warunkach oświetlenia: a), b), c) - kolejno: obraz monochromatyczny, binarny i mapa dysparycji dla początkowej fazy gestów, d), e), f) - analogiczne obrazy dla końcowej fazy gestu.*



Rys. 5.3. *Wpływ dodania do wektora cech orientacji: a) obraz wejściowy dla gestu rachunek, b) obraz binarny otrzymany na podstawie obrazu a), c) obraz wejściowy dla gestu chory, d) obraz binarny otrzymany na podstawie obrazu c).*

Podobnie jak w przypadku języka mówionego, każda osoba wykonuje gest w specyficzny dla siebie sposób. Rozważano więc przypadek wykorzystania modeli wyuczonych na gestach innej osoby niż rozpoznawane. Zaobserwowano pogorszenie otrzymanych rezultatów. Metoda nie jest niezależna od użytkownika i baza danych gestów wykorzystanych do uczenia powinna zawierać wykonania poszczególnych wyrazów przez więcej osób.

Tab. 5.3. *Najczęściej mylone wyrazy dla wektorów cech 13 i 27*

wyraz	liczba pomyłek wektor 13	liczba pomyłek wektor 27
dzisiaj	10	5
boleć	9	2
żołądek	9	7
na	8	3
o	7	2
rachunek	7	2
wysyłać	7	3
inny	6	2
termometr	5	0
i	3	0
lekarstwo	3	0
opony	3	0
pisać	3	0
pogotowie	3	0
analiza	2	0
angina	2	0
leżeć	2	0
łóżko	2	0

Przeanalizowano skuteczność rozpoznawania na podstawie różnych wektorów cech. Globalne rezultaty przedstawione w tab. 5.1 ukazują różnice. Bliższe spojrzenie na skuteczność rozpoznawania poszczególnych słów pozwala stwierdzić, że pewne wektory cech okazują się korzystniejsze w rozpoznawaniu niektórych słów, inne zaś dominują w pozostałych przypadkach. Świadczą o tym zestawienia w tab. 5.4 i 5.5. Pierwsza tabela odnosi się do wektorów cech nieuwzględniających orientacji dłoni, druga do wszystkich 32 wektorów cech. Porównując kolumny (e), (f), (g) oraz (i) w tab. 5.4 widzimy, że najlepszy rezultat - kolumna (i) - jest wyraźnie gorszy od wyniku, jaki uzyskanoby biorąc wektor cech najlepszy w rozpoznawaniu konkretnych słów, co oczywiście nie jest możliwe w praktyce. Kolumny (b) - (h) pozwalają ocenić relacje między hipotetycznymi wynikami w zależności od możliwości wyboru: spośród wszystkich wektorów cech - kolumny (b), (e), spośród wektorów zawierających informację o położeniu i kształcie - kolumny (c), (f) - oraz spośród wektorów uwzględniających wymienione elementy wraz informacją o głębi. Z kolei kolumna (h), dotycząca wektora cech uwzględniającego tylko informację o położeniu, porównana z kolumnami (b), (c), (d) wskazuje, że najuboższy wektor cech nie musi być zawsze najgorszy.

Przedstawione obserwacje uzasadniają celowość przebadania skuteczności rozpoznawania z wykorzystaniem kombinacji klasyfikatorów opartych na różnych wektorach cech. Zastosowano metody znane z literatury (zob. np. [8, 15, 46]).

Tab. 5.4 pokazuje wyniki dla głosowania z większością zwykłą - kolumna (j), bezwzględną - kolumna (k), techniką Bordy - kolumna (l) oraz uśrednień Bayesa -kolumna (m) i całki rozmytej - kolumna (n). W odniesieniu do dwóch ostatnich metod konieczna była normalizacja wyjść modeli odpowiadających jednakowym wektorom cech do przedziału [0, 1]. Wyniki potwierdzają atrakcyjność metod głosowania i całki rozmytej, dla których otrzymano rezultat o około 4% lepszy niż najlepszy z uzyskanych za pomocą najlepszego klasyfikatora.

Tab. 5.4. *Średnie skuteczności rozpoznawania w zbiorze testowym przez różne klasyfikatory uwzględniające cechy 1-15,* [%]

zbiór	min 1-15	min 2-8	min 9-15	max 1-15	max 2-8	max 9-15	klas. 1	naj.	głos.	50% +1	Borda	Bayes	cał. roz.
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)
A_{te}	67.0	75.0	78.0	96.0	94.0	94.0	83.0	88.0	92.5	89.5	89.7	90.7	92.1
B_{te}	67.0	71.7	76.6	96.9	95.0	94.8	83.2	88.5	92.5	89.1	89.2	90.5	92.2
B'_{te}	50.4	55.0	73.1	97.9	91.6	95.4	73.0	87.8	91.5	87.0	63.5	63.2	91.2

(b) - (d): *klasyfikator wykorzystujący dla każdego słowa najmniej korzystny wektor cech spośród wymienionych*,
(e) - (g): *klasyfikator wykorzystujący dla każdego słowa najkorzystniejszy wektor cech spośród wymienionych*,
(h): *klasyfikator wykorzystujący pierwszy wektor cech*,
(i): *klasyfikator łącznie najlepszy*,
(j): *kombinacja klasyfikatorów metodą głosowania z większością zwykłą*,
(k): *kombinacja klasyfikatorów metodą głosowania z większością bezwzględną*,
(l): *kombinacja klasyfikatorów metodą pozycyjną Bordy*,
(m): *kombinacja klasyfikatorów ważoną metodą Bayesa (przyjęto jednakowe wagi dla każdego klasyfikatora)*,
(n): *kombinacja klasyfikatorów metodą całki rozmytej (dla każdego klasyfikatora przyjęto jednakowy stopień ważności 0.75)*

Tab. 5.4 odnosi się do klasyfikatorów bazujących na wektorach cech 1-15, tzn. nieuwzględniających orientacji dłoni. Skutek uwzględnienia także tej cechy ukazuje tab. 5.5. Tutaj bardzo wysoki wynik globalnie najlepszego klasyfikatora okazał się tylko nieznacznie gorszy od hipotetycznego rezultatu, jaki otrzymanoby dobierając klasyfikator indywidualnie do słowa. Fuzja klasyfikatorów generalnie pogorszyła rezultat rozpoznawania, zapewne wskutek uwzględnienia zbyt wielu klasyfikatorów słabych. W przypadku głosowania i całki rozmytej pogorszenie to okazało się jednak najmniejsze. Staranniejszy dobór kombinowanych klasyfikatorów (patrz tab. 5.6, 5.7) poprawił końcowy rezultat, chociaż w porównaniu z pojedynczym, globalnie najlepszym klasyfikatorem korzyść okazała się mało znacząca, czego można było oczekiwać.

Tab. 5.5. *Średnie skuteczności rozpoznawania w zbiorze testowym przez różne klasyfikatory uwzględniające cechy 1-29,* [%]

zbiór	min 1-29	max 1-29	klas. 1	naj.	głos.	50% +1	Borda	Bayes	cał. roz.
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
A_{te}	63.7	100	83.0	98.6	95.4	93.7	98.9	93.5	95.1
B_{te}	67.0	100	83.2	99.3	99.2	95.3	99.2	93.2	95.1
B'_{te}	50.4	99.7	73.0	97.0	95.4	89.2	71.3	71.3	94.8

(b): *klasyfikator wykorzystujący dla każdego słowa najmniej korzystny wektor cech spośród wymienionych*,
(c): *klasyfikator wykorzystujący dla każdego słowa najkorzystniejszy wektor cech spośród wymienionych*,
(d): *klasyfikator wykorzystujący pierwszy wektor cech*,
(e): *klasyfikator łącznie najlepszy*,
(f): *kombinacja klasyfikatorów metodą głosowania z większością zwykłą*,
(g): *kombinacja klasyfikatorów metodą głosowania z większością bezwzględną*,
(h): *kombinacja klasyfikatorów metodą pozycyjną Bordy*,
(i): *kombinacja klasyfikatorów ważoną metodą Bayesa (przyjęto jednakowe wagi dla każdego klasyfikatora)*,
(j): *kombinacja klasyfikatorów metodą całki rozmytej (dla każdego klasyfikatora przyjęto jednakowy stopień ważności 0.75)*

Tab. 5.6. *Średnie skuteczności rozpoznawania w zbiorze testowym przez różne klasyfikatory uwzględniające cechy 16-29,* [%]

zbiór	min 16-29	max 16-29	klas. 1	naj.	głos.	50% +1	Borda	Bayes	cał. roz.
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
A_{te}	85.5	100.0	93.9	98.6	97.6	94.8	99.1	95.2	97.1
B_{te}	86.4	100.0	95.2	99.3	100.0	99.9	99.5	94.9	97.1
B'_{te}	63.0	99.7	83.2	97.0	97.6	94.8	75.1	75.1	96.9

Tab. 5.7. *Średnie skuteczności rozpoznawania w zbiorze testowym przez różne klasyfikatory uwzględniające cechy 17, 19, 20, 22, 23, 25, 26, 27, 28, 29,* [%]

zbiór	min	max	klas. 1	naj.	głos.	50% +1	Borda	Bayes	cał. roz.
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
A_{te}	90.8	100.0	96.7	98.6	97.4	95.3	99.4	97.3	97.6
B_{te}	92.6	100.0	96.4	99.4	99.9	99.5	99.6	96.1	97.6
B'_{te}	70.5	99.6	84.5	97.0	97.4	95.3	76.3	77.0	96.5

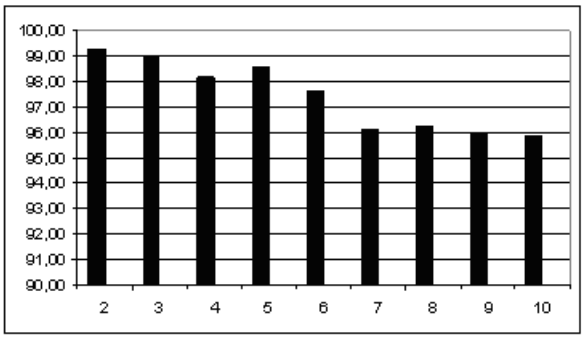
5.2 Dyskusja liczby stanów

Dla każdego z rozważanych wariantów wektora cech eksperymenty z rozpoznawaniem wyrazów powtórzono dla modeli Markowa mających od 2 do 10 stanów emitujących, przy czym dla danego testu wszystkie modele miały taką samą liczbę

stanów. W tab. 5.8 i na rys. 5.4 przedstawiono przykładowe wyniki otrzymane dla wektora cech 27 i dla osoby A. Najlepszy rezultat 98.6% otrzymano dla modeli

Tab. 5.8. *Skuteczność rozpoznawania wyrazów w zależności od liczby stanów w ukrytych modelach Markowa (w danym eksperymencie liczba stanów dla wszystkich wyrazów jest jednakowa),* [%]

-	-	liczba stanów emitujących w ukrytych modelach Markowa								
zbiór	liczność	2	3	4	5	6	7	8	9	10
A_{tr}	1010	100.0	100.0	100.0	100.0	100.0	100.0	99.9	100.0	100.0
A_{te}	1010	98.6	97.9	96.3	97.1	95.2	92.3	92.6	92.0	91.7
razem	2020	99.3	99.0	98.2	98.6	97.6	96.1	96.2	96.0	95.8



Rys. 5.4. *Skuteczność rozpoznania wyrazów ze zbioru A_{te} w zależności od liczby stanów emitujących w ukrytych modelach Markowa (w danym eksperymencie liczba stanów dla wszystkich wyrazów jest jednakowa).*

mających po 2 stany. Jednak kolejne rezultaty, dla 3, 4 i 5 stanów są tylko nieznacznie gorsze. Dla innych rozpatrywanych wariantów wektora cech najlepsze wyniki wystąpiły, gdy liczba stanów była równa 2, 3, 4 lub 5. We wszystkich przypadkach obserwowano gwałtowne pogarszanie się wyników, gdy liczba stanów była większa od 5.

W tab. 5.10 przedstawiono rezultaty otrzymane dla przypadku, gdy liczba stanów dla poszczególnych wyrazów nie była jednakowa. Do wyznaczenia liczby stanów dla danego wyrazu posłużono się wynikami eksperymentów dla liczby stanów zmieniającej się od 2 do 10, w których wszystkie modele miały jednakową liczbę stanów. Wyznaczono średnie prawdopodobieństwa wygenerowania danego wyrazu w odpowiadających mu modelach o różnej liczbie stanów i wybrano ten model, dla którego prawdopodobieństwo to było największe. Przyporządkowanie liczby stanów do modeli odpowiadających poszczególnym wyrazom przedstawiono w tab. 5.9.

Wyniki z tab. 5.10 dotyczą wektora cech 27. Analogiczne testy wykonano dla wszystkich rozważanych wariantów wektora cech otrzymując za każdym razem nieznacznie lepszy wynik dla przypadku, gdy modele odpowiadające poszczególnym wyrazom miały zróżnicowaną liczbę stanów.

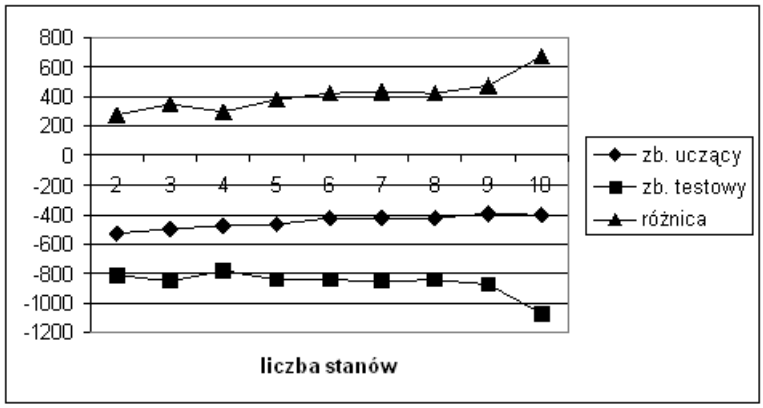
Tab. 5.9. *Przyporządkowanie liczby stanów do modeli poszczególnych wyrazów*

liczba stanów	wyrazy
5	być, katar, okulary, szpital, zastrzyk, zwolnienie
4	grypa, natychmiast, o, otrzymać, palić, poczta, pójść, pokazać, rachunek, się, wata
3	analiza, aparat, do, dokładny, głowa, i, ja, kosztować, leżeć, nie, opony, paczka, pogotowie, polecony, położyć, potrzebny, przeziębiony, recepta, słuch, słyszeć, tabletka, wykonać, wysyłać, zęby
2	pozostałe wyrazy

Tab. 5.10. *Rozpoznawanie wyrazów z modelami Markowa o zróżnicowanej liczbie (w danym eksperymencie liczba stanów dla poszczególnych wyrazów nie jest jednakowa),* [%]

zbiór	liczność	taka sama liczba stanów dla poszczególnych wyrazów = 2	liczba stanów zróżnicowana dla poszczególnych wyrazów
uczący	1010	100.0	100.0
testowy	1010	98.6	99.1
razem	2020	99.3	99.6

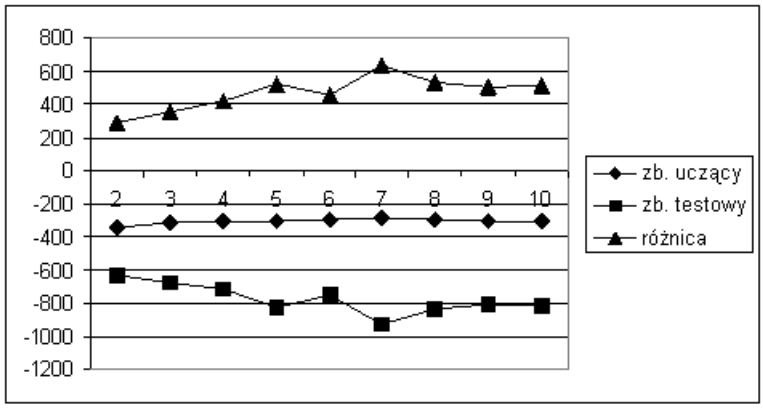
Badano zależność średnich wartości logarytmów prawdopodobieństw wygenerowania danego wyrazu w odpowiadającym mu modelu od liczby stanów tego modelu, osobno na zbiorze uczącym i testowym. Na rys. 5.5 przedstawiono przebiegi otrzymane dla wyrazu *analiza* i wektora cech 27.



Rys. 5.5. *Zależności średnich wartości logarytmów prawdopodobieństw wygenerowania wyrazu analiza w odpowiadającym mu modelu od liczby stanów tego modelu, wyznaczone na podstawie zbiorów uczącego i testowego oraz różnica dla obu zbiorów.*

Z porównania wykresów wynika, że dla dużej liczby stanów (> 5) model dobrze dopasowuje się do danych użytych w uczeniu (większe średnie wartości prawdopodobieństw), ale traci swe właściwości uogólniające (większa różnica średnich wartości

prawdopodobieństw otrzymanych na zbiorach uczącym i testowym). Większość z przebiegów dla innych słów miała podobny charakter. Na rys. 5.6 przedstawiono przebiegi uśrednione dla wszystkich wyrazów. Wyniki eksperymentów wskazują, że do rozpoznawania wybranych słów i zdań PJM powinno się raczej projektować modele Markowa o mniejszej liczbie stanów.



Rys. 5.6. *Zależności średnich wartości logarytmów prawdopodobieństwa wygenerowania wyrazu w odpowiadającym mu modelu od liczby stanów tego modelu, wyznaczone na podstawie zbiorów uczącego i testowego oraz różnica dla obu zbiorów.*

5.3 Podsumowanie

W rozdziale przedstawiono wyniki rozpoznawania podzbioru 101 wyrazów PJM występujących w typowych sytuacjach: u lekarza i na poczcie. Gesty wykonywane były przez lektorkę PJM oraz przez autora, który wyuczył się ich na użytek niniejszej pracy. Łącznie wykorzystano 6060 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przebadano 32 różne warianty wektora cech (tab. 3.7), uwzględniając przypadki, gdy uczenie i testowanie odbywało się na gestach tej samej osoby, uczenie i testowanie odbywało się na gestach różnych osób oraz gdy zbiór uczący został zbudowany z wykonań poszczególnych gestów przez obie osoby. Modele odpowiadające poszczególnym wyrazom miały po 4 stany (rys. 4.2), przy czym stany 1 i 4 były nieemitujące, zaś funkcja gęstości prawdopodobieństwa obserwacji dla każdego ze stanów emitujących miała postać sumy dwóch rozkładów Gaussa. Przedstawiono rezultaty otrzymane dla zbiorów testowych oraz wyniki powstałe wskutek uśrednienia wyników czterokrotnej walidacji skrośnej (tab. 5.1). Najlepszy rezultat 99.3% otrzymano dla przypadku, gdy wektor cech zawierał informację o położeniu, informację o kształcie wyrażoną za pomocą pola powierzchni i współczynnika zwartości, informację 3D i informację o orientacji dłoni. Przeprowadzono popartą przykładami dyskusję na temat wpływu poszczególnych elementów wektora cech na skuteczność rozpoznawania. Rozważano też przypadek wykorzystania modeli wyuczonych na gestach innej osoby niż rozpoznawane. Zaobserwowano wyraźne pogorszenie otrzymanych rezultatów, co wskazuje na to, że metoda

nie jest niezależna od użytkownika. Bliższa analiza pokazała, że pewne wektory cech okazują się korzystniejsze w rozpoznawaniu niektórych słów, inne zaś dominują w pozostałych przypadkach. Dlatego przebadano skuteczność rozpoznawania z wykorzystaniem kombinacji klasyfikatorów opartych na różnych wektorach cech. Zastosowano metodę głosowania z większością zwykłą i z większością bezwzględną, metodę pozycyjną Bordy, metodę uśrednień Bayesa oraz całkę rozmytą. W przypadku kombinacji pierwszych 15 wariantów wektora cech (tab 3.7) najlepsze wyniki otrzymano dla całki rozmytej. W przypadku, gdy uwzględniono wszystkie warianty wektora cech, fuzja klasyfikatorów nie dała zadowalających rezultatów, gdyż wynik globalnie najlepszy był tylko nieznacznie gorszy od hipotetycznego rezultatu, jaki otrzymanoby dobierając wektor cech indywidualnie dla każdego słowa.

Przedyskutowano wpływ liczby stanów ukrytych modeli Markowa na skuteczność rozpoznawania słów. Jak można było oczekiwać modele z większą liczbą stanów dobrze dopasowały się do zbiorów uczących, wykazując jednak gorsze właściwości uogólniania, co uwidoczniało się wzrostem liczby błędów rozpoznawania na zbiorach testowych. Stwierdzono, że najlepsze rezultaty otrzymuje się różnicując liczbę stanów w modelach słów, z ograniczeniem, że nie powinna być większa od pięciu. Wynik ten okazał się jednak tylko nieznacznie lepszy od rezultatu otrzymanego z modelami dwustanowymi.

Rozdział 6

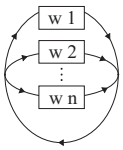
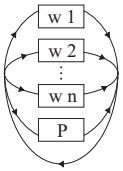
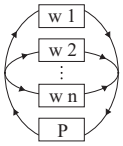
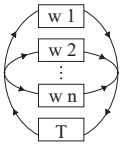
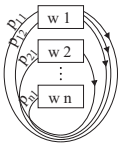
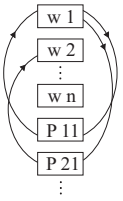
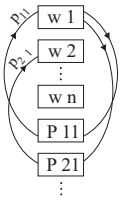
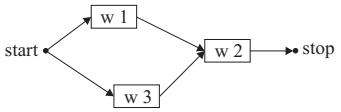
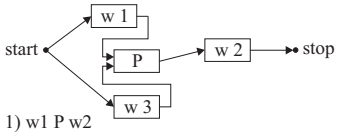
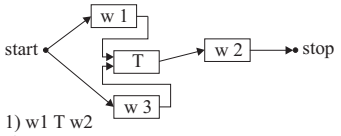
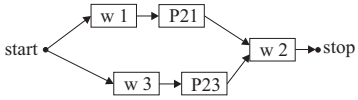
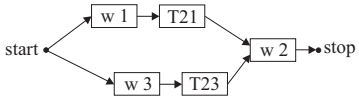
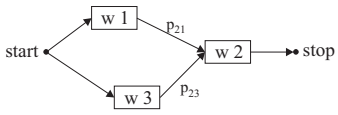
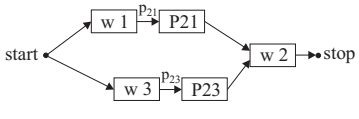
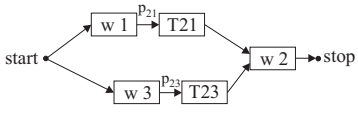
Rozpoznawanie zdań

Rozważano 35 zdań przedstawionych w dodatku E. Zdania wykorzystują wyrazy, których rozpoznawanie omówiono w rozdziale 5 i odnoszą się do typowych sytuacji u lekarza i na poczcie. Podobnie jak w przypadku wyrazów, przygotowano bazę danych po 20 wykonań każdego zdania przez osobę A, która wyuczyła się języka gestów na potrzeby tej pracy oraz przez osobę B - lektorkę PJM.

6.1 Wybór struktury układu rozpoznającego

Rozpoznawanie zdań odbywało się przy założeniu, że jeden model Markowa odpowiada pojedynczemu wyrazowi, całe zdanie rozpoznawane jest zaś z wykorzystaniem jednej ze struktur przedstawionych w tab. 6.1. Zamieszczone w tabeli schematy mają charakter poglądowy i przedstawiają tylko wybrane fragmenty poszczególnych konfiguracji. W konfiguracjach c_1 .. c_7 po danym wyrazie (modele wyrazów oznaczono literą w) może wystąpić dowolny wyraz natomiast w konfiguracjach c_8 .. c_{15} dopuszczalne są tylko sekwencje wyrazów występujące w rozpatrywanych zdaniach, co znacznie ułatwia rozpoznawanie. W konfiguracjach c_2, c_3, c_4, c_6, c_7, c_9, c_{10}, c_{11}, c_{12}, c_{14} i c_{15} oprócz modeli odpowiadających poszczególnym wyrazom zastosowano także modele opisujące przejścia pomiędzy wyrazami, oznaczone na schematach literami P (model z rys. 4.7a) i T (model z rys. 4.7b). Modele przejścia były modelami trójstanowymi, przy czym stan pierwszy i trzeci były nieemitujące. Funkcja gęstości prawdopodobieństwa obserwacji miała postać sumy dwóch gaussianów. W konfiguracjach c_4, c_{10}, c_{12}, c_{15} zastosowano model przejścia typu 'Tee'. W modelu tym istnieje dodatkowe przejście ze stanu pierwszego do stanu ostatniego (patrz rys. 4.7). W konfiguracjach c_2, c_3, c_4, c_9, c_{10} zastosowano jeden model przejścia natomiast w konfiguracjach c_6, c_7, c_{11}, c_{12}, c_{14}, c_{15} zastosowano inny model przejścia dla każdej pary następujących po sobie wyrazów. W konfiguracjach c_5, c_7, c_{13}, c_{14} i c_{15} uwzględniono prawdopodobieństwo bigram oznaczone małą literą p między blokami.

Tab. 6.1. *Konfiguracje HMM wykorzystane przy rozpoznawaniu zdań*

konfiguracja HMM	oznaczenie	konfiguracja HMM	oznaczenie
	c_1		c_2
	c_3		c_4
	c_5		c_6
	c_7	 1) w1 w2 2) w3 w2	c_8
 1) w1 P w2 2) w3 P w2	c_9	 1) w1 T w2 2) w3 T w2	c_{10}
 1) w1 P21 w2 2) w3 P23 w2	c_{11}	 1) w1 T21 w2 2) w3 T23 w2	c_{12}
 1) w1 w2 2) w3 w2	c_{13}	 1) w1 P21 w2 2) w3 P23 w2	c_{14}
 1) w1 T21 w2 2) w3 T23 w2	c_{15}		

Eksperymenty przeprowadzono dla wszystkich rozważanych wariantów wektora cech. W tab. 6.2 przedstawiono wyniki otrzymane dla wektora cech 15 i wektora 27. Drugi wektor uwzględnia dodatkowo orientację dłoni. Dane podzielono na dwa równe zbiory: uczący i testowy, zawierające po 10 wykonań każdego z 35 zdań.

Tab. 6.2. *Porównanie skuteczności rozpoznawania* [%] *zdań PJM wykonanych przez osobę A dla różnych konfiguracji HMM*

wektor cech 15			
konfiguracja	zbiór uczący (350 wykonań)	zbiór testowy (350 wykonań)	razem (700 wykonań)
c1	0.0	1.7	0.8
c2	0.3	0.3	0.3
c3	9.1	8.3	8.7
c4	0.0	1.7	0.9
c5	79.1	77.7	78.4
c6	2.9	1.4	2.1
c7	81.1	76.8	79.0
c8	96.3	90.3	93.3
c9	95.4	88.0	91.7
c10	96.3	90.3	93.3
c11	95.4	88.0	91.7
c12	96.3	90.0	93.2
c13	96.3	90.3	93.3
c14	95.4	88.0	91.7
c15	96.3	90.0	93.2
wektor cech 27			
konfiguracja	zbiór uczący (350 wykonań)	zbiór testowy (350 wykonań)	razem (700 wykonań)
c1	0.6	0.0	0.3
c2	0.9	0.6	0.8
c3	11.1	10.0	10.6
c4	1.1	0.9	1.0
c5	80.9	79.1	80.0
c6	4.0	2.6	3.3
c7	83.1	79.7	81.4
c8	98.0	97.1	97.6
c9	97.1	95.1	96.1
c10	98.0	97.1	97.6
c11	97.1	95.1	96.1
c12	98.0	96.6	97.3
c13	98.0	97.1	97.6
c14	97.1	95.1	96.1
c15	98.0	96.6	97.3

Dla konfiguracji: c_1, c_2, c_3, c_4 i c_6 otrzymano bardzo słabe skuteczności rozpoznawania nie przekraczające 8.71% dla wektora cech 15 i 10.6% dla wektora 29, zatem nie nadają się one do rozpoznawania zdań PJM z wykorzystaniem opisywanej metody. Wymienione konfiguracje dopuszczają także takie sekwencje wyrazów, które nie tworzą logicznych wypowiedzi. W odniesieniu do większości rozpoznawanych zdań błędy polegały na wielokrotnym powtarzaniu danego wyrazu. Nie pomogło nawet dodanie modeli przejścia w konfiguracjach: c_2, c_3, c_4 i c_6. Dopiero uwzględnienie modelu lingwistycznego bigram [56] (zobacz też podrozdział 4.3) - wykorzystującego informację o prawdopodobieństwie wystąpienia danego słowa po poprzednim (na schematach w tab. 6.1 oznaczono je przez p) - pozwoliło na uzyskanie skuteczności 78.4% (80%) dla konfiguracji c_5 i 79.0% (81.4%) dla konfiguracji c_7. Prawdopodobieństwa bigram były wyznaczane na podstawie zbioru uczącego i przyjmowały niezerowe wartości tylko dla przejść pomiędzy wyrazami, które wystąpiły w rozpatrywanych zdaniach. To zmniejszyło prawdopodobieństwo rozpoznania sekwencji wyrazów, które nie tworzą zdań i pozwoliło na osiągnięcie znacznie lepszych wyników. Konfiguracje c_8 ... c_{15} eliminują nielogiczne sekwencje wyrazów poprzez układ ukrytych modeli Markowa odpowiadający strukturom rozpoznawanych zdań. Dla konfiguracji tych otrzymano najlepsze wyniki, przy czym zauważono, że dodanie modelu przejścia, który nie jest typu 'Tee' spowodowało obniżenie skuteczności rozpoznawania z 93.3% do 91.7% (97.6% do 96.1%). Dla modelu przejścia typu 'Tee' wyniki były analogiczne jak w sytuacji, gdy w ogóle nie uwzględniono modelu przejścia. Proces uczenia ukrytych modeli Markowa składał się z dwóch etapów. Początkowo poszczególne modele uczone były z wykorzystaniem wyrazów, które nie były wykonywane w sekwencji, tylko pojedynczo. Następnie wykonano douczanie z wykorzystaniem całych zdań, tzw. *embedded training* (patrz rozdz. 4.3 i [81]). W procesie *embedded training* modele są modyfikowane tak, aby uwzględnić spowodowane procesem koartykulacji zniekształcenia wyrazów pokazywanych w sekwencjach. Jeżeli model przejścia był typu 'Tee', to część początkowa przejścia pomiędzy dwoma wyrazami została potraktowana jako końcówka pierwszego wyrazu, część końcowa przejścia została zaś dołączona jako początek drugiego wyrazu. W konsekwencji w wyuczonych modelach przejścia typu 'Tee' prawdopodobieństwo tranzycji z pierwszego stanu nieemitującego do właściwego stanu modelu było zdecydowanie mniejsze aniżeli prawdopodobieństwo przejścia od razu do trzeciego nieemitującego stanu. Zatem w procesie rozpoznawania z wykorzystaniem algorytmu Viterbiego z przekazywaniem znaczników, znaczniki przebywające optymalną dla danej obserwacji ścieżkę omijały stany emitujące w modelu typu 'Tee'. Zastosowanie oddzielnego modelu przejścia dla każdej pary występujących po sobie wyrazów nie powoduje zwiększenia skuteczności rozpoznawania, ale zwiększa rozmiar całej struktury i znacznie wydłuża czas odpowiedzi. Jako najlepszą wybrano zatem konfigurację c_8. Porównując rezultaty z tab. 6.2 można dostrzec pozytywny wpływ uwzględnienia orientacji dłoni (wektor cech 27) na skuteczność rozpoznawania. W tab. 6.3 zestawiono zdania, które nie zostały rozpoznane.

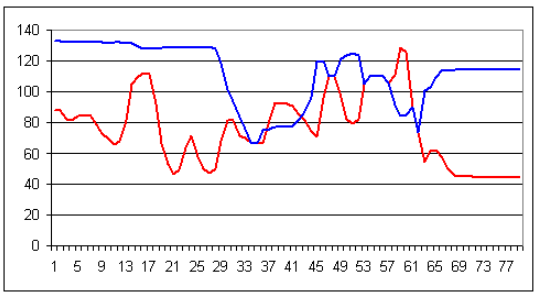
Tab. 6.3. *Błędnie rozpoznawane zdania dla konfiguracji* c_8

błędnie rozpoznane zdania	liczba pomyłek
Boli mnie gardło. [*boleć*][*mnie*][*gardło*]	20
Pan musi pójść do szpitala. [*pan*][*musieć*][*pójść*][*do*][*szpital*]	7
Nie słyszę po chorobie zapalenia opon mózgowych. [*ja*][*nie*][*słyszeć*][*po*][*chory*][*zapalenie*][*opony*]	6
Boli mnie głowa. [*boleć*][*mnie*][*głowa*]	4
Boli mnie żołądek. [*boleć*][*mnie*][*żołądek*]	4
Pan jest chory na grypę i musi leżeć w łóżku. [*pan*][*być*][*chory*][*na*][*grypa*][*i*][*musieć*][*leżeć*][*w*][*łóżko*]	4
Chcę otrzymać paczkę. [*chcieć*][*otrzymać*][*paczka*]	1
Chcę wysłać paczkę. [*chcieć*][*wysyłać*][*paczka*]	1

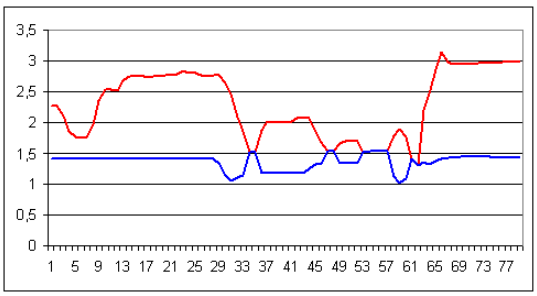
Na rys. 6.1 przedstawiono przykładowe przebiegi cech dla poprawnie rozpoznanego zdania:

Ja nie słyszę po chorobie zapalenia opon mózgowych.
[*ja*][*nie*][*słyszeć*][*po*][*chory*][*zapalenie*][*opony*]

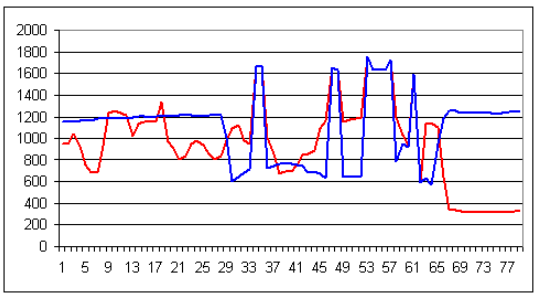
Numery próbek przyporządkowanych przez układ rozpoznający początkom kolejnych słów są następujące: 1, 9, 19, 31, 44, 59, 67, a długościami słów mierzonymi liczbą obserwacji są odpowiednio: 8, 10, 12, 13, 15, 8, 11.



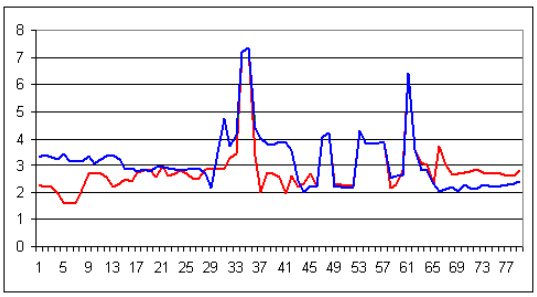
Odległość środka ciężkości dłoni od środka ciężkości twarzy



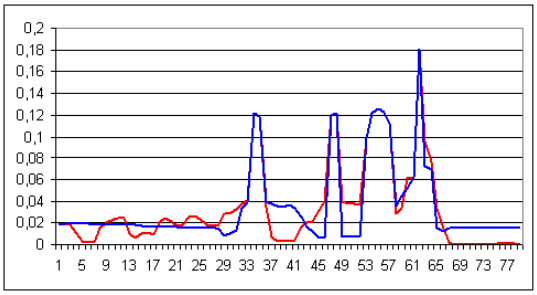
Orientacja odcinka łączącego środki ciężkości dłoni i twarzy



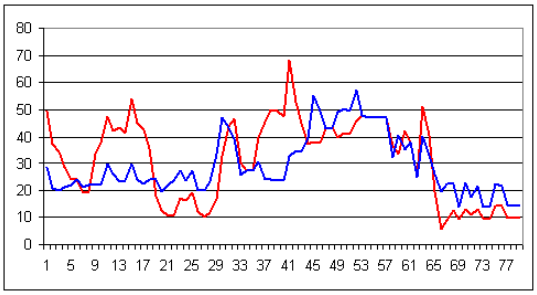
Pole powierzchni



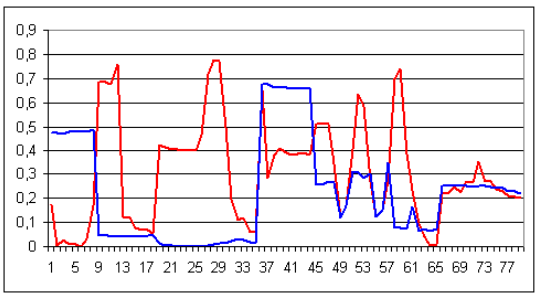
Współczynnik zwartości



Współczynnik ekscentryczności



Średnia różnica głębi twarzy i dłoni



Orientacja osi głównej

Rys. 6.1. *Przykładowe przebiegi cech; dłoń prawa-kolor czerwony, dłoń lewa-kolor niebieski.*

6.2 Rozpoznawanie z wykorzystaniem PaHMM

Przetestowano także rozpoznawanie zdań z wykorzystaniem równoległych modeli Markowa (PaHMM). Rozważono dwa warianty naturalnego podziału wektora cech na niezależne strumienie. W pierwszym z nich, oznaczonym $PaHMM_1$ model równoległy składał się z dwóch zwykłych modeli Markowa odpowiadających dłoni prawej i dłoni lewej. W drugim wariancie $PaHMM_2$, model równoległy składał się z czterech modeli Markowa odpowiadających składowym wektora cech opisującym: (1) położenie dłoni, (2) kształty dłoni, (3) orientację osi głównych i (4) głębię.

Jak pokazano w podrozdziale 4.4 w przypadku modeli równoległych konieczne jest zapamiętywanie dla każdego stanu i propagowanie w sieci kilku znaczników, z którymi skojarzone są najwyższe wartości prawdopodobieństw. W tab. 6.4 przedstawiono wyniki rozpoznawania 35 zdań wykonanych przez osobę B z wykorzystaniem sieci modeli $PaHMM_1$ dla przypadku, gdy liczba propagowanych w sieci znaczników wynosiła od 1 do 5. Zamieszczone dane dotyczą zbiorów testowych dla konfiguracji c_8 i dla wektora cech 27.

Tab. 6.4. *Rozpoznawanie 35 zdań wykonanych przez osobę B z wykorzystaniem sieci modeli $PaHMM_1$ w zależności od liczby propagowanych w sieci znaczników*

liczba znaczników	1	2	3	4	5
skuteczność [%]	96.2	96.6	97.1	98.3	98.3

Najlepszy wynik otrzymano dla czterech znaczników. Dalsze zwiększanie ich liczby nie powodowało poprawy skuteczności rozpoznawania. Dlatego w dalszych eksperymentach przyjęto, że liczba znaczników pamiętanych w każdym stanie i propagowanych w sieci będzie równa cztery.

W tab. 6.5 przedstawiono wyniki rozpoznawania zdań PJM wykonywanych przez dwie osoby. Porównano zwykłe i równoległe modele Markowa w konfiguracji c_8 i dla wektora cech 27.

Tab. 6.5. *Skuteczność rozpoznawania zdań ze zbiorów testowych przez zwykłe i równoległe modele Markowa dla wektora cech 27 i konfiguracji c_8,* [%]

zbiór	HMM	$PaHMM_1$	$PaHMM_2$
A	97.1	96.3	96.6
B	97.4	98.3	98.0
A/B	93.4	92.3	92.6
B/A	92.9	93.1	91.4
AB/A	96.3	98.0	98.3
AB/B	97.4	98.0	98.3

W przypadku A/B i B/A uczono i testowano na zbiorach uczących (testowych) odpowiednich osób. W przypadku AB/A i AB/B zbiór uczący zbudowano z części zbiorów uczących osób A i B i testowano na zbiorach testowych odpowiednich osób.

W wierszu 1 i 2 tab. 6.5 zamieszczono wyniki eksperymentów, w których uczenie i testowanie odbywało się na gestach wykonywanych przez tę samą osobę. Dla zwykłych modeli Markowa skuteczności otrzymane dla osoby A i osoby B są bardzo zbliżone. W przypadku równoległych modeli zaobserwowano, że zdania wykonywane przez osobę B rozpoznawane są lepiej. Wynika to ze sposobu wykonywania gestów przez poszczególne osoby. Osoba A wyuczyła się wybranych wyrazów i zdań PJM na użytek opisywanych eksperymentów i dlatego pokazywane przez nią gesty wykonywane są w sposób przesadnie poprawny, przejścia pomiędzy poszczególnymi wyrazami w zdaniu są wyraźnie akcentowane, obie dłonie zaś zsynchronizowane tak, aby jednocześnie rozpoczynały i kończyły wykonanie danego gestu dwuręcznego. Sposób wykonywania gestów przez osobę A można porównać do sposobu wypowiadania się osoby, która dopiero co nauczyła się obcego języka, potrafi już poprawnie wypowiadać poszczególne wyrazy, ale przejścia pomiędzy nimi w zdaniu nie są jeszcze płynne. Osoba B posługuje się językiem migowym na co dzień (jest tłumaczem języka Polskiego Języka Miganego i działaczką Polskiego Związku Głuchych). Zdania pokazywane przez osobę B wykonywane są w sposób spontaniczny, bardziej naturalnie i płynnie. Konsekwencją tego jest fakt, że ruchy obu dłoni, zwłaszcza w fazie zakończenia jednego wyrazu i rozpoczynania drugiego nie są tak idealnie zsynchronizowane jak w przypadku gestów wykonywanych przez niewprawnego 'mówcę'. Silne jest zjawisko koartykulacji polegające na tym, że dłonie już przy kończeniu danego gestu zaczynają przygotowywać się do wykonania gestu następnego.
W modelu PaHMM modelowane równolegle niezależne procesy nie muszą zmieniać stanów w tych samych dyskretnych chwilach czasowych. Dlatego model ten lepiej radzi sobie w sytuacji, gdy jedna z dłoni rozpocznie wykonywanie danego wyrazu kilka chwil czasowych wcześniej lub gdy dłoń przyjmie kształt charakterystyczny dla początkowej fazy gestu trochę za wcześnie, nie zdążywszy ustawić się w miejscu właściwym dla początku tego gestu. Podobnie jak w przypadku rozpoznawania wyrazów, metoda nie jest niezależna od użytkownika. Jeżeli uczenie i rozpoznawanie przeprowadzane są na gestach różnych osób, to pogorszenie skuteczności rozpoznawania jest większe niż w przypadku rozpoznawania wyrazów. Jeżeli do uczenia wykorzystano gesty wykonane przez obie osoby, otrzymywane wyniki są tylko nieznacznie gorsze niż dla przypadku, gdy uczenie i testowanie odbywało się na gestach tej samej osoby. Wyniki otrzymywane dla modelu $PaHMM_1$ i $PaHMM_2$ są zbliżone. Model $PaHMM_1$ jest jednak prostszy i bardziej naturalny więc zastosowano go w dalszych badaniach.

W celu porównania właściwości uogólniających HMM i PaHMM przeprowadzono eksperyment, w którym usunięto ze zbioru wykorzystywanego w uczeniu wszystkie 20 wykonań 9 wybranych zdań. Wyrazy wchodzące w skład tych zdań były jedynie pokazane w pierwszej wstępnej fazie uczenia. Jeżeli wystąpiły w innych, nieodrzuconych sekwencjach, to uwzględniono je także na etapie embedded training. Odrzucone zdania to:

1. Boli mnie głowa.
   [*boleć*][*mnie*][*głowa*]

2. Czy dzisiaj przyjmuje lekarz rodzinny?

[*czy*][*dzisiaj*][*przyjmować*][*lekarz*][*rodzinny*]

3. Czy przyjmuje inny lekarz w zastępstwie?
   [*czy*][*przyjmować*][*inny*][*lekarz*][*w*][*zastępstwo*]

4. Czy to lekarstwo jest bezpłatne?
   [*czy*][*ten*][*lekarstwo*][*być*][*bezpłatny*]

5. Ja nie słyszę po chorobie zapalenia opon mózgowych.
   [*ja*][*nie*][*słyszeć*][*po*][*chory*][*zapalenie*][*opony mózgowe*]

6. Jestem chory.
   [*być*][*chory*]

7. Mam gorączkę.
   [*mieć*][*gorączka*]

8. Proszę o skierowanie na badania.
   [*prosić*][*o*][*skierowanie*][*badać*]

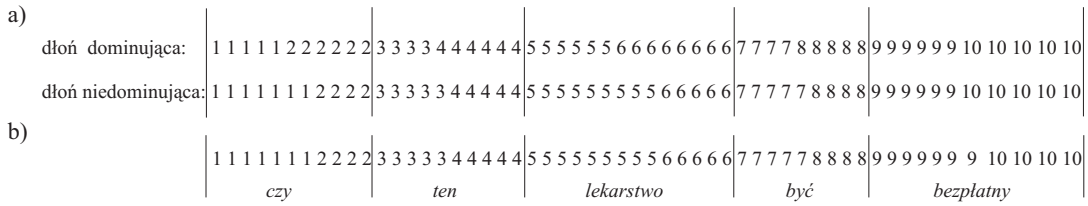
9. Źle się czuję.
   [*źle*][*się*][*czuć*]

Przeprowadzono także eksperyment z wykorzystaniem tylko wyuczonych wcześniej modeli pojedynczych wyrazów (tzn. z pominięciem etapu embedded training). W tab. 6.6 przedstawiono wyniki rozpoznawania sekwencji nie pokazanych w trakcie uczenia.

Tab. 6.6. *Rozpoznawanie sekwencji nie pokazanych w trakcie uczenia dla osoby A*

	180 zdań	700 zdań
HMM	26.7%	31.0%
$PaHMM_1$	58.3%	69.7%

Eksperymenty powtórzono dla gestów wykonywanych przez drugą osobę i dla innych zestawów 9 odrzucanych zdań, otrzymując za każdym razem zbliżone wyniki. Przy rozpoznawaniu z wykorzystaniem HMM i PaHMM miał zastosowanie algorytm Viterbiego w wersji z przekazywaniem znaczników. Zbudowano narzędzie pozwalające na odtwarzanie sekwencji ukrytych stanów odpowiadających wygrywającym znacznikom. Zaobserwowano, że dla wykonań, które PaHMM rozpoznaje poprawnie a HMM niepoprawnie, sekwencje ukrytych stanów nie są identyczne w obu kanałach odpowiadających dłoni dominującej i niedominującej. Schematycznie ilustruje to rys. 6.2 odnoszący się do przypadku, gdy model Markowa każdego słowa miał dwa stany emitujące.
W modelach równoległych, dla niektórych wyrazów, przejścia z pierwszego stanu emitującego do drugiego stanu emitującego nie odbywają się w tych samych chwilach. Najczęściej następują one szybciej dla dłoni dominującej. Zaobserwowano to dla wyrazów: *boleć*, *mnie*, *głowa*, *czy*, *lekarz*, *to*, *lekarstwo*, *być*, *ja*, *nie*, *słyszeć*,



Rys. 6.2. *Sekwencje ukrytych stanów przy rozpoznawaniu zdania Czy to lekarstwo jest bezpłatne?: a) model $PaHMM_1$, b) model HMM.*

chory, *opony mózgowe*, *mieć*, *o*, *na*, *badać*, *źle*, *się*, *czuć*.
Przeprowadzano także analizę sekwencji ukrytych stanów dla dwóch różnych wykonań tego samego zdania, z których pierwsze jest rozpoznawane przez regularny model Markowa a drugie nie. Wprawdzie porównywane wykonania mogą mieć różną długość ciągu obserwacji, ale daje się zauważyć, że dla większości z wymienionych wyrazów przejścia z pierwszego do drugiego stanu emitującego odbywają się zazwyczaj wcześniej dla zdań, które HMM rozpoznaje. Otrzymane wyniki wskazują, że rozbicie wektora cech na części i wykorzystanie ich równolegle wprowadza dodatkowe stopnie swobody. Dlatego model równoległy PaHMM jest bardziej elastyczny niż regularny HMM i lepiej radzi sobie w przypadku niedoboru danych uczących. Może to mieć duże znaczenie w przypadku budowania systemów rozpoznawania z dużymi słownikami gestów, gdy proces gromadzenia danych uczących jest uciążliwy i czasochłonny.

6.3 Podsumowanie

W rozdziale opisano wyniki rozpoznawania 35 zdań pokazanych w wariancie użytkowym PJM, występujących w typowych sytuacjach: u lekarza i na poczcie. Zdania wykonywane były przez lektorkę PJM oraz autora pracy. Łącznie wykorzystano 1400 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przyjęto, że pojedynczy model Markowa odpowiada wyrazowi, zaś całe zdanie modelowane jest z wykorzystaniem sieci złożonej z połączonych odpowiednio modeli. Modele odpowiadające poszczególnym wyrazom miały po cztery stany (rys. 4.2), przy czym stany 1 i 4 były nieemitujące, zaś funkcja gęstości prawdopodobieństwa obserwacji dla każdego ze stanów emitujących miała postać sumy dwóch rozkładów Gaussa.

Przebadano 15 różnych konfiguracji sieci ukrytych modeli Markowa. W części z nich zastosowano modele przejścia z jednym stanem emitującym. Testowano układy ze zwykłymi modelami przejścia oraz z modelami przejścia typu 'Tee', w których istniej dodatkowa tranzycja z pierwszego do ostatniego stanu nieemitującego, pozwalająca na pominięcie modelu w pewnych sytuacjach. Rozważano sieci HMM z jednym modelem przejścia oraz sieci, w których każda para występujących po sobie wyrazów połączona była innym modelem przejścia. Dla sieci, w których po danym wyrazie może wystąpić dowolny wyraz otrzymano bardzo słabe wyniki nie przekraczające 10.6%, niezależnie od tego czy zastosowano modele przejścia, czy też nie.

Zatem konfiguracje te nie nadają się do rozpoznawania zdań PJM z wykorzystaniem zaproponowanej metody. Dopiero wprowadzenie modelu lingwistycznego bigram, uwzględniającego informację o kontekście pozwoliło na uzyskanie skuteczności rozpoznawania 81.4% dla tych konfiguracji HMM, które dopuszczały jedynie sekwencje wyrazów tworzące logiczne wypowiedzi. Dla sieci, w których niedozwolone sekwencje wyrazów wyeliminowane są poprzez sam układ ukrytych modeli Markowa uzyskano skuteczność rozpoznawania 96.1%. Dla konfiguracji tych zauważono, że dodanie modelu przejścia, który nie jest typu 'Tee' pogarsza skuteczność rozpoznawania. Dzieje się tak dlatego, że w procesie uczenia poszczególnych modeli wykorzystywane są wyrazy pochodzące z rozpatrywanych sekwencji, zatem modele są modyfikowane tak, aby uwzględnić spowodowane procesem koartykulacji zniekształcenia na początku i końcu gestu. W konsekwencji, podczas rozpoznawania początkowa część przejścia pomiędzy dwoma wyrazami traktowana jest jako końcówka pierwszego wyrazu, zaś część końcowa przejścia zostaje dołączona na początek drugiego wyrazu.

W rozdziale opisano także eksperymenty z wykorzystaniem równoległych ukrytych modeli Markowa. Rozważono dwa warianty zrównoleglenia. W pierwszym równoległe kanały odpowiadały dłoni prawej i lewej, w drugim zaś poszczególnym grupom cech: położeniu, kształtowi, orientacji i informacji 3D. Dla gestów wykonywanych przez autora pracy wyniki otrzymywane w przypadku PaHMM były porównywalne z uzyskanymi dla modeli zwykłych. Poprawę zaobserwowano natomiast dla gestów wykonywanych przez lektorkę PJM. Mający więcej stopni swobody model równoległy lepiej poradził sobie w przypadku wykonania spontanicznego, w którym osoba pokazująca znaki czyni to prawie bezwiednie nie przywiązując tak dużej uwagi do tego, aby poszczególne dłonie były idealnie zsynchronizowane.

Omówiono także eksperyment pozwalający na przebadanie własności uogólniających zwykłych i równoległych modeli Markowa. W tym celu usuwano ze zbioru uczącego niektóre bądź wszystkie zdania, ucząc poszczególne modele jedynie na pojedynczych wykonaniach występujących w nich wyrazów. Badania pokazały, że modele równoległe są bardziej elastyczne i lepiej radzą sobie w przypadku niedoboru danych uczących, co może mieć znaczenie przy konstruowaniu złożonych układów rozpoznających zdania z obszernymi słownikami wyrazów.

Rozdział 7

Podsumowanie

Celem pracy było opracowanie i weryfikacja metody przybliżającej zbudowanie systemu wizyjnego do rozpoznawania słów i zdań Polskiego Języka Miganego. Według rozeznania autora, tak sformułowanego zadania dotyczącego PJM nie rozpatrywano dotąd. Konieczne było rozwiązanie od podstaw szeregu problemów cząstkowych, a następnie połączenie tych rozwiązań w prototypowym układzie informatycznym i eksperymentalne przebadanie całości.

Do najważniejszych osiągnięć pracy można zaliczyć:

- zaproponowanie schematu przetwarzania obrazów pozyskiwanych w stereowizyjnym układzie kamer kolorowych w celu wyznaczenia wektorów cech,
- dobór metody identyfikacji dłoni i twarzy w obrazach kolorowych uwzględniający jakość detekcji w zmiennych warunkach oświetlenia oraz czas przetwarzania pozwalający na zastosowania metody w trybie on-line; eksperymentalne uzasadnienie wyboru na podstawie oceny kilku wariantów detekcji koloru skóry w różnych przestrzeniach barw, w odniesieniu do obszernego zbioru danych,
- dobór metody wyznaczania mapy dysparycji na podstawie obrazów stereo pod kątem jakości otrzymywanych map oraz czasu przetwarzania determinującego zastosowanie w trybie on-line; eksperymentalne uzasadnienie wyboru z uwzględnieniem różnych rozwiązań,
- opracowanie i praktyczna weryfikacja algorytmu identyfikacji dłoni prawej, lewej i twarzy w otrzymanych obrazach,
- oparta na badaniach lingwistycznych dotyczących cech dystynktywnych znaków migowych propozycja wariantów wektora cech,
- opracowanie metody rozpoznawania pojedynczych słów oraz metody rozpoznawania zdań PJM; uzasadnienie szczegółów rozwiązań na podstawie wielostronnych badań eksperymentalnych, dotyczących zwłaszcza:
  - struktury ukrytych modeli Markowa i równoległych modeli Markowa,
  - wyboru wariantu wektora cech,

  - zależności skuteczności rozpoznawania od warunków oświetlenia i wykonawcy gestów,
  - zdolności rozpoznawania nowych zdań zbudowanych z wyuczonych przez system słów.

- przygotowanie zbioru narzędzi programowych o ogólniejszym przeznaczeniu oraz opracowanie z ich wykorzystaniem aplikacji umożliwiającej rozpoznawanie wybranych wyrazów i zdań PJM w trybie on-line, a także obszerna baza sekwencji wizyjnych, która została udostępniona w Internecie.

Schemat przetwarzania obrazów w celu wyznaczania wektorów cech wykorzystuje obrazy kolorowe pozyskiwane w układzie stereowizyjnym. Do identyfikacji dłoni i twarzy osoby wykonującej gest zastosowano metodę opartą o zbudowany uprzednio model rozkładu chrominancji skóry ludzkiej w znormalizowanej przestrzeni barw RGB. Wybór metody poprzedzony był studiami literaturowymi oraz opisanymi w rozdziale 3 eksperymentami dla 4 różnych metod identyfikacji i 9 wybranych przestrzeni barw o właściwościach istotnych z punktu widzenia segmentacji. Testy przeprowadzono z wykorzystaniem, przygotowanej w tym celu bazy danych, zawierającej 162 obrazy testowe dłoni i 108 obrazów twarzy, zarejestrowanych dla dwóch osób, w pomieszczeniu zamkniętym, w dzień słoneczny i pochmurny oraz w nocy przy oświetleniu sztucznym. Przetestowano metodę z aproksymacją histogramu kolorów za pomocą rozkładu normalnego, metodę z wygładzaniem histogramu kolorów filtrem Gaussa, metodę największej wiarygodności i metodę maksimum prawdopodobieństwa a posteriori. Rozważano znormalizowaną przestrzeń RGB oraz przestrzenie YUV, YIQ, barw przeciwstawnych OCS, barw przeciwstawnych w wersji logarytmicznej OCSL, I1I2I3, IHS oraz Lab. Jakość uzyskiwanych obrazów binarnych oceniano porównując je z obrazami otrzymanymi w wyniku ręcznej segmentacji wszystkich obrazów testowych. Przy wyborze metody i przestrzeni barw uwzględniono także czasy wykonania obliczeń na typowym komputerze PC.

Do identyfikacji dłoni prawej, lewej i twarzy w otrzymanym obrazie binarnym opracowano algorytm wykorzystujący informację o polach powierzchni i środkach ciężkości otrzymanych obiektów binarnych w bieżącej i poprzedniej klatce. Przeprowadzone eksperymenty wykazały, że dla rozważanego słownika gestów, typowego tempa wykonywania oraz częstotliwości przetwarzania 25 klatek/s obiekty identyfikowane są poprawnie przy założeniu, że algorytm rozpoczyna działanie, gdy dłonie i twarz dają w obrazie rozłączne obiekty i dłoń prawa znajduje się po prawej, a lewa po lewej stronie osi ciała (dokładniej linii pionowej przechodzącej przez środek ciężkości twarzy).

Ponieważ kształty przyjmowane przez dłonie oraz trajektorie ruchu dłoni mają charakter przestrzenny, zastosowano układ stereowizyjny, aby dodać do wektora cech informację 3D. W tym celu wykorzystano rzadkie mapy dysparycji otrzymane w wyniku przetworzenia obrazów stereo. Wybór odpowiedniej metody poprzedzony był testami różnych algorytmów. Przetestowano dające zwarte mapy dysparycji: (1) korelacyjne metody poszukiwania odpowiedników dla okien korelacji o rozmiarach 3×3, 5×5, ..., 31×31 i 10 różnych miar dopasowania, (2) metody korelacyjne zmodyfikowane w ten sposób, aby mapy dysparycji generowane były tylko dla obszarów

dłoni i twarzy, (3) metodę poszukiwania z wykorzystaniem programowania dynamicznego i (4) metodę generowania rzadkiej mapy dysparycji na podstawie obrazów krawędzi otrzymanych w wyniku filtracji LOG. Oceniano wizualnie jakości otrzymywanych map dysparycji oraz zmierzono czasy przetwarzania. Metody korelacyjne dla okien o rozmiarach 17×17 i większych umożliwiały otrzymywanie map dysparycji dobrej jakości, ale czasy przetwarzania, nawet po zastosowaniu optymalizacji algorytmu lub ograniczeniu poszukiwania odpowiedników tylko do obszarów o barwie skóry były dłuższe niż 400 ms, co uniemożliwiło zastosowanie tych metod do przetwarzania w trybie on-line na wykorzystywanym w systemie typowym komputerze PC. W przypadku metody programowania dynamicznego problematyczny okazał się dobór takich parametrów algorytmu, które pozwoliłyby uzyskiwać mapy dysparycji dobrej jakości dla wszystkich klatek w danej sekwencji wideo. Czas przetwarzania 800 ms także nie pozwalał na zastosowanie tej metody w czasie rzeczywistym. Dopiero metoda generowania rzadkiej mapy dysparycji na podstawie obrazów krawędzi dała mapy zadowalającej jakości w czasie przetwarzania rzędu 15 ms.

Przesłanką do wyboru wektorów cech były dostępne w literaturze wyniki badań lingwistycznych nad tzw. wiązkami cech dystynktywnych, pozwalającymi jednoznacznie opisać znak migowy PJM. Cechy podzielono na cztery grupy opisujące miejsce artykulacji, kształt, orientację i głębię dłoni. Miejsce artykulacji, analogicznie jak w przypadku gestogramów (patrz podrozdział 2.2), określono względem innej części ciała. Jako odniesienie wybrano twarz, ponieważ podczas konwersacji przeprowadzanej z wykorzystaniem języka migowego zazwyczaj jest ona statyczna i musi być zwrócona w kierunku odbiorcy, tak aby możliwa była obserwacja wykonywanych gestów i ewentualnie czytanie z ruchu ust. Dodatkowym uzasadnieniem takiego wyboru, z punktu widzenia przetwarzania, jest fakt, że chrominancja twarzy jest zbliżona do chrominancji dłoni, co pozwala na zastosowanie tych samych metod identyfikacji. Przyjęto, że położenie dłoni względem twarzy będzie określone za pomocą długości odcinka łączącego środki ciężkości dłoni i twarzy i orientacji przechodzącej przez nie prostej. Rozpoznawanie wyrazów i zdań PJM wymaga, aby pole widzenia kamer obejmowało sylwetkę osoby wykonującej gest co najmniej od pasa w górę. W takim przypadku rozmiary dłoni w obrazie są zbyt małe, aby możliwa była dokładna analiza ich kształtu z uwzględnieniem układu i orientacji poszczególnych palców, tak jak ma to miejsce w zapisie gestograficznym. Dlatego przy wyborze cech opisujących kształt dłoni zdecydowano się na opis zgrubny z wykorzystaniem jedynie przybliżonej "miary" danego kształtu. Zastosowano trzy różne miary: pole powierzchni, współczynnik zwartości oraz ekscentryczność i rozważono różne ich kombinacje. Ponieważ kształt dłoni i wykonywane ruchy mają charakter przestrzenny, dodano do wektora cech informację o wzajemnym, przestrzennym usytuowaniu dłoni i twarzy osoby wykonującej gesty. Opis dłoni uzupełniono o orientację osi głównej odpowiadającego jej obiektu binarnego.

W najbardziej złożonym przypadku mamy zatem wektor cech złożony z 14 elementów, po 7 na każdą dłoń. Cechy, które z lingwistycznego punktu widzenia są dystynktywne, mogą w przypadku przetwarzania obrazów te właściwości utracić. Ponadto w zaproponowanym wektorze cech opis kształtu jest znacznie uboższy aniżeli w gestograficznym zapisie statycznej konfiguracji dłoni. Rozpoznawanie wyrazów

PJM z wykorzystaniem rozważanych wariantów wektora cech wymaga więc eksperymentalnej weryfikacji.

Przetestowano rozpoznawanie podzbioru 101 wyrazów PJM występujących w typowych sytuacjach: u lekarza i na poczcie. Gesty wykonywane były przez lektorkę PJM oraz przez autora, który wyuczył się ich na użytek niniejszej pracy. Łącznie wykorzystano 6060 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przebadano 32 różne warianty wektora cech (tab. 3.7), uwzględniając przypadki, gdy uczenie i testowanie odbywało się na gestach tej samej osoby, gdy testowanie odbywało się na gestach innej osoby niż uczenie oraz gdy zbiór uczący został zbudowany w oparciu o gesty wykonane przez obie osoby. Modele odpowiadające poszczególnym wyrazom miały po 4 stany (rys. 4.2), przy czym stany 1 i 4 były nieemitujące, zaś funkcja gęstości prawdopodobieństwa obserwacji dla każdego ze stanów emitujących miała postać sumy dwóch rozkładów Gaussa. Przedstawiono rezultaty otrzymane dla zbiorów testowych oraz wyniki powstałe wskutek uśrednienia wyników czterokrotnej walidacji skrośnej (tab. 5.1). Najlepszy rezultat 99.3% otrzymano dla przypadku, gdy wektor cech zawierał informację o położeniu, informację o kształcie wyrażoną za pomocą pola powierzchni i współczynnika zwartości, informację 3D i informację o orientacji dłoni. Przeprowadzono popartą przykładami dyskusję na temat wpływu poszczególnych elementów wektora cech na skuteczność rozpoznawania. W przypadku wykorzystania modeli wyuczonych na gestach innej osoby niż rozpoznawane zaobserwowano wyraźne pogorszenie otrzymanych rezultatów, co wskazuje na to, że metoda nie jest całkiem niezależna od użytkownika. Bliższa analiza pokazała, że pewne wektory cech okazują się korzystniejsze w rozpoznawaniu niektórych słów, inne zaś dominują w pozostałych przypadkach. Dlatego przebadano skuteczność rozpoznawania z wykorzystaniem kombinacji klasyfikatorów opartych na różnych wektorach cech. Zastosowano metodę głosowania z większością zwykłą i z większością bezwzględną, metodę pozycyjną Bordy, metodę uśrednień Bayesa oraz całkę rozmytą. W przypadku kombinacji pierwszych 15 wariantów wektora cech (tab 3.7) najlepsze wyniki otrzymano dla głosowania i całki rozmytej. Przewyższały one o ok. 4% rezultat, jaki dawał najlepszy ze wspomnianych wektorów cech. W przypadku, gdy uwzględniono wszystkie warianty wektora cech wynik globalnie najlepszy był tylko nieznacznie gorszy od hipotetycznego rezultatu, jaki otrzymanoby dobierając wektor cech indywidualnie dla każdego słowa. Dlatego kombinacja wymagała staranniejszego wyboru wektorów cech, na których opierały się klasyfikatory, a poprawa skuteczności rozpoznawania okazała się niewielka, jak można było oczekiwać.

Przeprowadzono także eksperymenty dla różnej liczby ukrytych stanów. Dla każdego z rozważanych wariantów wektora cech, eksperyment z rozpoznawaniem wyrazów powtórzono wykorzystując ukryte modele Markowa mające od 2 do 10 stanów emitujących. Dla wszystkich wariantów wektora cech, najlepsze skuteczności rozpoznawania otrzymywano, gdy liczba stanów była równa 2 lub 3 lub 4 lub 5. We wszystkich przypadkach obserwowano też pogarszanie się wyników, gdy liczba stanów była większa od 5. Wykonano także eksperyment, w którym liczby stanów w modelach odpowiadających poszczególnym wyrazom były ustalane indywidualnie. Wykorzystując wyniki otrzymane dla liczby stanów zmieniającej się od 2 do 10

wyznaczono średnie prawdopodobieństwa wygenerowania danego wyrazu w odpowiadających mu modelach o różnej liczbie stanów i wybrano ten model, dla którego prawdopodobieństwo to było największe. Dla wszystkich rozważanych wariantów wektora cech wyniki otrzymane dla modeli o ustalonej indywidualnie, zróżnicowanej liczbie stanów okazały się nieznacznie lepsze aniżeli w przypadku, gdy liczba stanów modeli odpowiadających poszczególnym wyrazom była jednakowa. Przeprowadzone eksperymenty pokazały, że dla liczby stanów większej od 5 model dobrze dopasowuje się do danych użytych w uczeniu, ale traci swe właściwości uogólniające, zatem do rozpoznawania wyrazów PJM powinno się raczej projektować modele Markowa o mniejszej liczbie stanów.

Przebadano rozpoznawanie 35 zdań pokazanych w wariancie użytkowym PJM, występujących w typowych sytuacjach: u lekarza i na poczcie. Zdania wykonywane były przez lektorkę PJM oraz autora pracy. Łącznie wykorzystano 1400 wykonań, zarejestrowanych w korzystnych i niekorzystnych warunkach oświetlenia. Przyjęto, że pojedynczy model Markowa odpowiada wyrazowi, całe zdanie modelowane jest zaś z wykorzystaniem sieci złożonej z połączonych odpowiednio modeli słów. Modele odpowiadające poszczególnym wyrazom miały po cztery stany (rys. 4.2), przy czym stany 1 i 4 były nieemitujące, zaś funkcja gęstości prawdopodobieństwa obserwacji dla każdego ze stanów emitujących miała postać sumy dwóch rozkładów Gaussa.

Przebadano 15 różnych konfiguracji sieci ukrytych modeli Markowa. W części z nich zastosowano modele przejścia posiadające jeden stan emitujący. Testowano układy ze zwykłymi modelami przejścia oraz z modelami przejścia typu 'Tee', w których istnieje dodatkowa tranzycja z pierwszego stanu nieemitującego do ostatniego stanu nieemitującego, pozwalająca na pominięcie modelu przejścia. Rozważano sieci HMM z modelami przejścia o jednakowych parametrach oraz sieci, w których parametry te zależały od pary wyrazów, między którymi przejście modelowano. Dla sieci, w których po danym wyrazie może wystąpić dowolny wyraz otrzymano bardzo słabe wyniki nie przekraczające 10.6%, niezależnie od tego, czy zastosowano modele przejścia czy też nie. Zatem konfiguracje te nie nadają się do rozpoznawania zdań PJM z wykorzystaniem zaproponowanej metody. Dopiero wprowadzenie modelu lingwistycznego bigram, uwzględniającego informację o kontekście pozwoliło na uzyskanie skuteczności rozpoznawania 81.4% dla tych konfiguracji HMM, które dopuszczały sekwencje wyrazów nie tworzące logicznych wypowiedzi. Dla sieci, w których niedozwolone sekwencje wyrazów wyeliminowane są poprzez sam układ ukrytych modeli Markowa uzyskano skuteczność rozpoznawania 96.1%. Dla konfiguracji tych zauważono, że dodanie modelu przejścia, który nie jest typu 'Tee' pogarsza skuteczność rozpoznawania. Dzieje się tak dlatego, że w procesie uczenia poszczególnych modeli wykorzystywane są wyrazy pochodzące z rozpatrywanych sekwencji, zatem modele są modyfikowane tak, aby uwzględnić spowodowane procesem koartykulacji zniekształcenia na początku i końcu gestu. W konsekwencji podczas rozpoznawania początkowa część przejścia pomiędzy dwoma wyrazami traktowana jest jako końcówka pierwszego wyrazu, część końcowa przejścia zostaje zaś dołączona na początek drugiego wyrazu.

Przeprowadzono także eksperymenty z wykorzystaniem równoległych ukrytych modeli Markowa. Rozważono dwa warianty zrównoleglenia. W pierwszym równoległe

kanały odpowiadały dłoni prawej i lewej, w drugim zaś poszczególnym grupom cech, tj.: położeniu, kształtowi, orientacji i informacji o głębi. Dla gestów wykonywanych przez autora pracy wyniki otrzymywane w przypadku PaHMM były porównywalne z tymi uzyskanymi dla modeli zwykłych. Poprawę zaobserwowano natomiast dla gestów wykonywanych przez lektorkę PJM. Mający więcej stopni swobody model równoległy lepiej poradził sobie w przypadku wykonania spontanicznego, w którym osoba pokazująca znaki czyni to prawie bezwiednie, nie przywiązując tak dużej uwagi do tego aby poszczególne dłonie były idealnie zsynchronizowane.

Porównano własności uogólniające zwykłych i równoległych modeli Markowa. W tym celu usuwano ze zbioru uczącego część bądź wszystkie zdania, ucząc poszczególne modele jedynie na wykonaniach występujących w nich pojedynczych wyrazów. Badania pokazały, że modele równoległe są bardziej elastyczne i lepiej radzą sobie w przypadku niedoboru danych uczących. Ich skuteczność rozpoznawania w rozważanej sytuacji okazała się bliska 70% i była około dwukrotnie wyższa od skuteczności uzyskanej ze zwykłymi HMM. Może to mieć znaczenie przy konstruowaniu złożonych układów rozpoznających z obszernymi słownikami wyrazów.

Wynikiem przedstawionych w pracy badań jest prototypowy układ pozwalający na rozpoznawanie w trybie on-line pojedynczych słów i prostych zdań PJM. Ocena układu dokonana dla 101 wyrazów i 35 zdań potwierdza jego przydatność w zastosowaniu do ograniczonego słownika. Uzyskana skuteczność rozpoznawania 99,3% dla słów i 97,4% dla zdań przedstawia się korzystnie w świetle zamieszczonych w tab. 1.1 wyników opublikowanych w literaturze w odniesieniu do układów wizyjnych niewymagających zastosowania pomocniczych rękawic. Odnosi się to również do wielkości słownika i szybkości działania.

Nieco ponad 30% znaków Polskiego Języka Miganego ma charakter ikoniczny. Oznacza to, że charakterystyczny dla nich układ dłoni i palców jest podobny do opisywanego kształtu, albo wykonywany ruch jest zbliżony do ruchu typowego dla opisywanej czynności. Intuicyjność tej grupy znaków sprawia, że mogą one z powodzeniem być wykorzystane w innych zastosowaniach, np. w konstruowaniu wielomodalnych interfejsów komunikacji człowieka z maszyną albo w środowisku gdzie z uwagi na wysoki poziom hałasu słowne wydawanie komend dla komputera bądź robota jest niemożliwe. Doświadczenia zdobyte w trakcie realizacji niniejszej pracy mogą więc być wykorzystane także przy rozwiązywaniu szeroko rozumianych zadań z dziedziny interakcji człowieka z maszyną.

Obecny stan prac nad rozpoznawaniem języków migowych oraz doświadczenia zgromadzone w niniejszej pracy wyznaczają kierunki dalszych badań. W komunikacji z osobami niesłyszącymi, obok gestów wykonywanych przy udziale dłoni i ramion, duże znaczenie mają także sygnały niemanualne przekazywane za pośrednictwem mimiki twarzy, ruchów głowy, układu ciała czy ruchów torsu. Automatyczna interpretacja przekazu migowego w sposób pełny i niezawodny wymaga zatem przetwarzania informacji o charakterze wielomodalnym. Pociąga to za sobą konieczność rozwiązania ważnych problemów badawczych i technicznych związanych z jednoczesnym rejestrowaniem, precyzyjną synchronizacją i integracją kanałów informacyjnych o różnorodnym charakterze. Złożoność przekazu migowego sprawia, że konieczne staje się łączenie doświadczeń zdobytych przy rozwiązywaniu pokrewnych,

aczkolwiek traktowanych zazwyczaj niezależnie, zadań z zakresu sztucznej inteligencji, takich jak: detekcja, śledzenie i identyfikacja osób, automatyczna analiza i interpretacja akcji i zachowań człowieka, rozpoznawanie mimiki twarzy, śledzenie i analiza ruchów ciała, budowanie wielomodalnych interfejsów do komunikacji z maszyną, czy wreszcie interakcja z obiektami rzeczywistości wirtualnej.

Kolejnym wyzwaniem jest rozwiązanie problemu skalowalności. Ponieważ dotychczasowe prace nad rozpoznawaniem języków migowych dotyczą głównie ograniczonych słowników gestów. Problem wiąże się z wyodrębnieniem i rozpoznawaniem elementarnych składników gestów, pełniacych podobną rolę jak fonemy w języku mówionym. Wymaga to badań w zakresie języka miganego oraz badań nad systemami wizyjnymi z kamerami śledzącymi dłonie i pozwalającymi na obserwowanie układu palców z rozdzielczością umożliwiającą rozpoznanie niezbędnych szczegółów. Sygnalizowane dotąd rozpoznawanie obszerniejszych słownikow bazuje na danych otrzymywanych za pośrednictwem specjalnych rękawic.

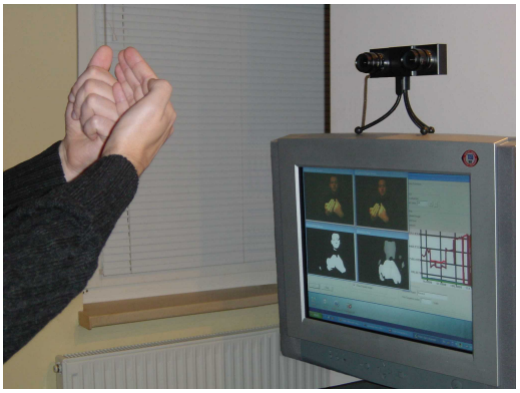
Dodatek A

Stanowisko do rozpoznawania wyrazów i zdań PJM

Rezultatem niniejszej pracy jest stanowisko badawcze pozwalające na rozpoznawanie w trybie on-line wybranych 101 wyrazów i 35 zdań PJM (patrz dodatek E), używanych w typowych sytuacjach życiowych: u lekarza i na poczcie. Wymagane jest aby osoba wykonująca gesty miała ubranie z długim rękawem w kolorach odmiennych od barwy skóry. Osoba ta powinna znajdować się w odległości około 1.5 m od kamery i być zwrócona twarzą do niej. System musi być ustawiony tak, aby w tle nie pojawiły się inne osoby lub obiekty o barwach zbliżonych do koloru skóry.

A.1 Wymagania sprzętowe

Stanowisko składa się z typowego komputera PC wyposażonego w kartę interfejsu IEEE 1394 FireWire, do której dołączona jest stereowizyjna kamera STH-MDCS/-C firmy Videre Design(rys. A.1. A.2). Opcjonalnie komputer wyposażony może być



Rys. A.1. *Stanowisko badawcze.*

także w kartę dźwiękową i mikrofon, pozwalające na usprawnienie procesu rejestrowania sekwencji wizyjnych poprzez zaznaczanie ich początków i końców za pomocą komend głosowych. W eksperymentach wykorzystano komputer o następujących

Rys. A.2. *Kamera stereowizyjna STH-MDCS/-C firmy Videre Design.*

parametrach: procesor AMD Athlon 1.8GHz, pamięć operacyjna 1GB, dysk twardy 100GB. Przygotowane oprogramowanie będzie także działać na jednostkach wyposażonych w minimalną, zalecaną dla systemu Windows XP ilość pamięci operacyjnej 256 MB.

A.2 Składniki oprogramowania

Na komputerze zainstalowany jest system operacyjny Windows XP. Przygotowane oprogramowanie będzie działać także na wcześniejszych wersjach systemu Windows z wyjątkiem modułu do sterowania procesem nagrywania sekwencji za pomocą komend głosowych. Moduł ten wymaga obecności platformy Microsoft .NET Framework i Microsoft Speech SDK, dostępnych standardowo w systemie XP.

Podstawowym składnikiem systemu jest aplikacja rozpoznająca. Aplikacja ta napisana została przez autora pracy w języku C++ z wykorzystaniem środowiska programistycznego Microsoft Visual Studio .NET i biblioteki Microsoft Foundation Class (MFC). Do konfiguracji kamery, akwizycji i rektyfikacji par obrazów stereo oraz wyznaczania map głębi aplikacja wykorzystuje wybrane funkcje z biblioteki Small Vision System (SVS) [42]. SVS jest dostarczana wraz z kamerą w formie biblioteki DLL. Jest to produkt licencjonowany i niektóre funkcje działają tylko na komputerze wyposażonym w kartę i kamerę Videre Design. Do przetwarzania pozyskiwanych sekwencji wizyjnych aplikacja wykorzystuje zoptymalizowane względem czasu wykonania funkcje z napisanej przez autora pracy biblioteki podstawowych procedur przetwarzania obrazu (patrz dodatek B). Rozpoznawanie odbywa się z wykorzystaniem funkcji z pakietu Hidden Markow Toolkit (HTK). Wraz z tym pakietem dostarczane są kody źródłowe dostępnych w nim narzędzi, co pozwala na przeniesienie zaimplementowanych w pakiecie algorytmów do własnej aplikacji. Aplikacja rozpoznająca łączy się poprzez interfejs ODBC z zaprojektowaną przez autora bazą danych przygotowaną w MySQL (patrz dodatek C). Umożliwia to rejestrację nagrywanych sekwencji i wyników ich przetwarzania, co ułatwia proces gromadzenia i zarządzania obszernym materiałem badawczym.

Niezależnie od aplikacji rozpoznającej wykorzystywane są także: SRI Smallv Stereo System (Smallvcal.exe) - dostarczane wraz z kamerą oprogramowanie do ka-

libracji układu, skrypty narzędziowe pakietu HKT do tworzenia i uczenia ukrytych modeli Markowa (patrz dodatek D) i aplikacja Shoot.exe - darmowe oprogramowanie do sterowania komputerem za pomocą komend głosowych. Sterowanie głosem zostało skonfigurowane tak, aby po rozpoznaniu słów start (stop) wysłać do okienka aktywnej aplikacji (w naszym przypadku aplikacji rozpoznającej) komunikaty systemu Windows, symulując w ten sposób naciskanie przycisku wznawiającego (zatrzymującego) rejestrację sekwencji wizyjnej.

A.3 Przygotowanie do uruchomienia

Przed wykorzystaniem układ stereowizyjny musi zostać skalibrowany. Wykorzystywana jest w tym celu dostarczana wraz z kamerą aplikacja Smallvcal.exe i specjalnie przygotowana plansza (rys. 3.9). Kalibracja odbywa się z wykorzystaniem algorytmu Tsai. Procedurę kalibracji należy powtórzyć każdorazowo, gdy położenie układu kamer uległo zmianie (zmieniły się zewnętrzne parametry modelu każdej z kamer), albo gdy zmieniono ustawienia ostrości poszczególnych obiektywów (zmieniły się parametry wewnętrzne modeli kamer). Wynikiem kalibracji jest plik tekstowy zawierający wyznaczone parametry zewnętrzne i wewnętrzne dla obu kamer oraz macierze, które wykorzystywane będą w procesie rektyfikacji par stereo.

Następnie należy przygotować obraz, który zostanie wykorzystany do zbudowania modelu rozkładu chrominancji dla skóry ludzkiej. W tym celu należy zarejestrować obraz zawierający wyprostowaną dłoń ze złączonymi palcami skierowanymi ku górze, zwróconą wewnętrzną stroną do kamery (znak daktylograficzny dla litery B, patrz podrozdział 2.1, rys. 2.1). Najlepiej jest, aby dłoń znajdowała się w takiej odległości od kamery, w jakiej będą wykonywane gesty. Obraz można zarejestrować z wykorzystaniem aplikacji Smallvcal.exe, która wykorzystywana była uprzednio do kalibracji, albo za pomocą aplikacji rozpoznającej. Następnie należy wyciąć z otrzymanego obrazu prostokątny obszar zawierający fragment dłoni. Należy to zrobić tak, aby w wyciętym okienku obrazu znalazły się tylko piksele skóry. Można do tego wykorzystać dowolną aplikację do przetwarzania obrazów, także standardowo dostępny w systemie Windows pakiet Paint. Wycięte okienko musi zostać zapisane jako obraz bmp z wykorzystaniem zapisu 24 bitowego (po 8 bitów na każdą składową).

W kolejnym kroku należy przygotować pliki z wyuczonymi modelami Markowa (patrz dodatek D) i przegrać je do jednego folderu.

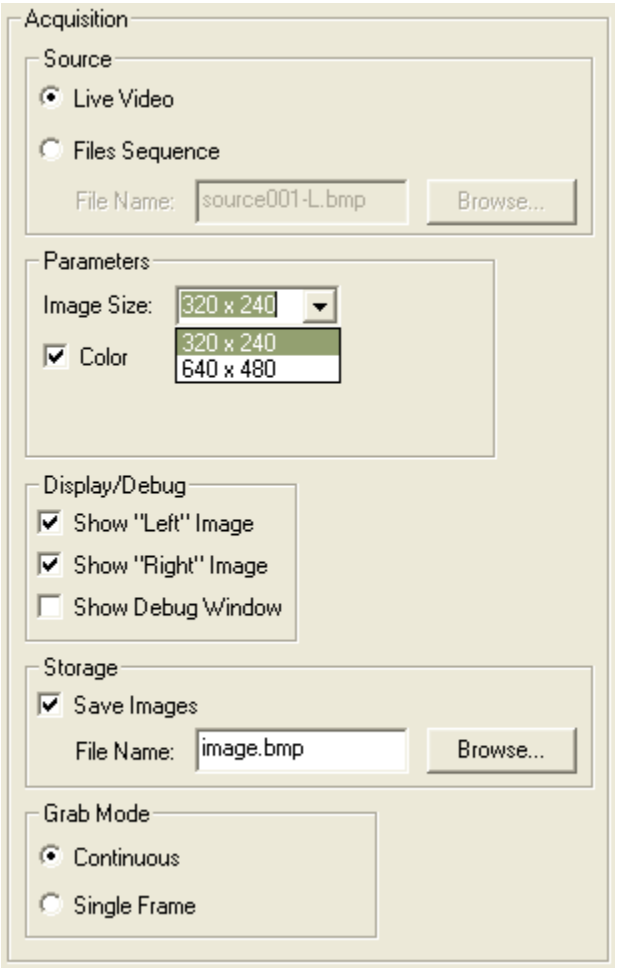
Przygotowane pliki zostają załadowane do aplikacji rozpoznającej na etapie konfiguracji.

A.4 Konfiguracja aplikacji rozpoznającej

Przygotowana aplikacja rozpoznająca ma charakter badawczy, dlatego pozostawiono w niej wszystkie ustawienia z wersji prototypowej pozwalające na zadawanie różnych wariantów przetwarzania sekwencji wizyjnych. Parametry konfiguracyjne podzielono na 7 grup: *Acquisition*, *Thresholding*, *Color*, *Morphology*, *Stereo*, *Features* i

Recognition.

Okienko *Acquisition* grupuje parametry sterujące procesem pozyskiwania obrazów (rys. A.3). Możliwa jest praca w trybie on-line z wykorzystaniem obrazów z

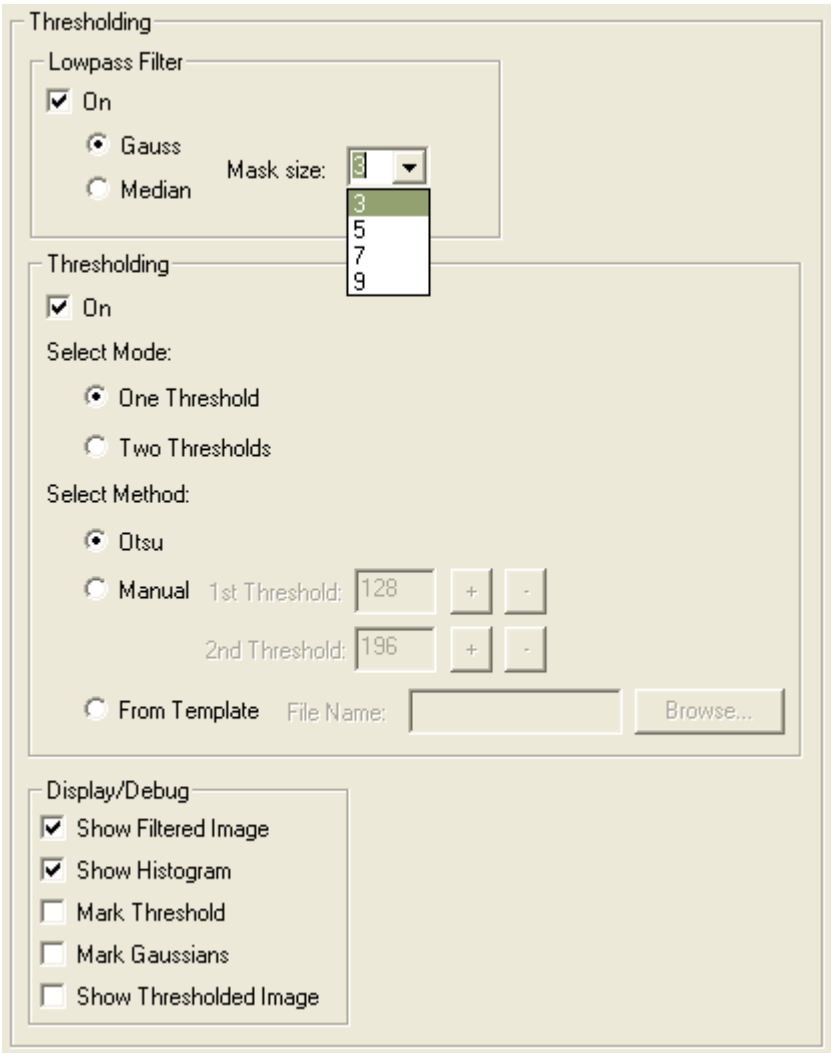


Rys. A.3. *Parametry konfiguracyjne z grupy Acquisition.*

kamery Videre-Design (przycisk opcji *Live Video*) albo praca offline z sekwencjami zapisanymi uprzednio na dysku (przycisk opcji *File Sequence*, pole edycyjne *File Name* i przycisk *Browse*). Dopuszczalne są dwie rozdzielczości obrazów: 320×240 lub 640×480 (lista rozwijana *Image Size*). Możliwe jest włączenie/wyłączenie przetwarzania sekwencji kolorowych (przycisk wyboru *Color*). Dla celów diagnostycznych możliwe jest wyświetlanie w trakcie pracy systemu obrazu z kamery lewej (pole wyboru *Show Left Image*), z kamery prawej (*Show Right Image*) oraz dodatkowego okienka z informacją tekstową (*Show Debug Window*). Obrazy mogą zostać zapisane na dysku (przycisk wyboru *Save Images*) w folderze wskazanym przez użytkownika (pole edycyjne *File Name* i przycisk *Browse*). Akwizycja obrazów może odbywać się w dwóch trybach. W pierwszym z nich w odpowiedzi na żądanie użytkownika pobierana jest ciągła sekwencja obrazów z częstotliwością 25 ramek na sekundę (przycisk opcji *Continuous*). W drugim trybie pobierana jest tylko jedna klatka z kamery

lewej i prawej (przycisk opcji *Single Frame*). Tryb ten może być wykorzystywany do nagrywania pojedynczych par stereo do kalibracji układu bądź budowania modelu chrominancji skóry ludzkiej.

Okienko *Thresholding* grupuje parametry sterujące przetwarzaniem obrazów szarych (rys. A.4). Dostępne tu kontrolki stają się aktywne, gdy użytkownik odznaczy



Rys. A.4. *Parametry konfiguracyjne z grupy Thresholding.*

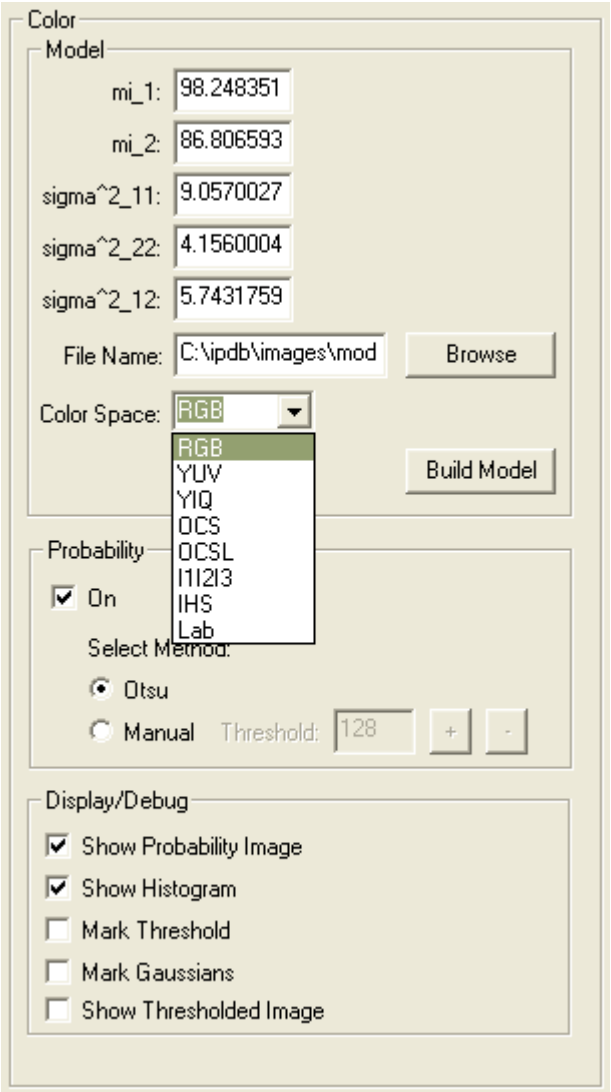
opcję *Color* na okienku *Acquisition*. W pewnych warunkach, gdy jasność twarzy i dłoni różni się wyraźnie od pozostałych obiektów, możliwe jest zastosowanie techniki progowania obrazów z poziomami szarości. Przed progowaniem obrazy mogą zostać poddane filtracji dolnoprzepustowej usuwającej szum (przycisk wyboru *Lowpass Filter*). Dostępne są dwie metody: z wykorzystaniem filtru Gaussa (pole wyboru *Gauss*) lub filtracji medianowej (*Median*). W obu przypadkach można ustawić rozmiary masek od 3 do 9 (lista rozwijana *Mask Size*). W przypadku, gdy twarz i dłonie są wyraźnie jaśniejsze od pozostałych obiektów, możliwa jest filtracja z wykorzystaniem jednej wartości progowej (pole opcji *One Threshold*). Jeżeli w obrazie

znajdują się obiekty o jasnościach zarówno mniejszych jak i większych od jasności twarzy i dłoni, można spróbować binaryzacji z dwiema wartościami progowymi (pole opcji *Two Thresholds*). W trybie z jedną wartością progową można użyć metody automatycznego doboru progu na podstawie histogramu obrazu i algorytmu Otsu [53] (pole opcji *Otsu*) lub ustawić próg ręcznie (pole opcji *Manual*, pole edycyjne *1st Threshold* i przyciski +, -). Korekta wartości progowej możliwa jest także po uruchomieniu przetwarzania danej sekwencji. W trybie z dwiema wartościami progowymi możliwy jest ich wybór ręczny (pole opcji *Manual* i pola edycyjne *1st Threshold*, *2nd Threshold* i przyciski +, -) albo ustalenie ich na podstawie wczytanego obrazu zawierającego tylko piksele dłoni i skóry zarejestrowanego w podobnych warunkach oświetlenia (pole opcji *From Template*, pole edycyjne *File Name* i przycisk *Browse*). W trakcie pracy systemu można wyświetlać dla celów diagnostycznych: obraz po filtracji dolnoprzepustowej (pole wyboru *Show Filtered Image*), histogram (*Show Histogram*), linie znaczącą wartość progową na histogramie (*Mark Threshold*), krzywe Gaussa aproksymujące histogram (*Mark Gaussians*) i obraz binarny (*Show Thresholded Image*).

Okienko *Color* grupuje parametry sterujące przetwarzaniem obrazów kolorowych (rys. A.5). Okienko to staje się aktywne, gdy użytkownik zaznaczy opcję *Color* na okienku *Acquisition*. Możliwe jest przetwarzanie obrazów kolorowych w jednej z 8 wybranych przestrzeni barw (lista rozwijana *Color Space*). Parametry modelu chrominancji skóry ludzkiej mogą zostać wprowadzone bezpośrednio (pola edycyjne *mi_1*, *mi_2*, *sigma2_11*, *sigma2_22*, *sigma2_12*) lub wyznaczone na podstawie wczytanego obrazu (pole edycyjne *File Name* i przyciski *Browse*, *Build Model*). Otrzymywane obrazy prawdopodobieństwa mogą być progowane z wykorzystaniem adapcyjnego doboru progu metodą Otsu (pole opcji *Otsu*) lub wartość progu może stać ustalona ręcznie (pole opcji *Manual*, pole edycyjne *Threshold* i przycisk +, -). W trakcie przetwarzania możliwe jest wyświetlanie dla celów diagnostycznych: obrazu prawdopodobieństwa (pole wybory *Show Probability Image*), jego histogramu (*Show Histogram*), wartości progu na obrazie histogramu (*Mark Threshold*), krzywych Gaussa aproksymujących histogram (*Mark Gaussians*) i otrzymanego obrazu binarnego (*Show Thresholded Image*).

Obrazy binarne otrzymane w wyniku przetwarzania sekwencji kolorowych albo sekwencji z odcieniami szarości mogą zostać poddane operacjom morfologicznym w celu poprawy ich jakości (rys. A.6). Można zastosować jedną z 6 operacji morfologicznych: erozję (pole opcji *Erosion*), dylatację (*Dilation*), otwarcie (*Openinig*), zamknięcie (*Closing*), filtrację OC (*OC*) i CO (*CO*). Dla celów diagnostycznych można wyświetlać otrzymywane obrazy w trakcie pracy systemu (opcja wyboru *Show Morph Image*).

Okienko *Stereo* grupuje parametry sterujące przetwarzaniem obrazów stereo (rys. A.7). Do rektyfikacji obrazów stereo wykorzystywane są parametry wyznaczone podczas kalibracji układu kamer. Do ich wczytania służy pole edycyjne *File Name* i przycisk *Browse*. Proces rektyfikacji włącza się za pomocą przycisku wyboru *Rectification*. Możliwe jest podanie maksymalnej wartości dysparycji (pole edycyjne *Max Disparity*), współczynnika wykorzystywanego do usuwania błędnych wartości dysparycji dla obszarów o ubogiej teksturze (pole edycyjne *Confidence*) i rozmiaru okienka

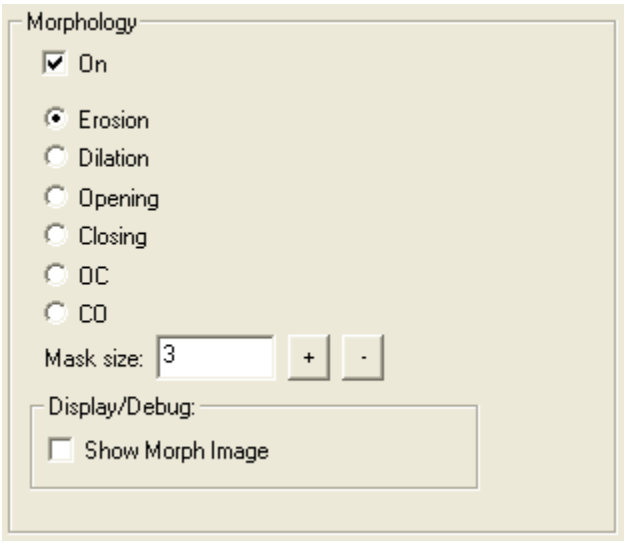


Rys. A.5. *Parametry konfiguracyjne z grupy Color.*

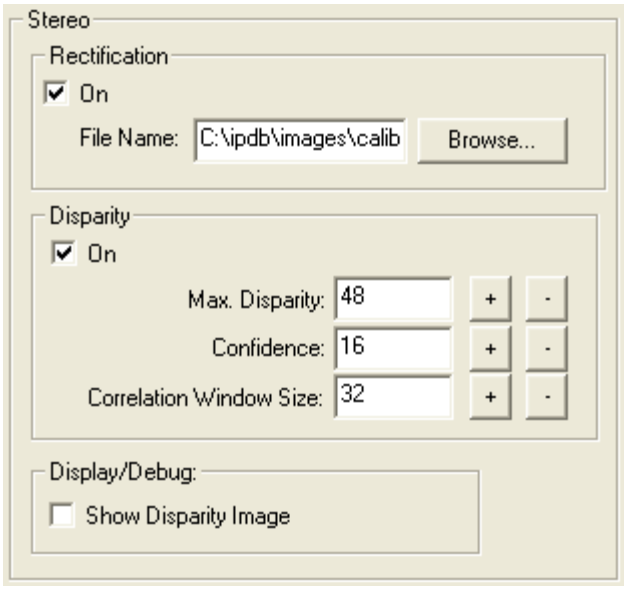
korelacji (*Correlation Window Size*). Dla celów diagnostycznych można wyświetlić podczas pracy systemu otrzymywane obrazy dysparycji (pole wyboru *Show Disparity Image*).

Okienko *Feature* grupuje parametry związane z procedurą wyznaczania wektorów cech (rys. A.8). W polu edycyjnym *Min object area* wprowadza się minimalny rozmiar obiektu. Wszystkie otrzymane w wyniku etykietowania obrazu binarnego obiekty o polach powierzchni mniejszych od tej wartości zostaną odrzucone. W trakcie działania programu możliwe jest wyświetlenie obrazu binarnego po etykietowaniu (pole wyboru *Show Labeled Image*), obrazu oryginalnego z obszarami twarzy, dłoni prawej i lewej zaznaczonymi odpowiednio za pomocą kolorów zielonego, czerwonego i niebieskiego (pole wyboru *Hands and Face*) i przebiegów wybranych wektorów cech (pole wyboru *Show Features*).

Okienko *Recognition* zawiera parametry dotyczące klasyfikacji z wykorzystaniem

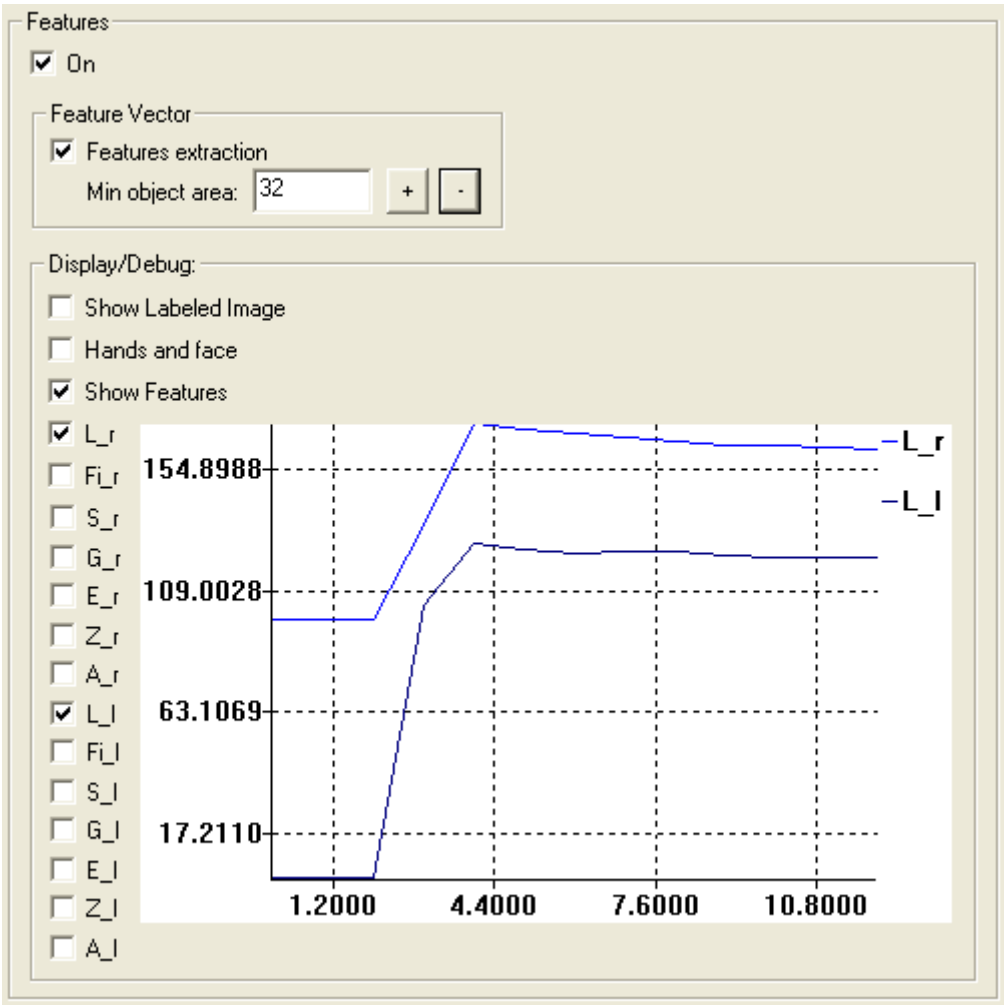


Rys. A.6. *Parametry konfiguracyjne z grupy Morphology.*

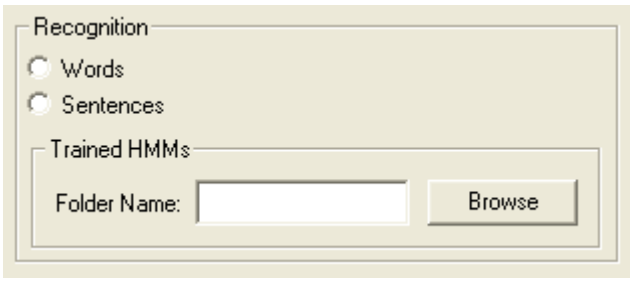


Rys. A.7. *Parametry konfiguracyjne z grupy Stereo.*

ukrytych modeli Markowa (rys. A.9). Możliwe jest ustalenie, czy rozpoznawane będą pojedyncze wyrazy (pole opcji *Words*), czy też całe sekwencje (*Sentences*). Pole edycyjne *Folder Name* oraz przycisk *Browse* pozwalają na wczytanie wyuczonych ukrytych modeli Markowa i plików pomocniczych niezbędnych do rozpoznawania, z wykorzystaniem algorytmu Viterbiego, ze wskazanego folderu (patrz dodatek D). Pozostałe parametry sterujące procesem klasyfikacji (liczba stanów emitujących, liczba rozkładów Gaussa aproksymujących obserwację, obecność i rodzaj modeli przejść) ustalane są w plikach zawierających definicję ukrytych modeli Markowa, bądź w plikach pomocniczych przygotowywanych w trakcie uczenia modeli.



Rys. A.8. *Parametry konfiguracyjne z grupy Features.*



Rys. A.9. *Parametry konfiguracyjne z grupy Recognition.*

A.5 Rozpoznawanie

Po wykonaniu całej sekwencji następuje jej rozpoznanie z wykorzystaniem przygotowanych uprzednio wyuczonych modeli Markowa. Jako rezultat wyświetlona zostaje transkrypcja rozpoznanej sekwencji w formie wykorzystywanej w wariancie

użytkowym PJM oraz jej pełny zapis z końcówkami fleksyjnymi. Ponadto w wyniku wywołania procedur z pakietu HTK na dysku tworzony jest plik wynikowy zawierający dodatkowe statystyki. W tab. A.1 przedstawiono czasy osiągane przy przetwarzaniu pojedynczej pary stereo dla różnych wariantów przetwarzania sekwencji wizyjnej.

Tab. A.1. *Czasy przetwarzania pary obrazów stereo dla różnych wariantów przetwarzania sekwencji wizyjnej*

wariant przetwarzania sekwencji wizyjnej	czas przetwarzania [ms]
obrazy szare, binaryzacja z ręcznym doborem progu, etykietowanie, wyznaczanie mapy dysparycji	18
obrazy szare, filtracja Gaussa, binaryzacja metodą Otsu, filtracja OC, etykietowanie, wyznaczanie mapy dysparycji	29
obrazy kolorowe, progowanie obrazu prawdopodobieństwa, etykietowanie, wyznaczanie mapy dysparycji	25
obrazy kolorowe, progowanie obrazu prawdopodobieństwa, filtracja OC, etykietowanie, wyznaczanie mapy dysparycji	36

Dodatek B

Biblioteka funkcji przetwarzania obrazów

W trakcie przeprowadzania prac badawczych konieczne było tworzenie różnorodnych aplikacji testowych z zakresu przetwarzania obrazów. Dlatego opracowano bibliotekę podstawowych funkcji przetwarzania obrazów. Podczas tworzenia tej biblioteki zwracano szczególną uwagę na takie jej cechy jak szybkość przetwarzania i obliczeń, prostota użycia, spójny i intuicyjny interfejs oraz skalowalność i przenośność. Aby maksymalnie ułatwić tworzenie aplikacji z wykorzystaniem tej biblioteki przyjęto jednolity standard kodowania i dokumentowania kodu. Biblioteka napisana została w języku C i tworzące ją moduły, z wyjątkiem kilku specyficznych dla systemu Windows, mogą być łatwo przeniesione na inne platformy. Krótką charakterystykę modułów biblioteki przedstawiono w tab. B.1

Tab. B.1. *Moduły biblioteki przetwarzania obrazów*

lp.	nazwa modułu	przeznaczenie
		operacje wejścia/wyjścia
1	avi	Przetwarzanie plików filmowych zapisanych w formacie AVI.
2	bmp	Odczyt i zapis z/do pliku BMP.
3	display	Wyświetlanie obrazów w trybach VESA.
4	img2scr	Wyświetlanie obrazów w aplikacjach działających w Windows.
5	meteor	Akwizycja obrazów z kart frame-grabberów firmy Matrox (Meteor II, Morphis)
6	vesa	Obsługa trybów graficznych VESA.
7	video	Akwizycja obrazów z wykorzystaniem standardu Video for Windows (kamery analogowe, kamery internetowe).
		ogólne
8	errors	Moduł obsługi błędów.
9	image	Reprezentacja obrazów różnych typów w pamięci komputera.
10	stoper	Funkcje ułatwiające pomiar czasu.

lp.	nazwa modułu	przeznaczenie
		Przetwarzanie obrazów monochromatycznych i binarnych
11	and	Operacja AND na obrazach binarnych.
12	area	Obliczenie pola powierzchni obiektu o danej jasności.
13	axis	Wyznaczenie kąta nachylenia osi głównej obiektu w obrazie binarnym.
14	bright	Korekcja jasności.
15	centroid	Obliczenie środka ciężkości obiektu o danej jasności.
16	circum	Obliczenie obwodu obiektu o danej jasności.
17	cluster	Grupowanie pikseli.
18	contrast	Korekcja kontrastu.
19	conv	Konwolucja dyskretna obrazów (otrzymane wartości ujemne są zastępowane zerami).
20	conv1	Konwolucja dyskretna obrazów (otrzymane wartości ujemne są zastępowane wartościami bezwzględnymi).
21	conv2	Konwolucja dyskretna obrazów.
22	correl	Dyskretna korelacja dla obrazów z odcieniami szarości.
23	correlb	Dyskretna korelacja dla obrazów binarnych.
24	differ	Różnica obrazów (otrzymane wartości ujemne są zastępowane zerami).
25	differ1	Różnica obrazów (otrzymane wartości ujemne są zastępowane wartościami bezwzględnymi).
26	dilatio1	Dylatacja obrazu binarnego - wersja zoptymalizowana dla elementów strukturalnych o dużych powierzchniach, ale zawierających małą liczbę białych pikseli.
27	dilation1	Dylatacja obrazu binarnego.
28	equalize	Wyrównywanie histogramu.
29	erosion	Erozja obrazu binarnego.
30	erosion1	Erozja obrazu binarnego - wersja zoptymalizowana dla elementów strukturalnych o dużych powierzchniach ale zawierających małą liczbę białych pikseli.
31	filters	Definicja najczęściej używanych filtrów.
32	findth	Adaptacyjne ustalenie wartości progu binaryzacji na podstawie aproksymacji histogramu krzywymi Gaussa.
33	findth1	Adaptacyjne ustalenie wartości progu binaryzacji na podstawie histogramu.
34	gauss	Filtracja Gaussa za pomocą maski 3 x 3.
35	hist	Wyznaczenie histogramu obrazu z poziomami szarości.
36	hitmiss	Operacja morfologiczna *hit or miss* dla obrazów binarnych.
37	hitmiss1	Operacja morfologiczna dla obrazów binarnych - wersja zoptymalizowana dla elementów strukturalnych o dużych powierzchniach zawierających małą liczbę białych pikseli.
38	hough	Transformacja Hougha do wykrywania linii prostych.
39	label	Segmentacja za pomocą algorytmu liniowego przeglądania obrazu i sklejeń.
40	laplace	Filtracja Laplace'a obrazów.
41	median	Filtracja medianowa obrazów.
42	momgeo	Wyznaczanie momentów geometrycznych.
43	negative	Negatyw obrazu z poziomami szarości.
44	nonlin	Nieliniowa transformacja poziomów szarości.
45	rotate	Obrót obrazu o zadany kąt.
46	saltpep	Dodanie do obrazu szumu typu sól i pieprz.
47	thresh	Binaryzacja obrazu.

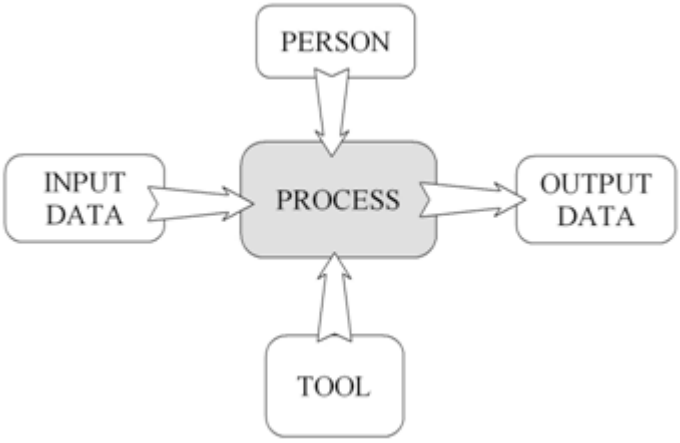
lp.	nazwa modułu	przeznaczenie
		Przetwarzanie obrazów kolorowych
48	colnorm	Normalizacja przestrzeni barw RGB.
49	colnorm1	Normalizacja przestrzeni barw RGB z odrzuceniem pikseli o małej jasności.
50	getrgb	Wyodrębnienie obrazów R, G i B z obrazu kolorowego RGB.
51	gray2rgb	Przekształcenie obrazu z poziomami szarości do obrazu kolorowego.
52	model	Wyznaczanie modelu rozkładu chrominancji.
53	pln2rgb	Przekształcenie obrazów typu PLANAR (Matrox) do obrazów RGB.
54	rgb2	Liniowe przekształcenie przestrzeni barw.
55	rgb2gray	Przekształcenie obrazu kolorowego do obrazu z poziomami szarości.
56	rgb2i123	Zmiana przestrzeni barw z RGB na I1I2I3.
57	rgb2ihs	Zmiana przestrzeni barw z RGB na IHS.
58	rgb2lab	Zmiana przestrzeni barw z RGB na L*a*b*.
59	rgb2ocs	Zmiana przestrzeni barw z RGB na Opponent Colour Space.
60	rgb2ocsl	Zmiana przestrzeni barw z RGB na Opponent Colour Space - wersja logarytmiczna [18].
61	rgb2xyz	Zmiana przestrzeni barw z RGB na XYZ.
62	rgb2yiq	Zmiana przestrzeni barw z RGB na YIQ.
63	rgb2yuv	Zmiana przestrzeni barw z RGB na YUV.
64	satur	Zmiana nasycenia barwy w obrazie kolorowym.
65	setrgb	Złożenie obrazu kolorowego RGB z obrazów R, G i B.
66	tint	Korekcja barwy dla obrazu kolorowego.
		Funkcje pomocnicze
67	alscal	Wyznaczenie prawdopodobieństwa P(y—M) dla dyskretnego ciągu obserwacji y i ukrytego modelu Markowa M metodą w przód.
68	arctan	Wyznaczenie arctg2.
69	combine	Wyznaczenie obrazu na podstawie dwóch innych obrazów wg reguły określonej przez parametr (będący wskaźnikiem na funkcje).
70	combine1	Jak wyżej, ale dla obrazów, w których do zapisu pikseli wykorzystano dwa bajty.
71	det	Obliczenie wyznacznika macierzy.
72	drawrec	Narysowanie prostokąta w obrazie.
73	extract	Wycinanie prostokątnego okienka z obrazu.
74	gau2img	Narysowanie krzywych Gaussa na obrazie histogramu.
75	gauss2D	Wyznaczenie wartości dwuwymiarowej funkcji Gaussa.
76	h2d2img	Przedstawienie histogramu 2D za pomocą obrazu.
77	hflip	Poziome odbicie obrazu.
78	hist2d	Wyznaczenie histogramu 2D na podstawie dwóch obrazów z wybranymi składowymi barw.
79	hist2img	Wyświetlenie histogramu.
80	insert	Wstawienie okna do obrazu.
81	int2bas	Konwersja obrazu do formatu, w którym do zapisu jednego piksela wykorzystano 2 bajty
82	rotate	Obrót obrazu.
83	scale	Przeskalowanie obrazu.
84	sort	Sortowanie bąbelkowe w porządku malejącym.
85	sort1	Sortowanie bąbelkowe w porządku malejącym dla danych zapisanych z wykorzystaniem jednego bajta
86	stereo	Wyznaczanie zwartych map dysparycji dla różnych miar dopasowania.
87	egraph	Dopasowywanie elastycznego grafu na podstawie parametrów węzłów, wyznaczonych wg zadanej metody (np. średnia w otoczeniu lub wynik działania filtrów Gabora)

Dodatek C

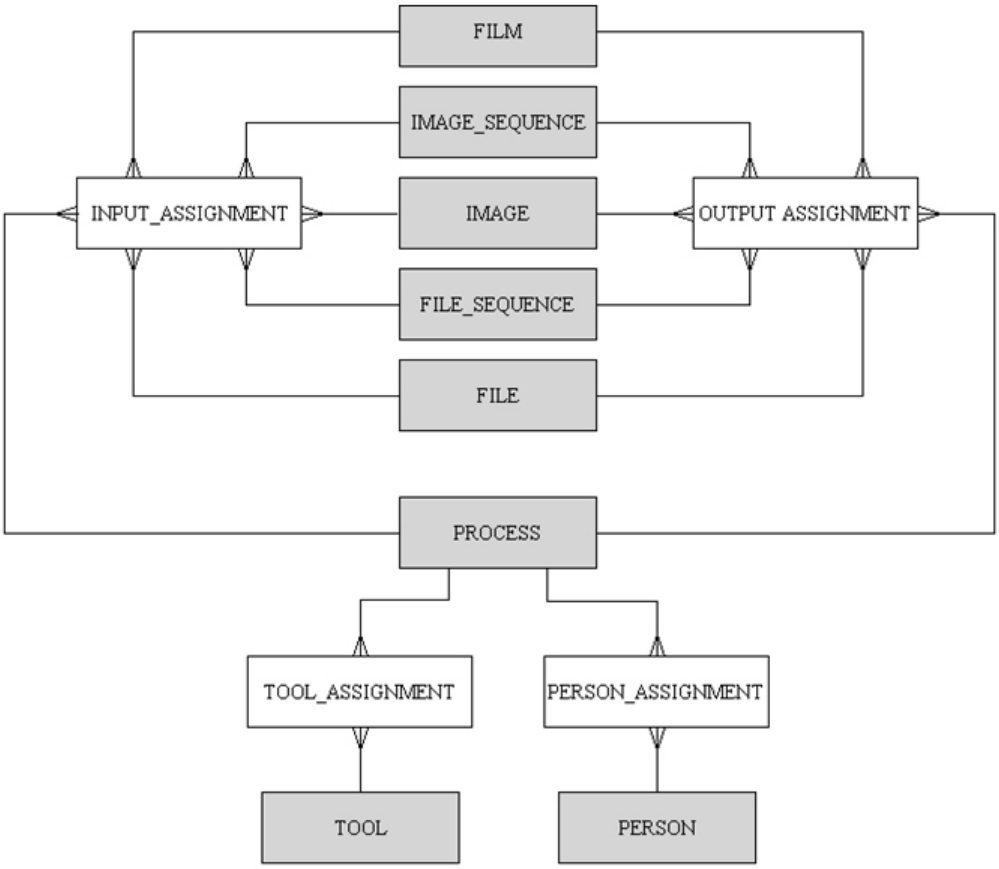
Aplikacja bazy danych

Eksperymenty z zakresu rozpoznawania wyrazów i słów PJM w układzie wizyjnym wymagają pracy z dużą ilością danych o charakterze multimedialnym, przy wykorzystaniu różnorodnych narzędzi sprzętowych i programów obliczeniowych, których interfejsy wejścia/wyjścia nie zawsze są ze sobą zgodne. Badawczy charakter prac sprawia, że w ich trakcie generowana jest spora liczba plików pośrednich i rozmaitych statystyk pomocniczych, które muszą być archiwizowane na wypadek, gdy okażą się jeszcze przydatne. Próby uporządkowywania oparte jedynie na wprowadzeniu hierarchicznego układu folderów i sformalizowanego systemu nazewnictwa plików mogą być nieefektywne. Dlatego na użytek opisywanych badań konieczne stało się opracowanie narzędzi, które uczynią z heterogenicznego zestawu instrumentów spójne i łatwe w użyciu środowisko badawcze.

Elementem umożliwiającym zachowanie integralności całego środowiska badawczego jest specjalnie zaprojektowana baza danych, która służy nie tylko do archiwizacji danych, ale pozwala także na zarejestrowanie narzędzi, za pomocą których dane zostały uzyskane, ustawień i parametrów konfiguracyjnych, przy których przeprowadzano dany eksperyment, kolejności w jakiej poszczególne pliki były generowane, celu przeprowadzania danego testu oraz dowolnych dodatkowych informacji wpisywanych przez użytkownika, gdy zajdzie taka potrzeba. Aby to osiągnąć, przyjęto podczas projektowania struktury bazy założenie, że jej podstawową funkcją będzie raczej rejestrowanie aktywności użytkownika wykonującego eksperyment, odpowiednie dane będą zaś wpisywane jako argumenty wejściowe bądź rezultaty wspomnianej aktywności. Takie podejście umożliwia odtworzenie całej historii eksperymentu na podstawie wpisów w bazie. Na rys. C.1 przedstawiono logiczną strukturę bazy danych. W tabeli PROCESS zapisywane są kolejne akcje podejmowane podczas przeprowadzania danego eksperymentu. Proces wykonywany jest przez osobę, której dane zapisane są w tabeli PERSON z wykorzystaniem narzędzi opisanych w tabeli TOOL. Proces może wymagać pewnych danych wejściowych, oznaczonych na rysunku INPUT DATA i może generować rezultaty oznaczone jako OUTPUT DATA. Danymi dla poszczególnych procesów mogą być filmy, sekwencje obrazów, pojedyncze obrazy, sekwencje plików i pojedyncze pliki. Wyniki wykonania danego procesu mogą być jednocześnie danymi wejściowymi dla innego procesu. Strukturę bazy danych pokazano na rys. C.2. Przy wyborze zestawu atrybutów opisujących daną



Rys. C.1. *Logiczna struktura bazy danych*



Rys. C.2. *Struktura bazy danych*

encję przyjęto, że projektowana baza danych będzie miała charakter uniwersalny i w przyszłości będzie mogła także być zastosowana do wspierania innych badań związanych z przetwarzaniem dużej ilości plików multimedialnych. W tabelach C.1-C.12 zamieszczono zestawy atrybutów opisujące poszczególne encje. Do budowy

systemu obsługującego bazę zastosowano MySQL. Interfejs użytkownika opracowano z wykorzystaniem PHP.

Tab. C.1. *Opis encji PROCESS*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator procesu.
2	name	Nazwa procesu.
3	execution_date	Data wykonania procesu.
4	short_description	Krótki opis procesu.
5	long_description	Długi opis procesu.

Tab. C.2. *Opis encji PERSON*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator osoby.
2	name	Imię i nazwisko.
3	function	Funkcja (admin, user, guess).
4	login	Identyfikator używany podczas logowania.
5	password	Hasło używane podczas logowania.
6	short_description	Krótki opis osoby.
7	long_description	Długi opis osoby.

Tab. C.3. *Opis encji TOOL*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator narzędzia.
2	name	Nazwa narzędzia.
3	path	Ścieżka dostępu do pliku.
4	launch_method	Sposób wywołania narzędzia.
5	config_file	Ścieżka dostępu do pliku konfiguracyjnego.
6	version	Numer wersji narzędzia.
7	short_description	Krótki opis narzędzia.
8	long_description	Długi opis narzędzia.

Tab. C.4. *Opis encji FILM*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator filmu.
2	nazwa	Nazwa filmu.
3	path	Ścieżka dostępu do pliku.
4	format	Format pliku (avi, mpeg, mov, itp.).
5	width	Szerokość klatki.
6	height	Wysokość klatki.
7	bits_per_pixel	Liczba bitów na jeden piksel.
8	codec	Kompresja zastosowana przy zapisie filmu.
9	short_description	Krótki opis filmu.
10	long_description	Długi opis filmu.

Tab. C.5. *Opis encji IMAGE_SEQUENCE*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator sekwencji obrazów.
2	name	Nazwa sekwencji obrazów.
3	path	Ścieżka dostępu do folderu zawierającego sekwencję obrazów.
4	basename	Nazwa bazowa pliku.
5	start_index	Numer pierwszej klatki w sekwencji.
6	stop_index	Numer ostatniej klatki w sekwencji.
7	format	Format pojedynczego obrazu (bmp, jpeg, gif, itp.).
8	width	Szerokość pojedynczego obrazu.
9	height	Wysokość pojedynczego obrazu.
10	bits_per_pixel	Liczba bitów na jeden piksel.
11	short_description	Krótki opis sekwencji obrazów.
12	long_description	Długi opis sekwencji obrazów.

Tab. C.6. *Opis encji IMAGE*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator obrazu.
2	nazwa	Nazwa obrazu.
3	path	Ścieżka dostępu do pliku.
4	format	Format obrazu (bmp, jpeg, gif, itp.).
5	width	Szerokość obrazu.
6	height	Wysokość obrazu.
7	bits_per_pixel	Liczba bitów na jeden piksel.
8	short_description	Krótki opis obrazu.
9	long_description	Długi opis obrazu.

Tab. C.7. *Opis encji FILE_SEQUENCE*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator sekwencji plików.
2	name	Nazwa sekwencji plików.
3	path	Ścieżka dostępu do folderu zawierającego sekwencję plików.
4	basename	Nazwa bazowa pliku.
5	start_index	Numer pierwszego pliku w sekwencji.
6	stop_index	Numer ostatniego pliku w sekwencji.
7	format	Format pojedynczego pliku (doc, xls, mfc, itp.).
8	length	Rozmiar pliku.
9	short_description	Krótki opis sekwencji plików.
10	long_description	Długi opis sekwencji plików.

Tab. C.8. *Opis encji FILE*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator pliku.
2	name	Nazwa pliku.
3	path	Ścieżka dostępu do pliku.
4	format	Format pliku (doc, xls, mfc, itp.).
5	length	Rozmiar pliku.
6	short_description	Krótki opis pliku.
7	long_description	Długi opis pliku.

Tab. C.9. *Opis encji PERSON_ASSIGNMENT*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator przyporządkowania.
2	process_id	Identyfikator procesu.
3	person_id	Identyfikator osoby.

Tab. C.10. *Opis encji TOOL_ASSIGNMENT*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator przyporządkowania.
2	process_id	Identyfikator procesu.
3	tool_id	Identyfikator narzędzia.

Tab. C.11. *Opis encji INPUT_ASSIGNMENT*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator przyporządkowania.
2	process_id	Identyfikator procesu.
3	image_id	Identyfikator obrazu.
4	file_id	Identyfikator pliku.
5	image_sequence_id	Identyfikator sekwencji obrazów.
6	file_sequence_id	Identyfikator sekwencji plików.
7	film_id	Identyfikator filmu.

Tab. C.12. *Opis encji OUTPUT_ASSIGNMENT*

lp.	nazwa pola	przeznaczenie
1	id	Unikatowy identyfikator przyporządkowania.
2	process_id	Identyfikator procesu.
3	image_id	Identyfikator obrazu.
4	file_id	Identyfikator pliku.
5	image_sequence_id	Identyfikator sekwencji obrazów.
6	file_sequence_id	Identyfikator sekwencji plików.
7	film_id	Identyfikator filmu.

Dodatek D

Przewodnik użytkownika HTK

Hidden Markov Model Toolkit (HTK) jest narzędziem do budowania systemów rozpoznawania wykorzystujących ukryte modele Markowa. Narzędzie to zostało opracowane w zespole Speech, Vision and Robotics na Uniwersytecie Cambridge i pierwotnie było przeznaczone do rozpoznawania mowy. Jednak jądro HTK jest ogólnego przeznaczenia i budowane z jego wykorzystaniem ukryte modele Markowa mogą być wykorzystane do modelowania dowolnych przebiegów czasowych. Dlatego narzędzie to stosowane jest także do syntezy mowy, rozpoznawania sekwencji DNA, rozpoznawania pisma i rozpoznawania gestów.

HTK jest zbiorem modułów bibliotecznych i programów narzędziowych wspierających rejestrowanie i przetwarzanie sygnału mowy, konstruowanie złożonych układów HMM, uczenie, testowanie i analizę otrzymanych rezultatów. Zastosowanie HTK do rozpoznawania gestów wymaga jedynie zastąpienia modułów dedykowanych do przetwarzania sygnału mowy własnymi, które zapiszą dane i konfigurację układu rozpoznającego w formacie przyjętym w HTK. Dla każdego z programów narzędziowych dostępne są kody źródłowe w języku C. Pozwala to na zastosowanie algorytmów wbudowanych w HTK we własnych programach.

HTK umożliwia budowanie ukrytych modeli Markowa z obserwacją ciągłą i dyskretną. W przypadku modeli z obserwacją ciągłą funkcja gęstości prawdopodobieństwa obserwacji ma postać sumy rozkładów Gaussa, które mogą być opisywane za pomocą pełnej macierzy kowariancji bądź wektorów z wartościami wariancji, gdy poszczególne składowe wektora cech są statystycznie niezależne. W każdym modelu stany pierwszy i ostatni są nieemitujące, tzn. nie generują obserwacji. Stany te wykorzystywane są do łączenia poszczególnych modeli w większe struktury. Strukturę ukrytego modelu Markowa zadaje się wprowadzając do macierzy tranzycji niezerowe wartości początkowe dla dozwolonych przejść pomiędzy stanami. Umożliwia to projektowanie modeli o dowolnych topologiach. Poprzez wprowadzenie przejść jednokierunkowych z możliwością pominięcia pewnych stanów otrzymuje się tzw. modele Bakisa, nadające się do rozpoznawania gestów, które mogą być wykonywane z różną szybkością. Możliwe jest także budowanie tzw. modeli typu 'Tee' (zob. podrozdział 4.3 i rys. 4.7b). Modele te mają dodatkowe przejście pomiędzy pierwszym i ostatnim stanem nieemitującym i wykorzystywane są do modelowania opcjonalnych fragmentów sekwencji czasowych. W przypadku rozpoznawania

sekwencji gestów modele typu 'Tee' można wykorzystać do modelowania przejść pomiędzy poszczególnymi gestami.

Uczenie ukrytych modeli Markowa z wykorzystaniem HTK odbywa się w dwóch etapach. Na początku dokonywana jest wstępna estymacja parametrów modelu z wykorzystaniem algorytmu Viterbiego, następnie zaś douczanie z wykorzystaniem algorytmu Bauma-Welcha. Do budowania systemów rozpoznawania sekwencji przebiegów czasowych, z których każdy modelowany jest za pomocą jednego HMM zaimplementowano uczenie z wbudowanym wyodrębnianiem elementów składowych w ciągu uczącym, a więc np. wyrazów w zdaniach, tzw. *embedded training*. *Embedded training* można zastosować do modelowania sekwencji gestów. W tym przypadku po wstępnej estymacji poszczególne modele uczone są równolegle z wykorzystaniem całych sekwencji, przy czym nie jest wymagana dokładna segmentacja danej sekwencji, a jedynie informacja o kolejności gestów, z których jest ona zbudowana. Taka strategia uczenia ułatwia proces gromadzenia danych uczących, co ma duże znaczenie zwłaszcza w przypadkach, gdy zbiór rozpoznawanych sekwencji jest liczny. W HTK rozpoznawanie odbywa się z wykorzystaniem algorytmu Viterbiego w wersji z przekazywaniem znaczników (zob. podrozdział 4.3).

W przypadku rozpoznawania sekwencji gestów strukturę sieci ukrytych modeli Markowa zadaje się za pomocą specjalnego języka, który składa się ze zbioru definicji i następującego po nich wyrażenia regularnego opisującego układ modeli. Tak zapisana sieć modeli jest następnie automatycznie przekształcana do formatu HTK Standard Lattice Format, który jest wykorzystywany w trakcie uczenia i rozpoznawania. Umożliwia to zapisanie w prosty i przejrzysty sposób nawet bardzo złożonych konfiguracji ukrytych modeli Markowa.

D.1 Zdefiniowanie problemu

Pierwszym etapem podczas tworzenia systemu rozpoznającego z wykorzystaniem pakietu HTK jest zdefiniowanie problemu. W tym celu wykorzystuje się specjalny język i tworzy się plik tekstowy z opisem tzw. gramatyki zadania. W ogólnym przypadku opis ten składa się z zestawu definicji i następującego po nich wyrażenia regularnego. Wyrażenie regularne zapisuje się w nawiasach zwykłych (,) z wykorzystaniem specjalnych metaznaków, których znaczenie przedstawiono w tab. D.1.

Tab. D.1. *Metaznaki używane w opisie gramatyki zadania*

metaznak	opis
|	alternatywa
[ ]	element opcjonalny
{ }	zero lub więcej powtórzeń
< >	jedno lub więcej powtórzeń
<< >>	zapis informacji o kontekście

Przykład D.1

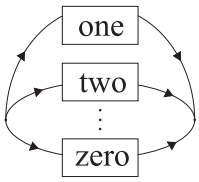
Załóżmy, że naszym zadaniem jest rozpoznawanie pojedynczych cyfr: 1, 2, ... 9, 0. Plik z opisem gramatyki zadania będzie miał postać:

```
(one|two|three|four|five|six|seven|eight|nine|zero)
```

lub z wykorzystaniem definicji:

```
$digit = one|two|three|four|five|six|seven|eight|nine|zero;
( $digit )
```

Opisowi takiemu odpowiada struktura układu rozpoznającego przedstawiona na rys. D.1.

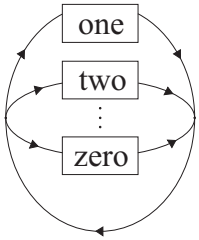


Rys. D.1. *Struktura HMM do rozpoznawania pojedynczych cyfr.*

Przykład D.2
Załóżmy teraz, że zamiast pojedynczych cyfr rozpoznawane mają być ich sekwencje. W tym przypadku należy użyć nawiasów <, > oznaczających jedno lub wiele wystąpień (patrz tab. D.1). Zatem plik z definicją gramatyki zadania będzie miał teraz postać:

```
$digit = one|two|three|four|five|six|seven|eight|nine|zero;
( < $digit > )
```

Rys. D.2 przedstawia strukturę HMM do rozpoznawania sekwencji cyfr.



Rys. D.2. *Struktura HMM do rozpoznawania sekwencji cyfr.*

Do zapisu informacji o kontekście wykorzystuje się podwójne nawiasy <<, >> i znaki '-', '+'. W nawiasach <<, >> umieszcza się wpisy rozdzielone znakiem alternatywy |. Pojedynczy wpis ma postać A-B+C, gdzie A reprezentuje słowo znajdujące się z lewej strony wyrazu, a C słowo ze strony prawej. Aktualny wyraz jest

oznaczony przez B. Elementy A i C są opcjonalne. Przy zapisywaniu informacji o kontekście pomija się znak $ stosowany w innych przypadkach przed nazwami zmiennych.

Przykład D.3
Załóżmy, że rozpoznajemy sekwencje złożone z dwóch cyfr, przy czym jeżeli pierwsza z nich jest nieparzysta, to druga jest parzysta i odwrotnie, jeżeli pierwsza jest parzysta, to druga nie. W takim przypadku plik z opisem gramatyki zadania może mieć postać:

```
$odd_digit = one|three|five|seven|nine;
$even_digit = two|four|six|eight|zero;
( << odd_digit+even_digit | even_digit+odd_digit >> )
```

Opis gramatyki zadania z wykorzystaniem wyrażenia regularnego jest wygodny dla użytkownika i pozwala na zapis nawet bardzo złożonych struktur. Narzędzia HTK wymagają jednak zapisu z wykorzystaniem specjalnego formatu HTK Standard Lattice Format (SLF), w którym struktura układu rozpoznającego zapisana jest za pomocą listy węzłów i listy tranzycji między nimi. Opis gramatyki zadania może zostać przekształcony do formatu SLF z wykorzystaniem wchodzącego w skład pakietu narzędzia HParse. Jeżeli założymy, że plik z przygotowanym opisem gramatyki ma nazwę `grammar` a wynikowy plik w formacie SLF ma nazywać się `word_network`, to konwersji dokonamy następująco:

```
HParse grammar word_network
```

D.2 Przygotowanie słownika

Kolejnym krokiem jest przygotowanie pliku słownika, który będzie zawierał wszystkie rozpoznawane wyrazy. Jest to plik tekstowy, w którym każda linia opisuje pojedynczy wyraz i ma postać:

```
word [output_symbol] pronunciation_probability hmm_1, hmm_2, ..., hmm_n
```

gdzie: `word` jest opisywanym wyrazem, `output_symbol` jest nazwą, która będzie używana przy wyświetlaniu wyników, `pronunciation_probability` przyjmuje wartości z przedziału (0, 1) i określa jakie jest prawdopodobieństwo, że opisywany wyraz jest modelowany za pomocą zestawu modeli o nazwach `hmm_1`, ... `hmm_n`.

W ogólnym przypadku dany wyraz może być wymawiany na wiele sposobów i wtedy wpisujemy go do słownika wielokrotnie z innym zestawem modeli i odpowiednim prawdopodobieństwem wystąpienia. To samo dotyczy gestów, które mogą przecież być wykonywane na wiele sposobów. W przypadku sygnału mowy jeden model Markowa może odpowiadać pojedynczej głosce i dlatego w opisie słowa może pojawić się więcej niż jeden HMM.

`Output_symbol` i `pronunciation_probability` są parametrami opcjonalnymi. Jeżeli `output_symbol` nie wystąpi w opisie danego wyrazu, to używana jest nazwa wpisana na początku odpowiadającej mu linii w pliku słownika. Można także użyć pustych nawiasów [] i wtedy nazwa danego wyrazu w ogóle nie pojawi się podczas wyświetlania wyników. Jest to przydatne w przypadku modeli przejść pomiędzy wyrazami, które muszą być także umieszczane w pliku słownika, ale nie powinny pojawiać się, gdy będzie wyświetlana rozpoznana sekwencja wyrazów.

W przykładach D.1-D.3 plik słownika mógłby wyglądać następująco:

```
one [jeden] hmm_1
two [dwa]  hmm_2
three [trzy] hmm_3
four [cztery] hmm_4
five [piec] hmm_5
six [szesc] hmm_6
seven [siedem] hmm_7
eight [osiem] hmm_8
nine [dziewiec] hmm_9
zero [zero] hmm_0
```

Wtedy przy wyświetlaniu wyników rozpoznawane cyfry będą wypisywane w języku polskim. Modelom Markowa odpowiadającym poszczególnym cyfrom nadano nazwy `hmm_1`, `hmm_2`, ..., `hmm_0`.

D.3 Przygotowanie modeli

Kolejnym krokiem jest przygotowanie modeli. Dla przykładów D.1-D.3 modele te muszą nazywać się `hmm_1`, `hmm_2`, ..., `hmm_0`, tak jak to zdefiniowano w pliku słownika i muszą być zapisane w plikach odpowiednio: `hmm_1.hmm`, `hmm_2.hmm`, ..., `hmm_0.hmm`. Poniżej pokazano zawartość pliku `hmm_1.hmm`, w którym zdefiniowano model HMM z czterema stanami, przy czym stan pierwszy i ostatni są nieemitujące. Jest to model z obserwacją ciągłą, w którym funkcja gęstości prawdopodobieństwa obserwacji w stanie emitującym opisana jest za pomocą rozkładu normalnego. Przyjęto, że wektor obserwacji złożony jest z czterech elementów.

```
~h "hmm_1"
<BeginHMM>
        <VecSize> 4 <USER>
        <NumStates> 4
        <State> 2
                <Mean> 4
                        1.0 1.0 1.0 1.0
                <Variance> 4
                        1.0 1.0 1.0 1.0
        <State> 3
                <Mean> 4
```

```
                        1.0 1.0 1.0 1.0
                <Variance> 4
                        1.0 1.0 1.0 1.0
        <TRANSP> 4
                0.0 0.5 0.5 0.0
                0.0 0.3 0.4 0.3
                0.0 0.0 0.5 0.5
                0.0 0.0 0.0 0.0
<EndHMM>
```

Definicja modelu zaczyna się od znaków ~ h i nazwy modelu. Następnie pojawia się znacznik <BeginHMM>, któremu odpowiada umieszczany na końcu definicji znacznik <EndHMM>. Linia <VecSize> 4 <USER> informuje, że wektor cech składa się z czterech elementów i że mogą to być dowolne parametry zdefiniowane przez użytkownika, a nie tylko wartości specyficzne dla przetwarzania sygnału mowy (znacznik <USER>). Kolejna linia <NumStates> 4 określa liczbę stanów ukrytego modelu Markowa. Po niej następują definicje emitujących stanów modelu. W przedstawionym przykładzie są to stany 2 i 3. Opis stanu rozpoczyna się od znacznika <State>, po którym podaje się numer stanu. Następnie umieszcza się opis funkcji gęstości prawdopodobieństwa obserwacji w danym stanie, która w naszym przypadku określona jest poprzez podanie wartości średnich i wariancji. Wartości średnie i wariancje wprowadza się w postaci oddzielonych spacjami liczb i poprzedza się je odpowiednio znacznikiem <Mean> lub <Variance>, po którym następuje liczba wartości. Wartości średnie i wariancje zostaną ustalone podczas uczenia, dlatego w trakcie tworzenia modelu można wpisywać dowolne, niezerowe liczby. Liczby te nie będą traktowane jako wartości początkowe podczas uczenia. Po opisaniu wszystkich stanów emitujących podaje się macierz prawdopodobieństw przejść poprzedzoną znacznikiem <TRANSP> i liczbą określającą jej rozmiar. W zapisie macierzy tranzycji stanów, element umieszczony w wierszu j i kolumnie i określa, jakie jest prawdopodobieństwo przejścia ze stanu i do j. Topologie modelu określa się poprzez wpisywanie niezerowych wartości w miejscach odpowiadających dozwolonym przejściom pomiędzy stanami. Sumy wartości w poszczególnych wierszach macierzy muszą być równe 1, z wyjątkiem ostatniego wiersza, w którym należy wpisać wartości zerowe.

W przypadku, gdy funkcja gęstości prawdopodobieństwa obserwacji w danym stanie jest sumą kilku rozkładów normalnych, opis stanu jest bardziej złożony. Poniżej zamieszczono przykład opisu stanu 2, w którym funkcja gęstości prawdopodobieństwa obserwacji jest sumą dwóch rozkładów normalnych, pierwszego z wagą 0.4 a drugiego z wagą 0.6.

```
<State> 2 <NumMixes> 2
        <Mixture> 1 0.4
                <Mean> 4
                        1.0 1.0 1.0 1.0
                <Variance> 4
                        1.0 1.0 1.0 1.0
        <Mixture> 2 0.6
```

```
<Mean> 4
        1.0 1.0 1.0 1.0
<Variance> 4
        1.0 1.0 1.0 1.0
```

W linii rozpoczynającej opis stanu pojawia się dodatkowy znacznik `<NumMixes>`, po którym następuje liczba określająca liczbę rozkładów normalnych (naszym przypadku jest to 2). Wartości średnie i wariancje dla poszczególnych rozkładów, poprzedzone są liniami zawierającymi znacznik `<Mixture>`, po którym następuje numer rozkładu (numeracja rozpoczyna się od 1) i wartość wagi. Wagi mogą przyjmować dowolne dodatnie wartości, ale ich suma w obrębie jednego stanu musi być równa 1.

W przypadku, gdy konieczne jest podanie pełnej macierzy kowariancji, opis stanu może wyglądać następująco:

```
<State> 2
        <Mean> 4
                1.0 1.0 1.0 1.0
        <InvCovar> 4
                1.0 0.1 0.0 0.0
                    1.0 0.2 0.0
                        1.0 0.1
                            1.0
```

W miejsce znacznika `<Variance>` wpisujemy teraz znacznik `<InvCovar>` i po nim rozmiar macierzy. W następnych liniach wpisujemy macierz odwrotną do macierzy kowariancji. Ponieważ macierz kowariancji jest symetryczna, wpisujemy tylko jej część zaczynając od elementu diagonalnego w każdym wierszu.

Zamieszczone przykłady pokazują tylko część możliwości pakietu HTK, który pozwala m.in. także na budowanie modeli z wyjściem dyskretnym. Kompletny opis dostępnego w pakiecie języka do konstruowania modeli Markowa różnych typów można znaleźć w pracy [81].

D.4 Przygotowanie danych

W pakiecie HTK plik z danymi zawiera nagłówek i następujące po nim sekwencje wektorów cech. Nagłówek ma 12 bajtów i składa się z następujących pól:

`nSamples` - długość sekwencji (liczba wektorów w sekwencji) (4 bajty),
`sampPeriod` - okres próbkowania w jednostkach 100 nanosekundowych (4 bajty),
`sampSize` - liczba bajtów przypadająca na jeden wektor (2 bajty),
`parmKind` - kod określający rodzaj próbki (2 bajty).

Liczba 9 wpisana jako `parmKind` oznacza, że są to dane zdefiniowane przez użytkownika. Po sekcji nagłówka umieszczone są kolejno wektory danych. Wartości w wektorach są zapisane w postaciach 4-bajtowych.

HTK jest pakietem przystosowanym do rozpoznawania mowy i posiada narzędzia umożliwiające nagranie sygnału mowy, przetworzenie go w celu wyznaczenia typowych dla przetwarzania mowy wektorów cech i zapisania w odpowiednim formacie. W przypadku przetwarzania danych o innym charakterze, co miało miejsce w niniejszej pracy, konieczne jest napisanie własnych programów, które potrafią zapisać otrzymane wektory cech w plikach o opisanym powyżej formacie.

D.5 Uczenie

Uczenie przygotowanych modeli Markowa można przeprowadzić w dwóch etapach. W pierwszym etapie dokonuje się wstępnej estymacji parametrów z wykorzystaniem metody Viterbiego. Służy do tego narzędzie `HInit`. Przykładowe wywołanie `HInit` dla modelu `hmm_1` mogłoby wyglądać następująco:

```
HInit -S one_training_list hmm_1.hmm
```

`hmm_1_training_list` zawiera listę plików z danymi, które zostaną użyte do wstępnej estymacji parametrów modelu `hmm_1.hmm` przygotowanego zgodnie ze wskazówkami w podrozdziale D.3. Jeżeli założymy, że do uczenia tego modelu wykorzystamy dziesięć sekwencji wektorów cech, przygotowanych zgodnie z opisem w podrozdziale D.4 i zapisanych w plikach `one_1.mfc`, `one_2.mfc`, ..., `one_10.mfc`, to zawartość pliku `one_training_list` powinna być następująca:

```
one_1.mfc
one_2.mfc
one_3.mfc
one_4.mfc
one_5.mfc
one_6.mfc
one_7.mfc
one_8.mfc
one_9.mfc
one_10.mfc
```

Po uczeniu wstępnym przeprowadza się reestymację parametrów z wykorzystaniem metody Bauma-Welcha. W przypadku rozpoznawania pojedynczych gestów należy użyć narzędzia `HRest`. Przykładowe wywołanie dla modelu `hmm_1` może mieć postać:

```
HRest -S one_training_list hmm_1
```

gdzie `hmm_1` jest plikiem z definicją modelu Markowa po wstępnej estymacji parametrów modelu zapisanych w pliku `hmm_1.hmm`.
W przypadku rozpoznawania sekwencji gestów, uczenie odbywa się z wykorzystaniem *embedded training*. Służy do tego program `HERest`. Poniżej przedstawiono wywołanie programu dla przykładu D.2.

```
HERest training_sentences_list -I training_sentences_structures hmms_list
```

Plik `training_sentences_list` zawiera listę plików z sekwencjami wektorów cech odpowiadającymi wszystkim zdaniom ze zbioru uczącego. W przypadku *embedded tranining* modele uczone są jednocześnie i nie jest wymagana segmentacja poszczególnych zdań na pojedyncze wyrazy. Konieczna jest jedynie informacja o tym, z jakich wyrazów składa się zdanie i w jakiej kolejności te wyrazy występują. Informacja ta zawarta jest w pliku `training_sentences_structures`. Plik `hmms_list` zawiera listę nazw wszystkich modeli Markowa odpowiadających pojedynczym wyrazom. Załóżmy, że dla przykładu D.2 uczenie *embedded training* odbywa się z wykorzystaniem m. in. sekwencji: 123, 23, 321, zapisanych w plikach odpowiednio: `one_two_three.mfc`, `two_three.mfc` i `three_two_one.mfc`. Odpowiadający tym sekwencjom fragment pliku `training_sentences_list` przedstawiono poniżej:

```
one_two_three.mfc
two_three.mfc
three_two_one.mfc
```

Fragment pliku `training_sentences_structures` powinien być następujący:

```
#!MLF!#
"one_two_three.lab"
one
two
three
.
"two_three.lab"
two
three
.
"three_two_one.lab"
three
two
one
.
```

Plik ten rozpoczyna się od znaków `#!MLF!#`, po których następują, rozdzielone znakiem '.' definicje struktury poszczególnych sekwencji ze zbioru uczącego. Każda struktura opisana jest poprzez podanie nazwy pliku z rozszerzeniem `lab` zamiast `mfc` i następnie listy wyrazów wchodzących w skład danej sekwencji.
Dla przykładu D.2 Plik `hmms_list` powinien zawierać listę modeli odpowiadających poszczególnym cyfrom:

```
hmm_1
hmm_2
hmm_3
hmm_4
hmm_5
hmm_6
```

```
hmm_7
hmm_8
hmm_9
hmm_0
```

Wynikiem działania skryptów `HRest` i `HERest` są uaktualnione pliki z definicjami wyuczonych HMM (w naszym przykładzie `hmm_1`, `hmm_2`, ..., `hmm_0`).

D.6 Testowanie i weryfikacja wyników

Rozpoznawanie wykonywane jest z wykorzystaniem algorytmu Viterbiego. Służy do tego skrypt `HVite`. Dla przykładów D.1 - D.3 wywołanie skryptu może mieć postać:

```
HVite -i wyniki.mlf -I testing_sentences_structures -w word_network
      -H newMacros -S testing_data_list dictionary hmms_list
```

gdzie: `hmms_list` jest listą ukrytych modeli Markowa, `dictionary` jest plikiem słownika, `word_network` jest plikiem z definicją struktury modeli utworzonym z pliku z definicją gramatyki zadania za pomocą skryptu `HParse`, `testing_data_list` zawiera listę plików z sekwencjami wektorów cech odpowiadającymi zdaniom ze zbioru testowego. Opis struktury zdań wykorzystanych w testowaniu zawarty jest w pliku `testing_sentences_structures`. Format tego pliku jest analogiczny jak format pliku `training_sentences_structures`, opisany w podrozdziale D.5. Plik `wyniki.mlf` zawiera wyniki rozpoznawania zapisane z wykorzystaniem takiego samego formatu jak plik `testing_sentences_structures`.

W celu określenia skuteczności rozpoznawania należy porównać pliki `wyniki.mlf` i `testing_sentences_structures`. Służy do tego skrypt `HResults`. Jego wywołanie może mieć postać:

```
HResults -t -I testing_sentences_structures words_list wyniki.mlf
```

gdzie: `words_list` jest listą słów. Poniżej przedstawiono przykładową statystykę otrzymaną w wyniku wywołania skryptu:

```
====================== HTK Results Analysis =======================
  Date: Thu Jul 29 08:19:45 2004
  Ref : testing_sentences_structures
  Rec : wyniki.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=90.29 [H=316, S=34, N=350]
WORD: %Corr=92.93, Acc=92.53 [H=1394, D=37, S=69, I=6, N=1500]
===================================================================
```

Linia rozpoczynająca się od znacznika `SENT:` zawiera statystykę dotyczącą całych zdań. Po słowie `%Correct` umieszczona jest skuteczność rozpoznawania, która w tym przypadku wynosi 90.29%. Następnie w nawiasie kwadratowym podane są statystyki liczbowe. `H=316` oznacza liczbę sentencji rozpoznanych, `S` liczbę sentencji

nierozpoznanych, zaś `N` liczbę wszystkich zdań w zbiorze wykorzystanym do testowania. Skuteczność rozpoznawania określona jest następująco:

$$\%Correct = \frac{H}{N}100\% \tag{D.1}$$

Linia rozpoczynająca się od znacznika `WORD:` zawiera statystyki dotyczące pojedynczych słów. Po słowie `%Corr` umieszczona jest skuteczność rozpoznawania słów, następnie dokładność rozpoznawania i umieszczone w nawiasach kwadratowych statystyki liczbowe. `H` oznacza teraz liczbę wyrazów rozpoznanych poprawnie, `D` oznacza liczbę wyrazów usuniętych z rozpoznawanych zdań, `S` liczbą wyrazów zastąpionych innymi, `I` liczbę wyrazów wstawionych do rozpoznawanych zdań i `N` łączną liczbę wyrazów wchodzących w skład zdań. Dokładność rozpoznawania wyrazów określona jest następująco.

$$Acc = \frac{H - I}{N}100\% \tag{D.2}$$

W niniejszym dodatku pokazano tylko niewielką część możliwości pakietu HTK. Opisywane skrypty mają liczne przełączniki i parametry sterujące oraz mogą być wywoływane na wiele różnych sposobów. Pełną dokumentację pakietu można znaleźć w pracy [81].

Dodatek E

Rozpoznawane wyrazy i zdania

Tab. E.1. *Rozpoznawane wyrazy i ich zapis gestograficzny*

lp	wyraz	gestogram	I	II	III
1	analiza	LH:25k.PH:25k # XII\II\VII<”	k	+	+
2	angina	[PBk:78s+ ## PEo]”	s	+	-
3	aparat	PP:77tpp+ # XII\IV-<	t	+	-
4	apteka	PA:53k)/LBk:13k # P:IX\VI-≪&PC:51k}{LBk:11k # P:II	k	+	+
5	audiogram		t, k	+	+
6	badać	PH:58k|LB:58k+ # P:XIII\IV+<”	k	+	+
7	bezpłatny	LE:13k}{PE:23k # P:III!;L:IV<! # P5:23k||L5:13k	k	+	+
8	boleć	P5z:58k+ # IV≪+ # @	k	+	-
9	być	PA:23k)/LA:53k # P:III-	k	+	+
10	chcieć	L5z:11kl..P5z:11kp # II\III”!	k	+	+
11	chory	PUm:53k/LUm:53k # P:II+<”	k	+	+
12	codziennie	P1:35tp+ # XI\III<”	t	+	-
13	czuć	P5z:16k+ # I-<”	k	+	-
14	czy	PCz* # V<!	k	+	-
15	do	PA:21kp.LB:51kl # P:VI<+!	k	+	+
16	dokładny	PO:35tpg+ & PO:51kg / LO:51k # P:II+II	t, k	+	+
17	dużo	L5:21klg..P5:21kpg # III<”	k	+	+
18	dzisiaj	LO:51kl..PO:51kp # II<!	k	+	+
19	gardło	PB:78s+	s	-	-
20	gdzie	L5:11kld..P5:11kpd # VII<”	k	+	+
21	głowa	PZ:25tppg+	t	-	-
22	gorączka	P5z:78td+ # XVIII! # P5z:37tpd & LB:35klg)(PZ:35klg # P:I-≤	t, k	+	+
23	grypa	PC:58s+ # II-<”	s	+	-
24	i	PZ:23k/LI:13k # P:II<+=	k	+	+
25	ile	P5:11kp # XIV	k	+	-
26	inny	LI:78kl..PI:78kp # XII\II\VII<** # LI:48klld...PI:48kppd	k	+	+
27	ja	PZ:54k+	k	-	-
28	kaszleć	PM:58k+ # XIII\II+<!”	k	+	-
29	katar	PU:38t+ # II-<<	t	+	-
30	kosztować	PE:13k/LBk:13k # P:II+”	k	+	+
31	krew	PG:23kX)/LA:53k # PG*”	k	+	+
32	kropla	PX:23k/LOm:53k # P:II<<!	k	+	+

lp	wyraz	gestogram	I	II	III
33	krople	PX:23k/LOm:53k # P:II<<!”	k	+	+
34	lekarstwo	PA:53k)/LBk:13k # P:IX\VI-<<	k	+	+
35	lekarz	PH:23k/LA:23k # P:XIII\II+<”	k	+	+
36	leżeć	PU:13k)/LBk:13k	k	-	+
37	list	PIk:3’5tpd+ # III\V! & PA:53k/LBk:13k # P:II+!	t, k	+	+
38	łóżko	LNw:51kl.PNw:51kp # IV	k	+	+
39	mieć	PB:58k+	k	-	-
40	mnie	PM:58k+	k	-	-
41	mózg	PB:25tw+	t	-	-
42	musieć	PTm:52kp # VI\II>!	k	+	-
43	na	PNw:23k/LBk:13k # P:II+	k	+	+
44	natychmiast	LT:51KL..PT:51k # XIII\V≥! # L5:5’1k..P5:5’1kp	k	+	+
45	nie	PN:51k # XI\V # PE:71kp	k	+	-
46	o		k	-	-
47	odpoczywać	LL:35kl..PL:35kp # IV+<	k	+	+
48	okulary		t	+	+
49	opatrunek	PUm:58k/LUm:58k # P:X\III\II	k	+	+
50	operacja	LZ:86kd.PZ:86kd # XVI\III\IV”	k	+	+
51	opony	PB:25tw+	t	-	-
52	otrzymać	L5z:79kl.P5z:79kp # XII\IV+ # LE:38klg+.PE:38kpg+	k	+	+
53	paczka	LBk:51kl/PBk:51kp ## PBk:58k||LBk:58k	k	+	+
54	palić	PUm:78tpd # IV+<<”	t	+	-
55	pan	[(PZ:37tp+ # III<)] & LO:53klld+...PO:53kppd+	t, k	+	+
56	pielęgniarka	PU:58tlgg # V	t	+	-
57	pisać	PE:23k}/LBk:13k # P:III\V<-”	k	+	+
58	płacić	PE:13kp # XX # PAw:13kp	k	+	-
59	płuca	P5:58kl+ ## P5:58kp+	k	+	-
60	po	LZ:75k}{PZ:77k # P:XIII\III # PZ:72k|LZ:35k	k	+	+
61	pobieranie	PPs:58kll+ ## P3:58kll	k	+	-
62	poczta	PIk:3’5tpd+ # III\V!	t	+	-
63	pocztówka	PIk:3’5tpd+ # III\V! & LL:23k}{PL:23k # VII<	t, k	+	+
64	pogotowie	PIk:35tpd+ # III< & LT:51kl..PT:51k # XII\V≥! # L5:5’1k..P5:5’1kp	t, k	+	+
65	pójść	PL:37tpg # III> # PP:73tpg	t	+	-
66	pokazać	PU:38t+ # III & LB:37klg..PB:37kpg # III	t, k	+	+
67	polecony	PZ:78tpd+ # XVIII\XI # PZ:37tppd # XII\III\II # PZ:21kpg	t, k	+	-
68	położyć	PU:13k/LBk:13k # P:II+	k	+	+
69	potrzebny	LB:51kl..PBz:58kp # P:VI<+”	k	+	+
70	prosić	LB:72k)(PB:7k # XIII\II\I # @	k	+	+
71	prześwietlenie	PEm:58k||L5:58k+ # P:IV+ # P5	k	+	+
72	przeziębiony	LZ:54kl+.PZ:54kp+ & LA:51k.PA:51k # XVII\VIII	k	+	+
73	przyjmować	LBk:21kl..PBk:21kp # IV+ # LE:48kl+..PE:48k+	k	+	+
74	przyjść	PZ:38kpg # XII\I;VI;II> # PZ:45kp	k	+	-
75	rachunek	PBz:73kp|LA:78k # P:IV\VI+”	k	+	+
76	ratunek	L5:13klld...P5:13kppd # I\VIII≥! # LA:13k.PA:13k	k	+	+
77	recepta	PA:53k)/LBk:13k # P:IX\VI-<< & PC:51k}{LBk:11k # P:II	k	+	+
78	rodzinny	L3:23kld..P3:23kpd # IX\VIII<”	k	+	+
79	rozebrać	L500:58k.P500:58kg # XII\I\VII-<”	k	+	+
80	się	P1:25klg+ ## P1:25kpd+	k	+	-
81	skierowanie	PO:23k}/LA:53k # III<	k	+	+
82	słuch	PZ:25tpp+	t	-	-
83	słyszeć	PZ:25tpp+	t	-	-
84	szpital	LB:78tlg+.PB:78tpg+ ## LBz:58klg+..PBz:58kpg+	t, k	+	+
85	tabletka	PZ:23k}/LBk:13k # P:IX\VI-<<	k	+	+

lp	wyraz	gestogram	I	II	III
86	ten	PZ:21kp # III\II<!	k	+	-
87	termometr	PZ:54kll+	k	-	-
88	w	LBk:58k++PBk:23k	k	-	+
89	wata	L5:16k}{P5:16k ## LE..PE	k	+	+
90	wykonać		k	+	+
91	wysyłać	LZ:58k}{PZ:58k # P:XII\II # PZ:21kpg	k	+	+
92	ząb	PX:78tpd+ # PX:78tld+	t	+	-
93	zapalenie	L5:72k/P5:72kd # P:I≥!+I≥! # P5:72kg/L5:72k	k	+	+
94	zastępstwo		k	+	+
95	zastrzyk	P3:58kll+ ## PPs:58kll	k	+	-
96	zdrowy	LM:88klld...PM:88kppd # I+<<”	k	+	+
97	zęby	PX:78tpd+ ## PX:78tld+	t	+	-
98	źle	PBk:51kp # XIII\VI\II≥!	k	+	-
99	znaczek	PUm:78td+ & PUm:23kg/LBk:13k # P:II+	t, k	+	+
100	żołądek	P5s:58k # IV+<”	k	+	-
101	zwolnienie		k	+	+

Oznaczenia:
I - miejsce wykonywania znaku (t - twarz, s - szyja, k - klatka piersiowa),
II - znak dynamiczny (+) bądź statyczny (-),
III - znak dwuręczny (+) bądź jednoręczny (-).

Tab. E.2. *Rozpoznawane zdania*

lp	zdanie
1	Boli mnie gardło. [*boleć*][*mnie*][*gardło*]
2	Boli mnie głowa. [*boleć*][*mnie*][*głowa*]
3	Boli mnie ząb. [*boleć*][*mnie*][*ząb*]
4	Boli mnie żołądek. [*boleć*][*mnie*][*żołądek*]
5	Jestem chory. [*być*][*chory*]
6	Chcę otrzymac list. [*chcieć*][*otrzymać*][*list*]
7	Chcę otrzymać paczkę. [*chcieć*][*otrzymać*][*paczka*]
8	Chcę zapłacić rachunek. [*chcieć*][*płacić*][*rachunek*]
9	Chcę wysłać list. [*chcieć*][*wysyłać*][*list*]
10	Chcę wysłac paczkę. [*chcieć*][*wysyłać*][*paczka*]
11	Chcę wysłać pocztówkę. [*chcieć*][*wysyłać*][*pocztówka*]
12	Czy dzisiaj przyjmuje lekarz rodzinny? [*czy*][*dzisiaj*][*przyjmować*][*lekarz*][*rodzinny*]
13	Czy potrzebne jest zwolnienie lekarskie? [*czy*][*potrzebny*][*być*][*zwolnienie*][*lekarz*]
14	Czy przyjmuje inny lekarz w zastępstwie? [*czy*][*przyjmować*][*inny*][*lekarz*][*w*][*zastępstwo*]
15	Czy to lekarstwo jest bezpłatne? [*czy*][*ten*][*lekarstwo*][*być*][*bezpłatny*]
16	Ile kosztuje wysłanie listu? [*ile*][*kosztować*][*wysyłać*][*list*]
17	Ile kosztuje wysłanie paczki? [*ile*][*kosztować*][*wysyłać*][*paczka*]
18	Ile kosztuje wysłanie pocztówki? [*ile*][*kosztować*][*wysyłać*][*pocztówka*]
19	Ile kosztuje znaczek? [*ile*][*kosztować*][*znaczek*]
20	Ja nie słyszę po chorobi zapalenia opon mózgowych. [*ja*][*nie*][*słyszeć*][*po*][*chory*][*zapalenie*][*opony*]
21	Mam gorączkę. [*mieć*][*gorączka*]
22	Musi pan leżec i dużo odpoczywać. [*musieć*][*pan*][*leżeć*][*i*][*dużo*][*odpoczywać*]
23	Pan jest chory. [*pan*][*byc*][*chory*]
24	Pan jest chory na anginę i otrzymuje zwolnienie lekarskie. [*pan*][*być*][*chory*][*na*][*angina*][*i*][*otrzymać*][*zwolnienie*][*lekarz*]
25	Pan jest chory na grypę i musi leżeć w łóżku. [*pan*][*być*][*chory*][*na*][*grypa*][*i*][*musieć*][*leżeć*][*w*][*łóżko*]
26	Pan musi codziennie przychodzić do pielęgniarki na zastrzyk. [*pan*][*musieć*][*codziennie*][*przyjść*][*do*][*pielęgniarka*][*na*][*zastrzyk*]
27	Pan musi pójść do szpitala. [*pan*][*musieć*][*pójść*][*do*][*szpital*]

lp	zdanie
28	Wypiszę panu receptę. [*pisać*][*pan*][*recepta*]
29	Proszę o skierowanie na badania. [*prosić*][*o*][*skierowanie*][*na*][*badać*]
30	Proszę pójść do apteki. [*prosić*][*pójść*][*do*][*apteka*]
31	Proszę pokazać dokładnie gdzie pana boli. [*prosić*][*pokazać*][*dokładny*][*gdzie*][*pan*][*boleć*]
32	Proszę wykonać audiogram. [*prosić*][*wykonać*][*audiogram*]
33	Proszę znaczek na list. [*prosić*][*znaczek*][*na*][*list*]
34	Proszę znaczek na pocztówkę. [*prosić*][*znaczek*][*na*][*pocztówka*]
35	Źle się czuję. [*źle*][*się*][*czuć*]

Dodatek F

Opis zawartości płyty DVD

Załączony dysk DVD zawiera przykładowe wykonania słów i zdań Polskiego Języka Miganego wykorzystywane w badaniach. Wyróżniono cztery główne foldery: `Words`, `Sentences`, `Words_Image_Processing`, `Sentences_Image_Processing`.

W folderze `Words` zamieszczono po jednym przykładowym wykonaniu wszystkich 101 wykorzystywanych wyrazów, zapisanym w folderze o nazwie odpowiadającej wyrazowi. Każdy folder wyrazu zawiera folder `color_left` z sekwencją obrazów kolorowych z kamery lewej, `color_right` z sekwencją obrazów kolorowych z kamery prawej, `gray_left` z sekwencją obrazów z poziomami szarości z kamery lewej, `gray_right` z sekwencją obrazów z poziomami szarości z kamery prawej oraz plik `*.mfc` z sekwencją wektora cech.

W folderze `Sentences` zapisano po jednym przykładowym wykonaniu wszystkich 35 wykorzystywanych zdań. Folder podzielony jest na foldery o nazwach odpowiadających nazwom sekwencji w transkrypcji wykorzystywanej przy rozpoznawaniu. Organizacja folderu sekwencji jest taka sama jak folderu wyrazu.

Wszystkie obrazy wejściowe są zrektyfikowane. Strukturę pliku `*.mfc` omówiono w dodatku D.4.

Foldery `Words_Image_Processing` i `Sentences_Image_Processing` zawierają wszystkie obrazy pośrednie generowane w trakcie wyznaczania wektora cech, odpowiednio dla wybranych 15 wyrazów i wszystkich 35 zdań. Zapisane obrazy pośrednie to obrazy: prawdopodobieństwa (folder `probability`), binarne (folder `binary`), po filtracji OC (folder `morph`), map dysparycji (folder `disparity`). Dołączono także obrazy wejściowe z kamery lewej z zaznaczonymi obszarami rozpoznanymi jako twarz, dłoń prawa i dłoń lewa (folder `segment`).

Wszystkie obrazy mają rozdzielczość 320 x 240 i zapisane są w formacie bmp.

Baza danych wizyjnych wykorzystywanych w niniejszej pracy jest dostępna na stronie `http://wizja.prz-rzeszow.pl/`.

Literatura

[1] S. Akyol and P. Alvarado. Finding Relevant Image Content for mobile Sign Language Recognition. *Proc. IASTED Int. Conf. Signal Processing, Pattern Recognition and Application* 48-52, 2001.

[2] M. Assan, and K. Grobel. Video-Based Sign Language Recognition Using Hidden Markov Models. *Proc. Gesture Workshop*, 97-109, 1997.

[3] B. Bauer, and K. F. Kraiss. Video-Based Sign Recognition Using Self-Organizing Subunits. *Proc. Int. Conf. Patter Recognition*, vol.2, 434-437, 2002.

[4] S. Birchfield, and C. Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo. *Proceedings of the 1998 IEEE International Conference on Computer Vision*, 1073-1080, Bombay, India, 1998.

[5] H. J. Boehme, U. D. Braumann, A. Brakensiek, A. Corradini, M. Krabbes, H.-M. Gross, and User localisation for Visually-based Human-Machine-Interaction. *Proceedings of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, 486-491, Nara, Japan 1998.

[6] H. Bourlard, and S. Dupont. Subband-based speech recognition. *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, vol. 2., 1251-1254, 1997.

[7] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. *IEEE Trans. PAMI*, 25, 8, 2003, 993-1008.

[8] S. B. Cho, and J. H. Kim. Multiple Network Fusion Using Fuzzy Logic. *IEEE Trans. On Neural Networks*, 6, 2, 497-501, 1995.

[9] R. S. Malina. *Komputerowa wizja. Metody interpretacji i identyfikacji obiektów.* Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2005.

[10] Y. Cui, and J. Weng. Appearance-Based Hand Sign Recognition from Intensity Image Sequences. *Computer Vision Image Understanding*, vol. 78, no. 2, 157-176, 2000.

[11] B. Cyganek. *Komputerowe przetwarzanie obrazów trójwymiarowych.* Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002.

[12] Y. Dai, and Y. Nakano. Face-texture model based on SGLD and its application in face detection in a colour scene. *Pattern Recognition*, Vol.29, No.6, pp.1007-1017, 1996.

[13] J. R. Deller, J. G. Proakis, and J. H. L. Hansen. *DiscreteTtime Processing of Speech Signals.* Macmillan Publ. Comp., New York, 1993.

[14] M. Domański. *Zaawansowane techniki kompresji obrazów i sekwencji wizyjnych.* Wyd. Politechniki Poznańskiej, Poznań, 2000.

[15] W. Duch, J. Korbicz, L. Rutkowski, and R. Tadeusiewicz. *Sieci Neuronowe.* EXIT, Warszawa 2000.

[16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* J. Wiley & Sons, INC., New York, 2001.

[17] G. Fang, W. Gao, X. Chen, C. Wang, and J. Ma. Signer Independent Continuous Sign Language Recognition Based on SRN/HMM. *Proc. Gesture Workshop*, 76-85, 2001.

[18] O. Faugeras. Digital color image processing within the framework of a human visual model. *Acoustics, Speech, and Signal Processing*, 27, 4, 380-393, 1979.

[19] O. Faugeras, B. Hotz, H. Matthieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications. *INRIA Technical Report* 2013, 1993.

[20] G. D. Jr. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61, 3, 268-278, 1973.

[21] K. Fukunaga. *Introduction to statistical patern recognition.* Acad. Press, New York 1972.

[22] Z. Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *Journal of Pattern Recognition and Artificial Intelligence*, 15, 1, 9-42, 2001.

[23] K. Grobel, and M. Assam. Isolated Sign Language Recognition Using Hiden Markov Models. *Proc. of the IEEE Int. Conf. on SMC*, 162–167, Orlando, 1997.

[24] J. K. Hendzel. *Słownik Polskiego Języka Miganego.* Wydawnictwo Żakiel", Olsztyn, 2000.

[25] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman. A New Instrumental Approach for Translating American Sign Language into Sound and Text. *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 547-552, 2004.

[26] E. J. Holden, and R. Owens. Visual Sign Language Recognition. *Proc. Int. Workshop Theoretical Foundations of Computer Vision*, 270-287, 2000.

[27] C. L. Huang, and W. Y. Huang. Sign Language Recognition Using Model-Based tracking and a 3D Hopfield Neural Network. *Machine Vision and Application*, vol. 10, 292-307, 1998.

[28] H. R. Hyler, and A. R. Weeks. *The Pocket Handbook of Image Processing Algorithms in C.* Prentice Hall, Englewood Cliffs, 1993.

[29] K. Imagawa, S. Lu, and S. Igi. Color-Based Hand Tracking System for Sign Language Recognition. *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 462-467, 1998.

[30] K. Imagawa, H. Matsuo, R.-i. Taniguchi, D. Arita, S. Lu, and S. Igi. Recognition of Local Features for Camera-Based Sign Language Recognition System. *Proc. Int. Conf. Pattern Recognition*, 4, 849-853, 2000.

[31] F. V. Jensen. *Bayesian Networks and Decision Graphs.* Springer, New York, 2001.

[32] M. J. Jones, and J. M. Rehg. Statistical Color Models with Application to Skin Detection. *Proc. of the 18th Conf. Computer Vision and Pattern Recognition*, vol. 1., 274-280, Fort Collins, Colorado 1999.

[33] M. W. Kadous. Machine Recognition ofAuslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language. *Proc. Workshop Integration of Gestures in Language and Speech*, 165-174, 1996.

[34] T. Kapuściński, J. Marnik, and M. Wysocki. Problemy rozpoznawania gestów wykonywanych rękami. *Pomiary, Automatyka, Kontrola* 8, 22-25, 1999.

[35] T. Kapuściński, and M. Wysocki. Identyfikacja koloru skóry dłoni w różnych przestrzeniach barw. *Archiwum Informatyki Teoretycznej i Stosowanej*, Tom 13, z. 1, 53-68, 2001.

[36] T. Kapuściński, J. Marnik, and M. Wysocki. Rozpoznawanie gestów rąk w układzie wizyjnym. *Pomiary, Automatyka, Kontrola* 1, 56-59, 2005.

[37] T. Kapuściński, and M. Wysocki . Ukryte modele Markowa i ich zastosowanie do rozpoznawania zdarzeń na podstawie sekwencji wizyjnych. *Pomiary Automatyka Kontrola*, 9bis, 306-308, 2005.

[38] T. Kapuściński, and M. Wysocki. Recognition of Isolated Words of the Polish Sign Language. *Proc. of the CORES'05, Computer Recognition Systems*, Springer, Heidelberg, 697-704, 2005.

[39] T. Kapuściński, and M. Wysocki. Automatic Recognition of Signed Polish Expressions. *Proc. of the 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, 261-264, Poznań, Poland, 2005, zob. też *Archives of Control Sciences.* Vol 15(II) 2005, No 3, 251-259.

[40] S. Katagiri (Ed.). *Handbook of neural networks for speech processing.* Artech House, London, 2000.

[41] T. Kobayashi, and S. Haruyama. Partly-Hidden Markov Model and Its Application to Gesture Recognition. *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, 3081-3084, 1997.

[42] K. Konoliege. Small Vision System: Hardware and Implementation. *8th International Symposium on Robotics Research*, Japan, 1997.

[43] M. Kurzyński. *Rozpoznawanie obrazów - metody statystyczne.* Oficyna Wyd. Politechniki Wrocławskiej, Wrocław, 1997.

[44] R. H. Liang, and M. Ouhyoung. A Real-Time Continuous Gesture Recognition System for Sign Language. *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 558-565, 1998.

[45] W. Malina, and M. Smiatacz. *Metody cyfrowego przetwarzania obrazów.* Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2005.

[46] A. Marciniak, and J. Korbicz. Neuronowe sieci modularne (135-177) [w:] *Sieci Neuronowe.* Akad. Ofic. Wyd. Exit, W-wa, 2000, red. W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz.

[47] J. Marnik. *Rozpoznawanie znaków Polskiego Alfabetu Palcowego z wykorzystaniem morfologii matematycznej i sieci neuronowych. Rozprawa doktorska.* Akademia Górniczo-Hutnicza, Kraków, 2002.

[48] A. D. Marshall, and R. R. Martin. *Computer Vision, Models and Inspection.* World Scientific, London, 1993.

[49] H. Matso, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima. The Recognition Algorithm with Non-Contact for Japanese Sign Language Using Morphological Analysis. *Proc. Gesture Workshop*, 273-285, 1997.

[50] K. Murakami, and H. Taguchi. Gesture Recognition Using Recurrent Neural Networks. *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 237-242, 1991.

[51] M. Nieniewski. *Morfologia matematyczna w przetwarzaniu obrazów.* Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1998.

[52] M. Ostrowski. *Informacja obrazowa.* Wydawnictwa Naukowo-Techniczne, Warszawa, 1992.

[53] N. Otsu. A threshold selection method from grey-level histograms. *IEEE Trans. on systems, Man, and Cybernetics*, SMC-8, No.1, pp. 62-66, 1979.

[54] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual Interpretation of Hand Gestrures for Human-Computer Interaction: A Review. *IEEE Trans. PAMI*, 19, 7, 677–693, 1997.

[55] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications In Speech Recognition. *Proceedings of the IEEE*, 77, 2, 257–286, 1989.

[56] R. Rosenfeld. Two Decades of Staristical Language Modeling: Where do we go from here? *Proceedings of the IEEE*, 88, 8, 1270-1278, 2002.

[57] H. Sagawa, and M. Takeuchi. A Method for Recognizing a Sequence of Sign Language Words Represented in a Japanese Sign Language Sentence. *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 434-439, 2000.

[58] S. J. Sangwin, and R. E. N. Horne (Eds.). *Colour Image Processing.* Chapman and Hall, London, 1998.

[59] J. Sherrah and S. Gong, Resolving Visual Uncertainty and Occlusion through Probabilistic Reasoning, *Proc. British Machine Vision Conf.*, 252-261, 2000.

[60] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision.* Chapman & Hall, London 1994.

[61] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Trans.PAMI*, 20, 12, 1371–1375, 1998.

[62] M. C. Su. A Fuzzy Rule-Based Approach to Spatio-Temporal Hand Gesture Recognition. *IEEE Trans. Systems, Man, and Cybernetics, Part C: Application Rev.*, vol. 30, no. 2, 276-281, 2000.

[63] N. Suszczanska, P. Szmal, and J. Francik. Translation Polish Text into Sign Language in the TGT System. *Proc. of the 20th IASTED International Multiconference Applied Informatics*, 282–287, Insbruck, 2002.

[64] N. Suszczanska, and P. Szmal. Categorial grammar elements in the Thetos system's parser. *Proc. of the 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, 338-342, Poznań, Poland, 2005.

[65] B. Szczepankowski. *Język migany w szkole.* Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1988.

[66] B. Szczepankowski. *Niesłyszący-Głusi-Głuchoniemi. Wyrównywanie szans.* Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1999.

[67] R. Tadeusiewicz. *Systemy wizyjne robotów przemysłowych.* WNT, Warszawa, 1992.

[68] S. Tamura, and S. Kawasaki. Recognition of Sign Language Motion Images. *Pattern Recognition*, vol. 21, no. 4, 343-353, 1998.

[69] N. Tanibata, N. Shimada, and Y. Shirai. Extraction of Hand Features for Recognition of Sign Language Words. *Proc. Int. Conf. Vision Interface*, 391-398, 2002.

[70] J. C. Terrillon, A. Piplr, Y. Niwa, and K. Yamamoto. Robust Face Detection and Japanese Sign Language Hand Posture Recognition for Human-Computer Interaction in an Intelligent Room. *Proc. Int. Conf. Vision Interface*, 369-376, 2002.

[71] S. Theodoridis, and K. Koutroumbas. *Pattern Recognition.* Acad. Press, London, 1999.

[72] J. Triesch, and Ch. Von der Malsburg. A Gesture Interface for Human-Robot Interaction. *Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, 546-551, Nara, Japan 1998.

[73] R. Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 364–374, Miami Beach, 1986.

[74] P. Vamplew, and A. Adams. Recognition of Sign Language Gestures Using Neural Networks. *Australian J. Intelligence Information Processing Systems*, vol. 5, no. 2, 94-102, 1998.

[75] C. Vogler, and D. Metaxas. A framework for Recognizing the Simultaneous Aspects of American Sign Language. *Computer vision and Image Understanding*, 81, 358-384, 2001.

[76] M. B. Waldron, and S. Kim. Isolated ASL Sign Recognition System for Deaf Persons. *IEEE Trans. Rehabilitation Eng.*, vol. 3, no. 3, 261-271, 1995.

[77] C. Wang, W. Gao, and S. Shan. An Approach Based on Phonemes to Large Vocabulary Chinese Sign Language Recognition. *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 393-398, 2002.

[78] J. Wu, and W. Gao. A Fast Sign Word Recognition Method for Chinese Sign Language. *Proc. Int. Conf. Advances in Multimodal Interfaces*, 599-606, 2000.

[79] J. Yang, Y. Xu, and C. Chen. Human Action Learning via Hidden Markov Model. *IEEE Trans. SMC*, 27, 1, 34-44, 1997.

[80] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24, 8, 1061-1074, 2002.

[81] S. Young, and at al. *The HTK Book.* Microsoft Corporation, 2000.

[82] J. Zieren, N. Unger, and S. Akyol. Hands Tracking from Frontal View for Vision-Based Gesture Recognition. *Proc. 24th DAGM Symp*, 531-539, 2002.

[83] Declaration on the Rights of Disabled Persons: http://www.unhchr.ch/html/menu3/b/72.htm.

[84] Europejska Karta Społeczna: http://www.mgip.gov.pl/Dialog+Spoleczny/REGULACJE+PRAWNE/Europejska+Karta+Spoleczna.