AKADEMIA GÓRNICZO–HUTNICZA

IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

KATEDRA INFORMATYKI

Autoreferat rozprawy doktorskiej

Zastosowanie grupowania szeregów czasowych do rozpoznawania wypowiedzi w języku migowym na podstawie sekwencji wizyjnych

mgr inż. Mariusz Oszust

Promotor:
dr hab. inż. Marian Wysocki

Kraków, 2013

1 Wstęp

Rozpoznawanie gestów wykonywanych rękami wpisuje się w ogólnoświatowy trend tworzenia naturalnych interfejsów człowiek–komputer, zwłaszcza tam, gdzie standardowe manipulatory (joystick, mysz, klawiatura) nie mogą mieć zastosowania lub bezpośredni kontakt z maszyną jest niewygodny. Jednym z ważnych zastosowań rozpoznawania gestów wykonywanych rękami jest automatyczna interpretacja wypowiedzi w języku migowym. Z tym obszarem wiąże się niniejsza praca dotycząca używanego w Polsce systemu językowo–migowego (SJM), noszącego też nazwę języka miganego.
Badania prowadzone w ramach pracy zmierzają do utworzenia narzędzi wspomagających rozpoznawanie wypowiedzi SJM. Narzędzia takie ułatwiłyby komunikację w typowych sytuacjach, gdzie zwykle wymagana jest obecność tłumacza. Mogłyby także spełniać rolę edukacyjną w zakresie nauki SJM zwiększając kompetencje pracowników. Ponadto wspomagałyby nauczanie dzieci głuchych wzbogacając środki komunikacji z nauczycielami lub rodzicami, którzy często są osobami słyszącymi. Byłoby to elementem sprzyjającym usuwaniu barier utrudniających osobom głuchym funkcjonowanie w społeczeństwie.

1.1 Charakterystyka polskiego języka miganego

Badania nad językami migowymi na świecie prowadzone są od kilkudziesięciu lat, zaś pierwsze badania dotyczące polskiego języka migowego pojawiły się dopiero w latach dziewięćdziesiątych ub. wieku. W Polsce wyróżnia się dwa języki migowe – polski język migowy, który jest językiem naturalnym, charakteryzującym się odrębną gramatyką oraz polski język migany (SJM). SJM jest językiem, którego można nauczyć się na kursach języka migowego, jest on również obecny w tłumaczeniach prezentowanych w telewizji. Popularyzacją oraz szkoleniami w zakresie SJM zajmują się stowarzyszenia takie jak Polski Związek Głuchych.

W znakach migowych może występować 48 układów palców u każdej z rąk, daje to $(48^2 + 48)$ 2352 układów dłoni. W praktyce wykorzystuje się około 200 układów.

Dłoń może występować w 32 możliwych orientacjach (32 dla znaków jednoręcznych i 1024 dla dwuręcznych), jednakże w codziennym użytku jest około 180 orientacji. Wyróżnia się 37 miejsc artykulacji, czyli położenia rąk w określonym miejscu przed ciałem lub w stosunku do siebie - 17 przed lub obok twarzy, 18 przed klatką piersiową, 2 przed dolną połową ciała. Każde z wymienionych miejsc może być w pewnym stopniu odsunięte od ciała, w części miejsc natomiast (26) występuje styk dłoni z ciałem. Ponieważ układ dłoni, jej orientacja i pozycja odnoszą się do gestów statycznych lub określają dłoń na początku wykonywania gestu dynamicznego, a większość gestów (98%) w języku migowym to gesty dynamiczne, należy uwzględnić także ruch dłoni. Ręka (ręce) w gestach dynamicznych zmieniają swoje pozycje, orientacje oraz układy dłoni. Sam ruch może być wykonany w odmienny sposób uwzględniając: (i) prędkość wykonania, (ii) długość wykonywania gestu (iii) powtórzenia, (iv) zatrzymania ręki po wykonaniu ruchu, (v) dotykania ciała lub ręki podczas wykonywania gestu i (vi) ułożenia ręki na drugiej ręce lub ciele w końcowej fazie wykonywania ruchu. Uwzględniając możliwości różnicowania wymienionych cech dystynktywnych otrzymuje się ponad 100 miliardów kombinacji znaków migowych [40].

Gesty są wykonywane w sposób zarówno sekwencyjny, jak i symultaniczny. Sekwencyjność oznacza wykonywanie poszczególnych części gestu w określonej kolejności, symultaniczność oznacza zaś, że pewne cechy dłoni podczas wykonywania gestu mogą zmieniać się w czasie, gdy pozostałe pozostają niezmienione [47].

Przykładowe gesty przedstawiono na rysunku 1.

Rysunek 1: W wierszach umieszczono kolejne klatki wykonań wyrazów *głowa*, *słuch* i *ząb* (od góry do dołu)

1.2 Przegląd literatury

Od czasu, kiedy automatyczne rozpoznawanie gestów było po raz pierwszy poruszane w literaturze [43], wiele różnych podejść zostało zaproponowanych. Zbiorcze porównanie części z nich można spotkać w pracach przeglądowych [5, 7, 27, 36], które przedstawiają większość problemów związanych z automatycznym rozpoznawaniem języka migowego, jednakże tematy (i) modelowania wyrażeń języka migowego za pomocą jednostek mniejszych niż słowa, (ii) rozszerzania słownika rozpoznawanych gestów o nowe słowa, czy (iii) rozpoznawania w oparciu o małą liczbę wykonań gestów uczących są poruszane w stopniu niewystarczającym wskazując tym samym trudność i potrzebę badań w tym obszarze.

Chcąc zbudować system umożliwiający interpretację gestów przez komputer, a być może w przyszłości nawet rozumienie wypowiedzi [22], w pierwszej kolejności należy rozwiązać problem akwizycji danych. Wiele takich systemów korzysta z akcelerometrów [16] lub specjalnie skonstruowanych rękawic sensorycznych [49] (ang. *data gloves*, *cyber gloves*) jako źródła informacji o pozycjach, orientacjach i kształcie dłoni. Wadą rozwiązań jest wpływ na wykonywane gesty przez krępowanie swobody dłoni. Ponadto rękawice posiadają ustalony rozmiar oraz nie zawsze są dostępne. Innym podejściem do akwizycji danych jest rejestrowanie sygnałów elektrycznych pochodzących z mięśni przez aktywne elektrody umieszczone na powierzchni skóry (elektromiografia EMG). Zastosowanie EMG umożliwia wykrycie nawet bardzo drobnych ruchów dłoni. Wadą rozwiązania jest wymóg uwzględniania efektu zmęczenia oraz trudności w wykonaniu klasyfikatora w przypadku rozpoznawania dużego słownika gestów [10]. Biorąc pod uwagę ograniczenia powyższych rozwiązań, wielu badaczy stosuje wizję komputerową do wykrywania i śledzenia dłoni, posiłkując się informacjami o kolorze, ruchu, bądź krawędziach [24]. By ułatwić odróżnienie dłoni od otoczenia, część autorów stosuje kolorowe rękawiczki lub markery [3]. Wiele prac opiera się także na wykrywaniu obiektów w kolorze skóry [27]. Innym, stosunkowo nowym podejściem, jest stosowanie kamer aktywnych (Time of Flight, Kinect) [44, 51] w celu uzyskania dodatkowej informacji o głębi i tym samym o kształcie przestrzennym dłoni. Alternatywnym podejściem jest zastosowanie systemu stereowizyjnego [8, 15, 51].

Kolejnym krokiem po detekcji dłoni jest ich śledzenie. Trudność w tym zadaniu polega na potrzebie nadążania za szybko wykonywanymi gestami, częstymi zmianami pozycji dłoni, ich

kształtu oraz orientacji. Dodatkowym utrudnieniem jest możliwość wzajemnego przesłaniania dłoni lub występowania ich na tle twarzy. Ostatnie zagadnienie jest ważne w przypadku detekcji opartej o kolor skóry.

Dwa główne podejścia do klasyfikacji gestów spotykane w literaturze wykorzystują różne wariacje sieci neuronowych [18, 54] oraz ukrytych modeli Markowa [32]. Sieci neuronowe [11] są często używane do klasyfikacji statycznych kształtów dłoni [51]. W nowszej literaturze można również spotkać podejścia wykorzystujące boosting [6, 12], a także zastosowanie nieliniowej transformacji czasowej DTW [20, 50] i klasyfikatora najbliższego sąsiada [34].

Oprócz prac dotyczących klasyfikatorów wykorzystujących modele całych słów można napotkać mniej liczne prace z użyciem jednostek mniejszych niż słowa (cheremy, ang. *subunits*) [2, 6, 34, 45, 49]. Takie podejście przypomina modelowanie wyrażeń języka mówionego za pomocą fonemów [48]. Część badaczy próbowała podziału gestów na cheremy modelując je za pomocą modeli Markowa [17]. Inni korzystają z modelu lingwistycznego zakładając, że gest składa się z ruchów i zatrzymań [49] lub dzielą szeregi czasowe wektorów cech na fragmenty, wyszukując punkty nieciągłości trajektorii ruchu dłoni [2].

Mimo że stosowanie jednostek mniejszych niż słowa należy do najnowszego nurtu badań nad rozpoznawaniem języka migowego, to wciąż tylko kilka prac wskazuje sposób wykorzystania cheremów do rozpoznawania dłuższych wypowiedzi [1] lub rozszerzania słownika budując modele nowych gestów z dostępnych cheremów [4]. Kolejnym, rzadko przedstawianym w literaturze zagadnieniem badawczym jest rozpoznawanie gestów na podstawie małej liczby wykonań uczących [21].

Podzbiorem języka migowego jest język palcowy, którego rozpoznawanie przedstawiono w pracach [10, 14, 23], służy on do przekazywania nazw własnych, skrótowców, czy literowania wyrazów. Odwrotnym zagadnieniem jest wyświetlanie gestów w języku migowym wykonywanych przez awatar [13, 39].

W rozprawie przedstawiono tabelę z porównaniem skuteczności rozpoznawania języków migowych.

Na podstawie przeglądu literatury można sformułować następujące wnioski:

- Języki migowe różnych krajów wykazują znaczącą odmienność.
- Informacje dotyczące szczegółów rozwiązań prezentowanych w literaturze są najczęściej niedostępne, dlatego też trudno ocenić uniwersalność zastosowanych podejść w odniesieniu do rozważanych języków migowych.
- Zachodzi potrzeba badań nad rozpoznawaniem języka migowego w Polsce z uwzględnieniem następujących zagadnień:
  - określenie sposobu definiowania jednostek mniejszych niż słowa (cheremów) na podstawie analizy danych,
  - synteza klasyfikatora, opartego na wykorzystaniu reprezentacji gestów za pomocą cheremów, do rozpoznawania wyrazów i zdań,
  - ocena możliwości rozszerzania słownika o nowe gesty przy użyciu modeli cheremów,
  - przygotowanie środowiska wspomagającego eksperymenty i gromadzenie oraz udostępniane danych pozwalającego na weryfikację zaproponowanych rozwiązań.

1.3 Cel, zakres i teza pracy

Celem pracy jest opracowanie metody rozpoznawania słów i pojedynczych zdań polskiego języka miganego na podstawie analizy sekwencji wizyjnych. Większość wypowiedzi to gesty

dynamiczne, które w wizyjnych systemach rozpoznawania są reprezentowane przez szeregi czasowe, tzn. przebiegi zmienności cech wyznaczonych na podstawie analizy obrazów. Przedmiotem badań jest metoda rozpoznawania wypowiedzi wykorzystująca modelowanie gestów za pomocą jednostek mniejszych niż słowa (cheremów). Przypomina to modelowanie za pomocą fonemów w przypadku języka mówionego. Nie wiadomo dokładnie, co w wypowiedzi przedstawianej za pomocą gestów stanowi odpowiedniki fonemów (cheremy). Proponowana metoda wyodrębnienia cheremów opiera się na analizie danych. Polega na określeniu sposobu segmentacji szeregów czasowych reprezentujących wypowiedzi, by powstałe fragmenty – traktowane jako poszukiwane cheremy – tworzyły jednorodne grupy (klastry). Punkty podziału są wyznaczane jako rozwiązanie zadania optymalizacji, znajdowane z wykorzystaniem ewolucyjnej procedury opartej na algorytmie immunologicznym.

W pracy położono nacisk na ocenę wpływu metod określania podobieństwa między cheremami, metod grupowania i wskaźników oceny klastrów, technik optymalizacji oraz typu klasyfikatora na skuteczność rozpoznawania. Integralną część pracy stanowi przygotowanie środowiska wspomagającego eksperymenty i gromadzenie oraz udostępnianie danych.

Zaplanowane, główne rezultaty rozprawy to:

1. Metoda wyznaczania cheremów oparta na grupowaniu szeregów czasowych,
2. Metoda modelowania słów i prostych zdań z wykorzystaniem cheremów,
3. Metoda rozpoznawania słów i prostych zdań z wykorzystaniem opracowanych modeli,
4. Środowisko wspomagające eksperymenty i gromadzenie oraz udostępnianie danych,
5. Wyniki eksperymentów weryfikujących opracowane metody.

Tezę pracy można sformułować następująco:

Zastosowanie grupowania szeregów czasowych do automatycznej segmentacji przebiegów zmienności cech, otrzymanych na podstawie analizy sekwencji obrazów rejestrujących wypowiedzi w języku migowym, pozwala wyznaczyć składniki gestów, za pomocą których można modelować wypowiedzi w celu ich skutecznego rozpoznawania.

2 Grupowanie szeregów czasowych reprezentujących wypowiedzi języka migowego

W rozdziale omówiono problematykę grupowania szeregów czasowych wykorzystując dane reprezentujące powtórzone wielokrotnie wypowiedzi polskiego języka miganego w formie 101 słów i 35 zdań. Dane otrzymano na podstawie zarejestrowanych sekwencji wizyjnych. Znajomość przynależności badanych szeregów czasowych do klas pozwoliła ocenić skuteczność zastosowanych metod grupowania i wykorzystywanych w nich sposobów określania podobieństwa.

2.1 Przetwarzanie obrazów i wyznaczanie wektora cech

Każdy gest SJM może być scharakteryzowany przez trzy następujące komponenty [38, 47]: (i) umiejscowienie wykonania względem ciała, (ii) kształt dłoni, (iii) ruch dłoni. Mimo że w praktycznej komunikacji za pomocą języka migowego używa się dodatkowych cech, takich jak kształt ust, czy wyraz twarzy, w niniejszej pracy nie są one uwzględniane.

Tabela 1: Wektor cech opisujący dłoń

Numer cechy	Nazwa cechy	Oznaczenie	Sposób wyznaczania jeżeli dotyczy
1	Położenie środka ciężkości – składowa pozioma	x	-
2	Położenie środka ciężkości - składowa pionowa	y	-
3	Informacja o głębi	ΔZ	Różnica średnich wartości głębi dłoni i twarzy
4	Pole powierzchni	$\tilde{S}$	-
5	Współczynnik zwartości	γ	$\gamma = \frac{\tilde{P}^2}{4\pi\tilde{S}}$, $\tilde{P}$ jest obwodem dłoni
6	Niecentryczność	ϵ	$\epsilon = \frac{(m_{20}-m_{02})^2+4m_{11}^2}{\tilde{S}^4}$, m_{11}, m_{20}, m_{02} – momenty centralne [41]
7	Orientacja	Ψ	$\Psi = 0.5 \operatorname{arc\,tg}[\frac{\tilde{P}^2}{m_{20}-m_{02}}]$

Ponieważ podejmowane prace badawcze w przeważającej większości obejmują wyszukiwanie podobnych fragmentów szeregów czasowych (nazywanych cheremami, zob. następne podrozdziały) i modelowanie oraz rozpoznawanie za ich pomocą dłuższych szeregów reprezentujących całe wyrażenia w języku migowym, do badań wykorzystano zapisane w postaci szeregów czasowych wektorów cech sekwencje wideo wykonane podczas badań opisanych w pracy [51]. We wspomnianych badaniach zastosowano system stereowizyjny. Identyfikacji pikseli należących do dłoni i twarzy dokonywano na podstawie kolorowego obrazu z kamery przyjętej jako referencyjna. W wyniku zastosowania metody uwzględniającej chrominancję skóry ludzkiej otrzymuje się obrazy binarne [42] zawierające trzy lub dwa obiekty o dominujących rozmiarach. W celu rozróżnienia, który z obiektów odpowiada dłoni prawej, lewej i twarzy, zastosowano heurystyczny algorytm rozróżniający dłoń lewą, prawą oraz twarz w obrazie binarnym, wykorzystujący informację o pozycjach i polach powierzchni obiektów w bieżącej i poprzedniej ramce. Obrazy binarne oraz mapy dysparycji [8] wykorzystano do budowy wektorów cech [15, 51]. Przyjęte składowe wektora cech można podzielić na cztery grupy: (1) opisującą położenie obu dłoni, (2) opisującą kształt dłoni, (3) zawierającą informacje o orientacji dłoni, (4) zawierającą informację przestrzenną.

W tabeli 1 przedstawiono składowe wektora cech (7 cech na dłoń, razem 14 cech), jako motywację stojącą za jego wyborem można wyróżnić: (i) analiza selektywności segmentacji dłoni i usystematyzowane badania oceniające wpływ cech na skuteczność rozpoznawania w teście walidacji krzyżowej z zastosowaniem klasyfikatora wykorzystującego ukryte modele Markowa przedstawione w pracy [51], (ii) cechy mają za zadanie odwzorowywać informacje, które pozwalają na opisanie gestu zgodnie ze wskazówkami lingwistów [38], (iii) zbliżone i arbitralnie wybrane wektory cech można napotkać w reprezentatywnych pracach z zakresu rozpoznawania języka migowego, np. [3, 21, 33].

2.2 Grupowanie i porównywanie szeregów czasowych

Gesty języka migowego w większości są wykonywane obiema rękami i są dynamiczne [38]. Niech $S = \{X_1, X_2, \ldots, X_n\}$ będzie zbiorem danych, gdzie sekwencja $X_i = \{x_i(1), x_i(2), \ldots, x_i(T_i)\}$

rzeczywistoliczbowych wektorów reprezentuje wyrażenie (słowo lub zdanie) w języku migowym. Wszystkie wektory $x_i(t)$, gdzie $i \in I = \{1, 2, \ldots, n\}$ i t jest czasem próbkowania, $t \in \mathcal{T}_i = \{1, 2, \ldots, T_i\}$, są wyznaczone na podstawie obrazów zarejestrowanych przez kamerę. Chcąc je grupować, w pierwszej kolejności należy wskazać sposób ich porównywania.

Celem grupowania (klasteryzacji) jest podział zbioru na podzbiory podobnych do siebie elementów [46, 52]. Najczęściej trudno jest określić a priori faktyczną liczbę tych podzbiorów oraz jaki jest najbardziej odpowiedni sposób grupowania odkrywający prawdziwy podział na podzbiory.

W przypadku grupowania szeregów czasowych o różnej długości wykonuje się zmianę reprezentacji szeregów do wektorów o równej długości. Proponuje się podejście z obliczaniem tzw. macierzy podobieństwa między wszystkimi parami szeregów [52] i traktowanie wiersza takiej macierzy jako nowej reprezentacji danego szeregu [35]. Można przyjąć, że podobne szeregi czasowe powinny mieć zbliżone wartości w odpowiadających im wierszach macierzy podobieństwa. Do obliczania elementów macierzy podobieństwa zastosowano nieliniową transformację czasową DTW [52] pozwalającą na nieliniowe mapowanie jednej sekwencji liczb w drugą, minimalizując odległość między nimi. Główną motywacją stosowania DTW jest jej zdolność rozszerzania i kompresowania osi czasu, co pozwala na porównywanie sekwencji, które są podobne, ale przesunięte w fazie. Przykładowo, niektóre powiązane części gestów reprezentujące to samo wyrażenie mogą być wykonane z różnymi prędkościami.

Po wyznaczeniu macierzy podobieństwa jako macierzy odległości DTW między szeregami czasowymi charakteryzującymi gesty języka migowego (zob. podrozdział 2.1) otrzymuje się wektory o długości równej liczbie wykonań gestów, tj. dla zbioru n szeregów $S = \{X_1, \ldots, X_n\}$ po obliczeniu odległości DTW d_{DTW} wynikowa macierz podobieństwa $\tilde{W}$ ma postać:

$$\tilde{W} = \begin{pmatrix} d_{DTW}(X_1, X_1) & d_{DTW}(X_1, X_2) & \cdots & d_{DTW}(X_1, X_n) \\ d_{DTW}(X_2, X_1) & d_{DTW}(X_2, X_2) & \cdots & d_{DTW}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ d_{DTW}(X_n, X_1) & d_{DTW}(X_n, X_2) & \cdots & d_{DTW}(X_n, X_n) \end{pmatrix} = \begin{pmatrix} \tilde{W}_1 \\ \tilde{W}_2 \\ \vdots \\ \tilde{W}_n \end{pmatrix}$$

Do grupowania wykorzystano algorytm k-średnich [52]. Zwykle pracuje on na wektorach obliczając wektory średnich zawierające wartości średnie dla każdego wymiaru. Podobieństwo grupowanego wektora do wektora średnich obliczane jest najczęściej za pomocą odległości euklidesowej. W tym miejscu warto zauważyć wrażliwość algorytmu na losowe warunki początkowego przydziału wektorów do grup oraz konieczność normalizacji wektorów przed grupowaniem, by wszystkie wymiary mogły mieć wpływ na wynik.

Odmianą algorytmu k-średnich charakteryzującą się mniejszym wpływem wektorów odstających oraz brakiem potrzeby obliczania wektorów średnich jest algorytm k-medoidów. W algorytmie tym, w każdej iteracji wybierany jest wektor reprezentujący środek klastra jako np. ten, którego suma odległości do innych wektorów w klastrze jest najmniejsza. Warto zauważyć, że medoid, w odróżnieniu od średniej jest zawsze elementem klastra.

Innym algorytmem grupowania, którego liczne testy porównawcze zawarto w pracy [19] wykazując posiadanie zdolności odkrywania istniejącego, naturalnego podziału danych poddanych grupowaniu, jest algorytm minimalnej entropii MEC (ang. *minimum entropy clustering*) [26]. Algorytm MEC wymaga początkowego przypisania elementów do klastrów, więc startuje z rozwiązania otrzymanego po zadanej liczbie iteracji algorytmu k-średnich i poprawia je. Algorytm sprawdza, czy większość sąsiadów danego elementu należy do jego grupy. Następnie przenosi go do owej grupy, jeżeli spowoduje to zmniejszenie całkowitej entropii zbioru grup. Warunkiem zakończenia algorytmu jest brak zmian przynależności elementów do grup w iteracji. Ważnym aspektem algorytmu jest jego zdolność do zmniejszenia liczby grup, ponieważ

wektory mogą migrować między grupami. Grupy puste są opuszczane, a zadana liczba grup odpowiednio pomniejszana. Dlatego też zaleca się zadawać większą liczbę grup niż oczekiwana, by w nadmiarowych grupach znalazły się ewentualne elementy odstające. Przy dużej liczbie grupowanych danych algorytm rzadko zmniejsza liczbę grup.

Każdy algorytm grupowania lub różne uruchomienia tego samego algorytmu mogą prowadzić do odmiennych rezultatów, z tego powodu wyniki często porównuje się korzystając z tzw. wskaźników jakości [25, 52]. W literaturze można spotkać wiele wskaźników jakości grupowania. W pracy rozważano kilka najbardziej znanych wskaźników: (i) Dunna, (ii) Daviesa–Bouldina, (iii) I, (iv) RAND.

Ponieważ początkowy przydział elementów do klastrów ma zasadniczy wpływ na rezultat zastosowanych metod, zdefiniowano problem optymalizacji, w którym poszukuje się przydziału elementów do grup, zaś funkcją celu jest kryterium grupowania. Do rozwiązania problemu zastosowano algorytm immunologiczny CLONALG [9].

2.3 Eksperymenty

Eksperymenty dotyczące algorytmów grupowania i wskaźników oceny przeprowadzono na wyrażeniach polskiego języka miganego (SJM), dalej zamieszczono wyniki przykładowego eksperymentu.

W tabeli 2 zawarto otrzymane wartości wskaźników jakości. Algorytmy grupowania były inicjowane losowo lub znanym podziałem wyrażeń, tj. podziałem zgodnym z intencją osoby wykonującej gest (podział wyróżniono literami ZP od *znany podział*). Przedstawiono również wartości wskaźników dla znanego podziału wyrażeń. Ponieważ początkowy przydział elementów do grup ma wpływ na rezultaty grupowania, obliczenia z udziałem algorytmów grupowania powtarzano stokrotnie. Oprócz wskaźników jakości klastrów tabela uwzględnia kryteria grupowania właściwe dla użytych metod. Dla algorytmu MEC jest to entropia, zaś dla algorytmów k–medoidów i k–średnich - średnia odległość wektorów od środka grupy. Badania przeprowadzono na macierzy podobieństwa obliczonej przy pomocy metody DTW. Ponieważ dane badawcze zawierają 14–wymiarowe wektory cech, do porównywania próbek szeregów czasowych w algorytmie DTW posłużono się 14–wymiarową metryką euklidesową. Wartości wektorów cech zostały wcześniej znormalizowane do wartości z przedziału $[0, 1]$ z uwzględnieniem wszystkich przebiegów. Macierz zawiera 4040 wektorów podobieństwa o długości 4040 dla wyrazów SJM i 1400 wektorów podobieństwa o długości 1400 dla zdań SJM.

W badaniach wykorzystano także wektory podobieństwa zredukowane przy użyciu analizy głównych składowych PCA (ang. *Principal Component Analysis*) [46].

Wyniki dla PCA odnoszą się do redukcji wymiaru wektorów podobieństwa wyrazów i zdań do 30. Liczbę grup w badaniach przyjęto odpowiednio równą liczbie przetwarzanych wyrazów i zdań SJM.

Tabela 2: Wartości średnie kryteriów jakości oceny grupowania oraz kryteria wewnętrzne rozważanych algorytmów grupowania otrzymane w eksperymentach. Algorytmy grupowania były inicjowane losowo lub znanym podziałem wyrażeń SJM (ZP). Eksperymenty uwzględniają redukcję wymiaru wektorów podobieństwa przez PCA. Dla wygody wartości maksymalizowanych kryteriów pomnożono przez -1

Metoda	Czas [ms]	Wskaźniki jakości klastrów				Kryteria wewnętrzne algorytmów		
		DB	I	Dunn	RAND	K-średnich	MEC	K-medoidów
Wyrazy								
Znany podział (ZP)	-	2.24	-3.05E-09	-0.039	-1	7.675	0.0365	0.0786
ZP + k-średnich	39874	2.11	-3.25E-09	-0.003	-0.9902	4.432	0.0584	0.1034
ZP + k-medoidów	187	1.66	-3.99E-09	-0.01	-0.9984	7.594	0.0478	0.0754
ZP + MEC	2776	2.25	-3.05E-09	-0.039	-0.9997	7.546	0.0217	0.0789
K-średnich	66772	2.39	-3.55E-09	-0.004	-0.9863	4.507	0.0648	0.1149
K-medoidów	233	4.55	-2.52E-09	-0.006	-0.9839	15.777	0.1491	0.1217
MEC	50255	2.3	-3.60E-09	-0.005	-0.9863	4.53	0.0329	0.1147
K-średnich, PCA	386	1.27	-3.90E-06	-0.016	-0.9862	4.501	0.0668	0.116
K-medoidów, PCA	183	1.57	-3.26E-06	-0.005	-0.981	7.454	0.1323	0.1373
MEC, PCA	710	1.26	-3.88E-06	-0.028	-0.9862	4.521	0.0481	0.1154
Zdania								
Znany podział (ZP)	-	1.58	-6.82E-08	-0.078	-1	1.524	0.0188	0.1098
ZP + k-średnich	1544	1.65	-9.27E-08	-0.065	-0.9802	0.864	0.0005	0.1248
ZP + k-medoidów	47	1.59	-6.81E-08	-0.01	-0.9978	1.529	0.0179	0.1096
ZP + MEC	312	1.56	-6.94E-08	-0.037	-0.9998	1.524	0.0182	0.1098
K-średnich	2407	1.85	-9.74E-08	-0.077	-0.9696	0.926	0.002	0.1365
K-medoidów	35	5.33	-7.82E-08	-0.034	-0.9636	2.177	0.0296	0.1482
MEC	1665	1.83	-9.88E-08	-0.08	-0.9698	0.923	0.0013	0.1361
K-średnich, PCA	78	1.18	-1.27E-05	-0.075	-0.9697	0.923	0.0022	0.1365
K-medoidów, PCA	30.6	1.45	-1.11E-05	-0.031	-0.9598	1.224	0.0074	0.1462
MEC, PCA	541	1.19	-1.30E-05	-0.09	-0.9702	0.922	0.0016	0.1361

Stosowanie wektorów zredukowanych przez PCA do długości 30 znacząco skraca czas obliczeń w przypadku algorytmów, które operują na wektorach (k–średnich i MEC). Najkrótszym czasem obliczeń wykazał się algorytm k–medoidów. Można zauważyć, że każdy z algorytmów zmienił podział początkowy, przyjęty jako *znany podział*, poprawiając wartość kryterium, które minimalizuje. Wartość kryterium każdego z algorytmów startującego ze znanego przydziału wyrażeń do klas okazała się znacznie lepszą niż wynik grupowania startujący z przydziału losowego.

Wartości wskaźników jakości klastrów DB, I, Dunn odnotowane w badaniach nie faworyzują żadnego wskaźnika oceny, ani metody grupowania. Spośród metod grupowania algorytm MEC nieco częściej był oceniany najlepiej. Wartość wskaźnika RAND wskazuje, że algorytm MEC we wszystkich przypadkach pogrupował dane w sposób, który można uznać za najbliższy znanemu podziałowi wyrażeń SJM. Dlatego też, mimo dłuższego czasu obliczeń w porównaniu z algorytmem k-medoidów, algorytm MEC wydaje się w najlepszy sposób odnajdywać naturalny podział analizowanych danych. W tym miejscu warto dodać, że wskaźnik RAND wykazuje niedoskonałość wynikającą z małego zróżnicowania wyników w przypadku dużej liczby grup.

2.4 Podsumowanie

Przedstawiono wyniki eksperymentów dotyczących grupowania wykonań wyrażeń polskiego języka miganego. Szeregi czasowe reprezentujące gesty, nawet dla tych samych wyrażeń, mają zróżnicowaną długość, dlatego do ich porównywania wykorzystano metodę nieliniowej transformacji czasowej DTW budując na jej podstawie macierz podobieństwa. Powstałe wektory podobieństwa, ze względu na dużą długość, redukowano za pomocą metody składowych głównych PCA. Zaobserwowano i omówiono problemy związane ze skalą oraz wymiarowością przetwarzanych danych. Porównano wybrane metody grupowania (k–średnich, k–medoidów, minimalnej entropii MEC) stosując popularne wskaźniki oceny jakości klastrów i wskazano algorytm minimalnej entropii MEC jako najlepiej odkrywający naturalny podział wykorzystywanych danych, zwłaszcza w przypadku małego słownika gestów. Ponieważ każda z użytych metod grupowania optymalizuje swój wskaźnik jakości znajdując rozwiązanie lokalne, zmierzając do uzyskania rozwiązań globalnych wykorzystano algorytm immunologiczny CLONALG. Wykazano, że poprawiając sposób początkowego podziału szeregów czasowych na klastry otrzymuje się wynik zbliżony do podziału naturalnego. Przy zastosowaniu metody ewolucyjnej z dużą populacją rozwiązań początkowe przypisanie elementów do klastrów nie ma znaczącego wpływu na rezultat grupowania. Wykonano również analizę błędów grupowania, które mogą wskazywać na trudność poprawnego rozpoznawania.

3 Rozpoznawanie wypowiedzi w języku migowym

Rozpoznawanie wypowiedzi w języku migowym, wykorzystujących duży zasób słów, wymaga modelowania wyrazów za pomocą mniejszych jednostek. Przypomina to modelowanie za pomocą fonemów w przypadku języka mówionego. Problem polega na wyodrębnieniu odpowiedników fonemów (cheremów) na podstawie przebiegów czasowych wektorów cech otrzymanych w układzie wizyjnym, a następnie na wykorzystaniu cheremów w zadaniu rozpoznawania.

W modelowaniu wyrażeń języka migowego należy wziąć pod uwagę jednoczesną sekwencyjność i symultaniczność obserwowanych cech. Przykładowo, kształt dłoni i jej pozycja mogą zmieniać się niezależnie w tym samym czasie [17]. Chcąc modelować procesy równoległe rozróżnia się N grup cech (kanałów), opierając się na założeniu, że odrębne procesy rozwijają się niezależnie od siebie i posiadają niezależne wyjścia. Podczas wyodrębniania cheremów wszystkie elementy w grupie będą rozważane łącznie, natomiast różne grupy będą rozważane oddzielnie. Można np. rozróżnić dwa

niezależne kanały ($N = 2$) związane z dłońmi lub 14 niezależnych kanałów ($N = 14$) związanych z cechami, wymienionymi w tabeli 1.

3.1 Wyznaczanie cheremów

W celu wyznaczenia cheremów rozważa się dekompozycję D^l względem czasu, która dla każdego $i \in I$, definiuje liczbę $k_i^l = k_i^l(D^l) \geqslant 1$ i $k_i^l - 1$ miejsc cięcia $t_{ij}^l = t_{ij}^l(D^l)$, gdzie $1 < t_{i1}^l < t_{i2}^l < \cdots < t_{i,k_i^l-1}^l < T_i$. Dekompozycja oznacza, że X_i^l będzie podzielony na k_i^l krótszych szeregów. Pierwszy szereg $s_{i1}^l(D^l)$ rozpoczyna się w $t = 1$ i kończy w $t = t_{i1}^l$, następny szereg $s_{i2}^l(D^l)$ zaczyna się w $t = t_{i1}^l$ i kończy w $t = t_{i2}^l$ i tak dalej aż do ostatniego szeregu $s_{i,k_i^l}^l(D^l)$, który zaczyna się w $t = t_{i,k_i^l-1}^l$ i kończy w T_i. Wynikowy zbiór danych $S'^l(D^l) = \{s_1'^l(D^l), s_2'^l(D^l), \ldots, s_n'^l(D^l)\}$, gdzie $s_i'^l(D^l) = \{s_{i1}^l(D^l), \ldots, s_{i,k_i^l}^l(D^l)\}, i \in I$, zawiera $n'^l = n'^l(D^l) = \sum_{i=1}^n k_i^l(D^l)$ krótszych szeregów. Długość każdego takiego szeregu jest ograniczona przez minimalną L_{min}^l i maksymalną L_{max}^l liczbę kolejnych próbek.

Proponuje się określanie dobrej dekompozycji szeregów poprzez rozwiązanie problemu decyzyjnego w oparciu o następujące kroki:

1. Podział zbioru $S'^l(D^l)$ na zadaną liczbę m^l grup, tj. $S'^l(D^l) = \{C_1^l(D^l), C_2^l(D^l), \ldots, C_{m^l}^l(D^l)\}$,
2. Ocena podziału D^l na podstawie kryterium $J(D^l)$, które charakteryzuje jakość otrzymanych grup.

Kryterium może być jeden ze wskaźników jakości grupowania lub kryterium metody realizującej grupowanie.

W celu rozwiązania tak zdefiniowanego problemu optymalizacji [28, 29, 30, 31] odpowiednio przystosowano dwa algorytmy biologiczne: algorytm selekcji klonalnej CLONALG i algorytm genetyczny.

W wyniku optymalizacji otrzymuje się dobrą dekompozycję D_{opt}^l. Można ją wykorzystać transformując każdy szereg X_i^l w ciąg etykiet $X_i^{ls} = \{e_{i1}^l, e_{i2}^l, \ldots, e_{i,ki}^l\}$, gdzie $e_{ik}^l \in E^l = \{\alpha_1^l, \alpha_2^l, \ldots, \alpha_{m^l}^l\}$, α_k^l oznacza etykietę przydzieloną $C_k^l(D_{opt}^l)$, a e_{ik}^l jest etykietą klastra, do którego należy fragment $s_{ik}^l(D_{opt}^l)$. Odpowiednik X_i zapisany w formie łańcucha będzie oznaczany przez X_i^s, tzn. $X_i^s = \{X_i^{1s}, X_i^{2s}, \ldots, X_i^{Ns}\}$, a w konsekwencji S^s będzie rozumiany jako odpowiednik zbioru S.

3.2 Rozpoznawanie

Chcąc określić odległość między dwoma szeregami stosowano dwa podejścia: (i) oparte na nieliniowej transformacji czasowej DTW, (ii) oparte na prostych wektorach charakteryzujących szeregi.

W drugim podejściu szereg czasowy Q jest zastępowany (reprezentowany) przez wektor v_Q zawierający wartość średnią i odchylenie standardowe. Zakładając, że elementy *q(i)* wektora Q są wektorami p-wymiarowymi, otrzymuje się wymiar równy $2p$. Odpowiedni $2p$-wymiarowy wektor v_R reprezentujący szereg R jest tworzony w analogiczny sposób. Odległość między szeregami Q i R jest zdefiniowana jako odległość między wektorami v_Q i v_R.

Pierwsze podejście jest stosowane w fazie rozpoznawania, drugie zaś tylko w procedurze grupowania podczas określania cheremów w zadaniu optymalizacji. Jeżeli grupowanie jest oparte na wektorowej reprezentacji elementów takich, jak k-średnich, stosowanie DTW wymaga długich wektorów. W takim przypadku każdy szereg jest reprezentowany przez wektor odległości do

innych szeregów. Operacje na takich wektorach są czasochłonne. Ma to zasadnicze znaczenie w aplikacji, w której grupowanie jest wykonywane wielokrotnie. Można wtedy stosować krótsze wektory utworzone przez PCA lub zmienić metodę grupowania na np. k-medoidów. Jeszcze innym rozwiązaniem mogłaby być reinterpolacja wszystkich szeregów do jednakowej długości. Oczekiwane cheremy są relatywnie krótkie, co usprawiedliwia proponowane podejście z wektorami v_Q i v_R. Jest ono proste i znacząco przyspiesza obliczenia. Ponadto, jak to zostanie pokazane w rozdziale 4, wykorzystywanie uzyskanych tą drogą cheremów daje bardzo dobrą skuteczność rozpoznawania.

Cheremy mogą być wybrane na dwa sposoby [31]: jako reprezentanci klastrów zawierających krótkie szeregi lub ukryte modele Markowa (HMM) takich klastrów. Załóżmy, że wyrażenie do sklasyfikowania jest reprezentowane przez szereg $Y = \{y(1), y(2), \ldots, y(T_y)\}$. Wektory cech $y(.)$ mają taką samą strukturę jak $x(.)$ i dlatego też szeregi $Y^l = \{y^l(1), y^l(2), \ldots, y^l(T_y)\}$, gdzie $l \in \mathcal{N} = \{1, 2, ..., N\}$, będą rozpatrywane osobno.

W przypadku gdy cheremy są reprezentantami klastrów, należy rozwiązać dwa problemy. Pierwszy problem polega na odnalezieniu odpowiedniej reprezentacji znakowej (transkrypcji) Y^l, tj. $Y^{ls} = \{e^l_{y1}, e^l_{y2}, \ldots, e^l_{y,k^l_y}\}$, gdzie $e^l_{yk} \in E^l$ i, konsekwentnie, reprezentacji Y^s szeregu Y. Drugim problemem jest znalezienie $NN(Y^s)$ – najbliższego sąsiada Y^s w zbiorze S^s. Wtedy nieznane wyrażenie zostanie przypisane do klasy, do której należy $NN(Y^s)$. Transkrypcję odnajduje się rozwiązując zadanie optymalizacji w odniesieniu do punktów cięcia w Y^l dla każdego $l \in \mathcal{N}$. Niech $D^l_y = [t^l_{y1}, t^l_{y2}, \ldots, t^l_{y,k^l_y-1}]$ charakteryzuje dekompozycję. W przeciwieństwie do poprzedniej optymalizacji wykorzystywanej do wyznaczenia cheremów, w tym przypadku kryterium jest

$$J(D^l_y) = \sum_{k=1}^{k^l_y} d_{DTW}(k)$$

gdzie $d_{DTW}(k)$ oznacza odległość DTW między k–tym szeregiem $s^l_{y,k}(D^l_y)$ w Y^l i jego najbliższym sąsiadem $NN(s^l_{y,k}(D^l_y))$ w zbiorze $S'^l(D^l_{opt})$. Zadanie optymalizacji może być rozwiązane np. przez algorytm CLONALG. Wtedy e^l_{yk} jest etykietą klastra, do którego należy $NN(s^l_{y,k}(D^l_{y,opt}))$. Procedura jest powtarzana dla każdego $l \in \mathcal{N}$. Drugi problem to też zadanie optymalizacji. Tutaj, tzw. odległość edycji jest wykorzystywana do obliczenia odległości między dwiema transkrypcjami (łańcuchami znakowymi). Miarą podobieństwa między szeregami Y i X_i jest suma

$$\hat{d}_i = \sum_{l=1}^{N} w^l \hat{d}^l_i$$

gdzie w^l oznacza wagę przydzieloną do $l-tej$ składowej wektora cech, zaś $\hat{d}^l_i$ jest odległością edycji $d_{ED}(Y^{ls}, X^{ls}_i)$ między łańcuchami znaków Y^{ls} oraz X^{ls}_i. W szczególności wszystkie wagi są równe jeden. Szereg Y zostaje przydzielony do klasy, do której należyX_j, gdzie $j = arg\ min_{i \in I}(\hat{d}_i)$.

Druga reprezentacja cheremów wykorzystuje HMM, które są wyuczone na podstawie klastrów zawierających krótkie szeregi. Do zaprojektowania modeli cheremów opartych na HMM autor wykorzystał oprogramowanie HTK [53].

Ukryty model Markowa HMM (ang. *Hidden Markov Model*) [46, 52, 55] opisuje proces stochastyczny jako sekwencję nieobserwowalnych stanów generujących obserwacje. Można wyróżnić dwa procesy. Pierwszy - łańcuch Markowa ze skończoną liczbą stanów - charakteryzuje się rozkładem prawdopodobieństwa stanu początkowego i macierzą prawdopodobieństw przejść między stanami, drugi zaś jest ciągiem obserwacji generowanym przez stany zgodnie z danymi rozkładami prawdopodobieństwa. Wykorzystując HMM można reprezentować rozkłady prawdopodobieństwa w ciągach obserwacji. Jako obserwację traktuje się symbol z dyskretnego alfabetu, zmienną rzeczywistą lub obiekt, nad którym jest możliwe zdefiniowanie rozkładu prawdopodobieństwa. Chcąc zastosować HMM o zadanej strukturze, należy poznać: (i) rozkład prawdopodobieństwa

stanu początkowego, (ii) prawdopodobieństwa przejść między stanami, (iii) model obserwacji. Dzięki swoim właściwościom modelowania sekwencji obserwacji modele Markowa są często wykorzystywane w różnych zagadnieniach związanych ze sztuczną inteligencją tj. w modelowaniu mowy, rozpoznawaniu pisma ręcznego, rozpoznawaniu wzorców, czy nawet klasyfikacji dźwięków muzycznych [46].

Ponieważ do modelowania wyrazów języka migowego zwykle wystarczą dwustanowe modele Markowa [51], do modelowania grupy krótkich szeregów czasowych wykorzystano model Bakisa z jednym stanem emitującym i dwoma stanami nieemitującymi, realizującymi wejście i wyjście modelu Markowa. Rozkład prawdopodobieństw emisji obserwacji każdego ze stanów modeli HMM opisano za pomocą rozkładu Gaussa.

Wyrażenia w języku migowym są rozpoznawane z wykorzystaniem połączonego modelu utworzonego jako sieć prostych modeli. Schemat wykorzystuje statystyczną informację o prawdopodobieństwach przejść między dwoma kolejnymi cheremami, obliczoną dla każdego cheremu w relacji do każdego poprzedzającego cheremu w słowniku uczącym (model języka bigram [17, 55]). Parsowanie zostało wykonane z zastosowaniem algorytmu Viterbiego opartego na przekazywaniu znaczników. Modelowanie przebiega w dwóch krokach. W pierwszym izolowane modele są trenowane korzystając z algorytmu Viterbiego i wybranych danych uczących. Następnie parametry modeli są poprawiane w oparciu o całe słowa lub zdania. HTK oferuje opcję *embedded training*, która to umożliwia. *Embedded training* [53] wykorzystuje te same procedury co dla izolowanych modeli, ale zamiast trenować każdy model osobno, trenuje je jednocześnie. Lokalizacja granic cheremów w tym wariancie nie jest konieczna, gdyż wystarczy symboliczna transkrypcja wykonań uczących. Transkrypcja jest otrzymywana podczas opisanego wcześniej procesu pozyskiwania cheremów. Sieć elementarnych modeli Markowa reprezentujących całe wyrażenia jest tworzona automatycznie.

4 Eksperymenty dotyczące rozpoznawania wypowiedzi języka migowego

W rozdziale przedstawiono rezultaty badań dotyczących rozpoznawania wyrazów i zdań polskiego języka miganego (SJM) w oparciu o szeregi czasowe wektorów cech pozyskanych z materiału filmowego z nagraniami gestów (zob. podrozdział 2.1).

Poniżej umieszczono wyniki wybranych badań.

4.1 Walidacja krzyżowa

Do wykonania testów walidacji krzyżowej dane zostały podzielone na 10 rozłącznych podzbiorów. Każdy podzbiór składał się z danych odpowiadających czterem powtórzeniom każdego słowa (po dwa wykonania wykonane przez każdego lektora). Wykonano dziesięć eksperymentów wykorzystując dziewięć podzbiorów jako zbiory uczące S i pozostały, dziesiąty podzbiór jako zbiór testowy. Z powodu losowej natury metody optymalizacji każdy eksperyment został powtórzony dziesięciokrotnie. Dane w S były użyte do wyodrębnienia cheremów, a pozostałe elementy były rozpoznawane.

Wyniki walidacji dla rozwiązania korzystającego z modeli całych słów i klasyfikującego je metodą najbliższego sąsiada porównano z podejściem wykorzystującym cheremy. Taka reprezentacja szeregów czasowych pozwoliła uzyskać niemalże stuprocentową skuteczność rozpoznawania zarówno przy metodzie grupowania MEC jak i k-średnich. Podobne rezultaty uzyskano dla podejścia stosującego modele całych słów. Podejście z modelami Markowa klastrów dało nieco słabsze wyniki (93.67%).

Tablica 3: Test walidacji krzyżowej. Badania z wykorzystaniem cheremów powtarzano dziesięciokrotnie ze względu na losową naturę optymalizacji, podejście z klasyfikatorem najbliższego sąsiada i odległością DTW (całe słowa) wykonywano jednokrotnie. Wyniki podano w %

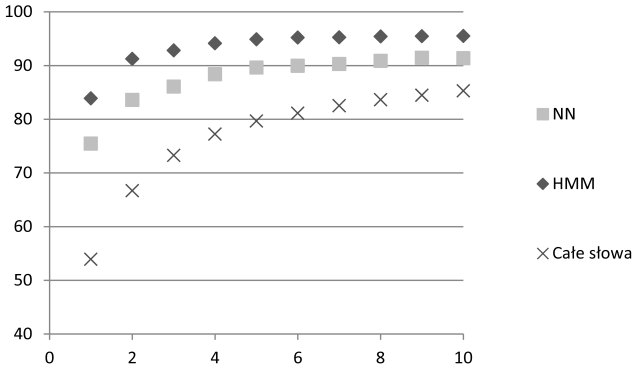
Warianty walidacji krzyżowej										
Podejście z jednostkami mniejszymi niż słowa										
	1	2	3	4	5	6	7	8	9	10
Metoda grupowania MEC, funkcja celu - entropia, średnia skuteczność - **99.03**%, o. std. - 0.8%										
Średnia	96.86	99.28	99.53	99.18	99.70	98.96	98.84	99.58	99.13	99.21
O. std.	0.63	0.27	0.38	0.52	0.16	0.30	0.33	0.23	0.39	0.42
Metoda grupowania MEC, funkcja celu - entropia, szeregi reprezentowane przez wektory podobieństwa DTW, średnia skuteczność - **82.08**%, o. std. - 3.34%										
Średnia	75.47	78.99	80.27	82.92	85.64	85.94	84.90	81.56	80.69	84.38
O. std.	3.06	1.60	1.34	1.28	1.57	1.22	2.00	1.34	2.07	1.81
Metoda grupowania k–średnich, funkcja celu - kryterium k–średnich, średnia skuteczność - **98.88**%, o. std. - 0.78%										
Średnia	96.88	99.11	99.38	99.11	99.58	99.01	98.32	99.41	99.08	98.94
O. std.	0.50	0.42	0.46	0.35	0.17	0.44	0.33	0.42	0.35	0.57
Wykorzystanie modeli Markowa klastrów, metoda grupowania k-średnich, funkcja celu - kryterium k–średnich, średnia skuteczność - **93.67**%, o. std. - 3.7%										
Metoda grupowania MEC, funkcja celu - entropia, szeregi reprezentowane przez wektory podobieństwa DTW, średnia skuteczność - **82.08**%, o. std. - 3.34%										
Średnia	75.47	78.99	80.27	82.92	85.64	85.94	84.90	81.56	80.69	84.38
O. std.	3.06	1.60	1.34	1.28	1.57	1.22	2.00	1.34	2.07	1.81
Średnia	89.70	95.22	84.93	97.00	95.17	96.51	94.03	93.29	94.23	96.66
O. std.	0.97	1.04	1.58	0.77	1.66	0.69	1.10	2.16	1.57	0.82
Podejście wykorzystujące całe słowa										
Szeregi czasowe + DTW, średnia skuteczność - **98.99**%, o. std. - 1.31%										
-	95.30	99.50	99.50	99.01	99.50	99.26	99.26	99.26	99.75	99.50

Średni czas potrzebny na rozpoznawanie jednego gestu ze zbioru testowego wynosił ok. jednej sekundy. Reprezentowanie klastra cheremów z wykorzystaniem medoidu zamiast wszystkich elementów klastrów około czterokrotnie przyspieszało proces optymalizacji opisany w punkcie 3.2 i tym samym skracało czas rozpoznawania do ok. 0.15 s. Rozpoznawanie z wykorzystaniem modeli Markowa klastrów zajęło ok. 0.26 s. Konsekwencją okazało się pogorszenie skuteczności rozpoznawania o ok. 2%.

4.2 Wpływ małej liczby przykładów uczących

Przebadano także skuteczność rozpoznawania izolowanych słów w przypadku, gdy zbiór uczący jest niewielki. Dla każdego wyrazu losowano p wykonań, na podstawie których wyznaczano cheremy, a pozostałe *40-p* wykonań używano do testów rozpoznawania. Eksperyment powtórzono 20 razy. Uśrednione wyniki dla różnej liczby klastrów przedstawia rysunek 2. Jak łatwo zauważyć, nawet dla

niewielkiej liczby elementów uczących skuteczność rozpoznawania jest na zadowalającym poziomie. Okazało się, że dla liczby klastrów większej niż 5 odnotowano lepsze wyniki przy wykorzystaniu cheremów reprezentowanych przez modele Markowa klastrów (HMM) niż metodą najbliższego sąsiada (NN) z odległością edycji. Do rozpoznawania przy użyciu modeli całych słów korzystano z klasyfikacji metodą najbliższego sąsiada i odległości DTW. Wyniki przez nie uzyskiwane są nawet 10% gorsze niż dla podejść wykorzystujących cheremy.



Rysunek 2: Skuteczność rozpoznawania (podana w %) w zależności od liczby przykładów uczących dla liczby klastrów grupujących cheremy równej 10

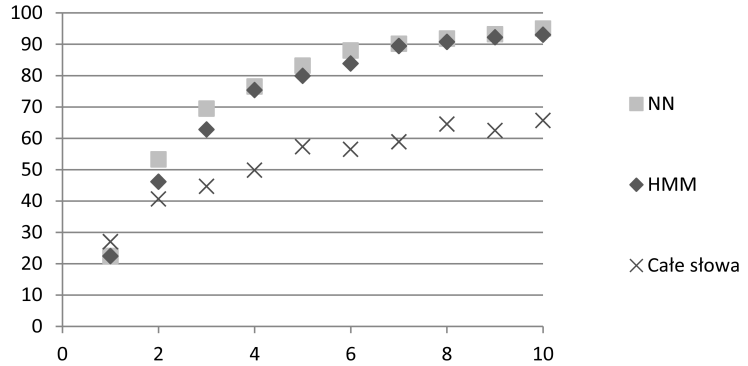
4.3 Rozpoznawanie nowych wyrazów

Wykorzystywanie obszernych zbiorów uczących jest kosztowne i czasochłonne. Dlatego warto przebadać wymagania dotyczące uczenia w odniesieniu do metody opartej na modelowaniu wypowiedzi z wykorzystaniem cheremów. W tym punkcie przebadano sytuację, gdy pewne nowe słowa są rozpoznawane na podstawie nielicznych zbiorów przykładów. Wykorzystano te same dane co poprzednio. Ze zbioru wybrano losowo dziesięć słów (teraz zwanych *nowymi*), które wyłączono z procesu wyznaczania cheremów. Tak więc do wyznaczenia cheremów użyto dane dla 91 słów. Niewielką liczbę w przykładów każdego z nowych słów wykorzystano do zbudowania cheremowych modeli tych słów. Pozostałe $40\text{-}w$ przykładów posłużyło do testowania. Eksperyment powtórzono 20 razy, za każdym razem wybierając inną grupę dziesięciu nowych słów. Rysunek 3 przedstawia średnie wartości skuteczności rozpoznawania w zależności od liczby przykładów uczących dla różnej liczby klastrów. Stosunkowo niewielka liczba przykładów zapewnia dobrą skuteczność rozpoznawania. Dla porównania powtórzono test z modelami całych słów i klasyfikatorem najbliższego sąsiada wykorzystującym odległości DTW. Najlepsze wyniki okazały się gorsze o ok. 30% (por. rysunek 3). Można to wyjaśnić następująco: modele całych słów były reprezentowane przez małą liczbę przykładów, podczas gdy modele wykorzystujące cheremy wykorzystywały dodatkowo informację skumulowaną w cheremach, które zostały wyznaczone na podstawie dużego zbioru danych.

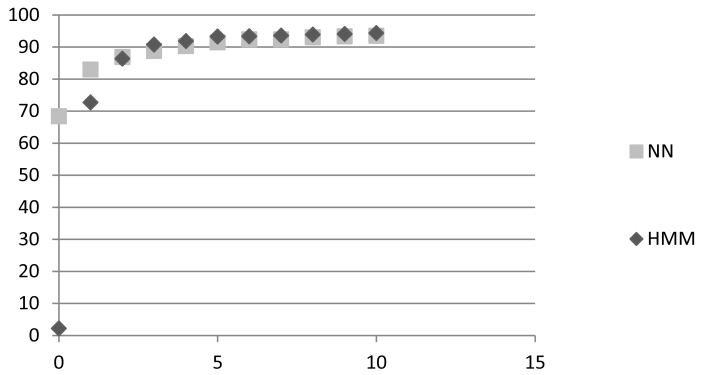
4.4 Rozpoznawanie zdań

Dla każdej z 40 realizacji 35 zdań wykonano transkrypcję do postaci łańcuchów z wykorzystaniem etykiet cheremów otrzymanych dla izolowanych słów. Sposób transkrypcji był taki jak opisany w pkt. 3.2, z tym, że teraz sekwencje reprezentowały zdania a nie słowa. Otrzymane modele będą nazywane modelami typu *I*. Inny model każdego zdania (model typu *II*) utworzono bezpośrednio na podstawie konkatenacji słów zapisanych z użyciem etykiet cheremów. Klasyfikacji

dokonywano metodą najbliższego sąsiada wykorzystującego odległość edycji. Badano skuteczność rozpoznawania w zależności od liczby przykładów. Dla każdego zdania wzięto jego model *II* i pewną niewielką liczbę α wylosowanych modeli *I*, jako zbiór uczący. $1400 - \alpha$ modeli *I* stanowiło zbiór testowy. Eksperyment powtarzano 20 razy. Wynik przedstawia rys. 4. Stosunkowo niewielka liczba przykładów zapewnia dobrą skuteczność rozpoznawania dla obu podejść modelowania cheremów. Gdy liczba klastrów jest niewielka, znacznie lepiej radzi sobie metoda NN z odległością edycji. Niezły rezultat 68.4% dla $\alpha = 0$ odpowiada przypadkowi, gdy występują tylko idealizowane przykłady uczące, które nie uwzględniają zjawiska koartykulacji [37], a tym samym - typowej sytuacji, gdy realizacje słów izolowanych różnią się od ich odpowiedników w zdaniach. Wykonano również dziesięciokrotną walidację krzyżową opartą na modelach typu *I*. Zbiór został podzielony na 10 rozłącznych podzbiorów z czterema modelami każdego zdania. Średnia skuteczność rozpoznawania wyniosła 98.3%.



Rysunek 3: Skuteczność rozpoznawania nowych słów (podana w %) w zależności od liczby przykładów w dla liczby klastrów grupujących cheremy (wartości średnie z 20 eksperymentów) równej 10



Rysunek 4: Skuteczność rozpoznawania zdań (podana w %) w zależności od liczby przykładów α dla liczby klastrów grupujących cheremy (wartości średnie z 20 eksperymentów) równej 10

4.5 Podsumowanie

Systemy rozpoznawania języka migowego wykorzystujące duży zasób słów wymagają modelowania gestów za pomocą jednostek mniejszych niż słowa. Ponieważ badania lingwistyczne w tym zakresie nie są zaawansowane, do konstrukcji wspomnianych jednostek i ich wykorzystaniu

do budowy modeli słów użyto analizy danych w formie szeregów czasowych pozyskanych z materiału filmowego. Problem polega na podziale szeregów na fragmenty, które zgrupowane w klastrach o dobrej jakości pełniły rolę cheremów. Granice cheremów traktowano jako zmienne decyzyjne w problemie optymalizacji rozwiązywanym za pomocą: algorytmu selekcji klonalnej CLONALG oraz algorytmu genetycznego. Jako funkcję celu użyto wskaźnik jakości grupowania. Rozpoznawanie wypowiedzi z wykorzystaniem otrzymanych w zaproponowany sposób cheremów dokonywano na podstawie transkrypcji przebiegów czasowych do postaci łańcuchów etykiet odpowiednich klastrów oraz metody najbliższego sąsiada wykorzystującej odległość edycji. Inny wariant polegał na wykorzystaniu ukrytych modeli Markowa modelujących klastry grupujące cheremy. Modele wypowiedzi budowano jako sieci modeli Markowa skonstruowane na podstawie wspomnianych wcześniej łańcuchów i dostrojone na podstawie przebiegów uczących. Do konstrukcji modeli Markowa i rozpoznawania wykorzystano pakiet HTK. Metodę zweryfikowano z powodzeniem na bazie 101 słów SJM i 35 zdań wykonując liczne eksperymenty dotyczące rozpoznawania wyrażeń SJM, różnicując metody optymalizacji, metody oceny jakości klastrów, metody wyznaczania podobieństwa między cheremami oraz wykorzystując różne wektory cech. Wykonane badania potwierdzają zasadność stosowania jednostek mniejszych niż słowa do modelowania wyrażeń SJM. Na podstawie niewielkiej liczby przykładów udało się uzyskać znacząco lepsze rezultaty niż dla klasyfikatora wykorzystującego wyłącznie modele całych słów. Analiza wyników rozpoznawania ukazała, że wykonywanie gestów w sposób naturalny może być przyczyną większości zaobserwowanych błędów rozpoznawania. Problemy związane ze wzajemnym podobieństwem wykonań różnych wyrazów zaobserwowano również podczas ich grupowania.

5 Podsumowanie

Celem pracy było opracowanie metody rozpoznawania słów i pojedynczych zdań polskiego języka miganego na podstawie analizy sekwencji wizyjnych. Większość wypowiedzi w języku migowym to gesty dynamiczne, które w wizyjnych systemach rozpoznawania są reprezentowane przez szeregi czasowe, tzn. przebiegi zmienności cech wyznaczonych na podstawie analizy obrazów. Przedmiotem badań była metoda rozpoznawania wypowiedzi wykorzystująca modelowanie gestów za pomocą jednostek mniejszych niż słowa. Przypomina to modelowanie za pomocą fonemów w przypadku języka mówionego. Ponieważ nie wiadomo dokładnie, co w wypowiedzi w języku migowym stanowi odpowiedniki fonemów (nazywanych cheremami), zaproponowana w pracy metoda wyodrębnienia cheremów opiera się na analizie danych. Polega ona na określeniu sposobu segmentacji szeregów czasowych reprezentujących wypowiedzi, by powstałe fragmenty - traktowane jako poszukiwane cheremy - tworzyły jednorodne grupy. Punkty podziału szeregów czasowych są wyznaczane jako rozwiązanie zadania optymalizacji, znajdowane z wykorzystaniem algorytmów opartych na analogiach biologicznych (algorytm immunologiczny i genetyczny). Według aktualnej wiedzy autora, tak sformułowane zadanie nie było dotąd rozpatrywane w odniesieniu do rozpoznawania wypowiedzi języka migowego.

W pracy położono nacisk na ocenę wpływu metod określania podobieństwa między cheremami, metody grupowania i wskaźników oceny klastrów, technik optymalizacji oraz typu klasyfikatora na skuteczność rozpoznawania. Integralną częścią pracy jest zaprojektowany przez autora prototyp środowiska wspomagającego eksperymenty i gromadzenie oraz udostępniane danych.

Za najważniejsze osiągnięcia niniejszej pracy autor uważa:

- Metodę wyznaczania cheremów opartą na grupowaniu szeregów czasowych,
- Metodę modelowania słów i prostych zdań z wykorzystaniem cheremów,

- Metodę rozpoznawania słów i prostych zdań z wykorzystaniem opracowanych modeli,
- Środowisko wspomagające eksperymenty i gromadzenie oraz udostępnianie danych,
- Wyniki eksperymentów weryfikujących opracowane metody.

Systemy rozpoznawania języka migowego wykorzystujące duży zasób słów wymagają modelowania gestów za pomocą jednostek mniejszych niż słowa. W zastosowanym podejściu cherem może być reprezentowany przez szereg czasowy znajdujący się w klastrze lub ukryty model Markowa klastra. Metodę zweryfikowano z powodzeniem na bazie 101 słów SJM i 35 zdań wykonując liczne eksperymenty dotyczące rozpoznawania wyrażeń SJM, różnicując metody optymalizacji, metody oceny jakości klastrów, metody wyznaczania podobieństwa między cheremami oraz wykorzystując różne wektory cech. Badania potwierdzają zasadność stosowania jednostek mniejszych niż słowa do modelowania wyrażeń SJM. Na podstawie niewielkiej liczby przykładów udało się uzyskać znacząco lepsze rezultaty niż dla klasyfikatora wykorzystującego wyłącznie modele całych słów.

Wyniki przedstawionych badań pozwalają stwierdzić, że podjęty cel badawczy został zrealizowany, a teza potwierdzona.

Rozwiązania projektowe i programistyczne związane z budową bazy danych wizyjnych mogą wspomóc inne zespoły badawcze pracujące na polu automatycznej analizy zachowań osób.

Rozszerzenie opracowanej metody mogłoby polegać na zastosowaniu przyszłej generacji kamer aktywnych do akwizycji obrazów (kamery Time of Flight lub Kinect), które ułatwią śledzenie palców osoby wykonującej gesty. Pozwoliłoby to na eliminację części ograniczeń systemu związanych z akwizycją obrazów z kamery [51]. Innym rozszerzeniem mogłoby być rozpoznawanie ciągłych wypowiedzi w języku migowym [7] wymagające rozwiązania problemu odnalezienia miejsca, gdzie kończy się jeden gest a rozpoczyna następny (tzw. *gesture spotting*) uwzględniając zjawisko koartykulacji [37] zniekształcające oba gesty. Kolejny kierunek dalszych badań dotyczy uwzględniania ruchu ust.

Literatura

[1] Assaleh K., Shanableh T., Fanaswala M., Amin F., Bajaj H.: Continuous Arabic Sign Language Recognition in User Dependent Mode. *JILSA*, 2(1):19–27, 2010.

[2] Awad G.: *A Framework for Sign Language Recognition using Support Vector Machines and Active Learning for Skin Segmentation and Boosted Temporal Sub–units*. Praca doktorska, Dublin City University, 2007.

[3] Awad G., Han J., Sutherland A.: Novel boosting framework for subunit-based sign language recognition. *Proceedings of the 16th IEEE international conference on Image processing*, ICIP'09, strony 2693–2696. IEEE Press, Piscataway, NJ, USA, 2009.

[4] Bauer B., Kraiss K.-F.: Video-Based Sign Recognition Using Self-Organizing Subunits. *ICPR (2)*, strony 434–437, 2002.

[5] Bilal S., Akmeliawati R., Shafie A., Salami M.: Hidden Markov model for human to computer interaction: a study on human hand gesture recognition. *Artificial Intelligence Review*, strony 1–22, 2011.

[6] Cooper H.: *Sign Language Recognition: Generalising to More Complex Corpora*. Praca doktorska, Centre For Vision Speech and Signal Processing, University Of Surrey, 2010.

[7] Cooper H., Holt B., Bowden R.: Sign Language Recognition. Moeslund T. B., Hilton A., Krüger V., Sigal L., redaktorzy, *Visual Analysis of Humans*, strony 539–562. Springer, 2011.

[8] Cyganek B.: *Komputerowe przetwarzanie obrazów trójwymiarowych*. Problemy Współczesnej Nauki: Informatyka. Akademicka Oficyna Wydawnicza EXIT, 2002.

[9] Castro L. N. de, Von Zuben F. J.: Learning and optimization using the clonal selection principle. *Trans. Evol. Comp*, 6(3):239–251, 2002.

[10] Flasiński M., Myśliński S.: On the use of graph parsing for recognition of isolated hand postures of Polish Sign Language. *Pattern Recognition*, 43(6):2249–2264, 2010.

[11] Flasiński M.: *Wstęp do Sztucznej Inteligencji*. Państwowe Wydawnictwo Naukowe PWN, 2011.

[12] Han J., Awad G., Sutherland A.: Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623–633, 2009.

[13] Huenerfauth M., Lu P.: Effect of spatial reference and verb inflection on the usability of sign language animations. *Universal Access in the Information Society*, 11(2):169–184, 2012.

[14] Kapuściński T.: Vision-Based Recognition of Fingerspelled Acronyms Using Hierarchical Temporal Memory. Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L., Zurada J., redaktorzy, *Artificial Intelligence and Soft Computing*, wolumen 7267 serii *Lecture Notes in Computer Science*, strony 527–534. Springer Berlin / Heidelberg, 2012.

[15] Kapuściński T., Wysocki M.: Using Hierarchical Temporal Memory for Recognition of Signed Polish Words. Kurzyński M., Woźniak M., redaktorzy, *Computer Recognition Systems 3*, wolumen 57 serii *Advances in Intelligent and Soft Computing*, strony 355–362. Springer Berlin / Heidelberg, 2009.

[16] Kosmidou V., Petrantonakis P., Hadjileontiadis L. J.: Enhanced Sign Language Recognition Using Weighted Intrinsic-Mode Entropy and Signer's Level of Deafness. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 41(6):1531–1543, 2011.

[17] Kraiss K.-F.: *Advanced man machine interaction*. Springer, Berlin, 2006.

[18] Li H.: *Model-based segmentation and recognition of continuous gestures*. Praca doktorska, Queen's University, Kingston, Ontario, Canada, 2010.

[19] Li H., Zhang K., Jiang T.: Minimum Entropy Clustering and Applications to Gene Expression Analysis. *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, CSB '04, strony 142–151. IEEE Computer Society, Washington, DC, USA, 2004.

[20] Lichtenauer J., Hendriks E. A., Reinders M. J. T.: Learning to recognize a sign from a single example. *8th IEEE International Conference on Automatic Face and Gesture Recognition, 2008. FG'08.*, strony 1–6. IEEE, 2008.

[21] Lichtenauer J., Holt G. A. ten, Reinders M. J. T., Hendriks E. A.: Person-Independent 3D Sign Language Recognition. Dias M. S., Gibet S., Wanderley M. M., Bastos R., redaktorzy, *Gesture Workshop*, wolumen 5085 serii *Lecture Notes in Computer Science*, strony 69–80. Springer, 2007.

[22] Lubaszewski W.: *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu*. AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, 2009.

[23] Marnik J.: The Polish Finger Alphabet Hand Postures Recognition Using Elastic Graph Matching. Kurzynski M., Puchala E., Wozniak M., Zolnierek A., redaktorzy, *Computer Recognition Systems 2*, wolumen 45 serii *Advances in Soft Computing*, strony 454–461. Springer Berlin / Heidelberg, 2007.

[24] Marnik J., Oszust M.: Rozpoznawanie statycznych gestów dłoni na potrzeby interfejsów człowiek-komputer. Trybus L., Samolej S., redaktorzy, *Projektowanie, analiza i implementacja systemów czasu rzeczywistego*, strony 573–582. WKŁ, 2012.

[25] Maulik U., Bandyopadhyay S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(12):1650–1654, 2002.

[26] MEC: Minimum Entropy Clustering Java package, http://www.cs.ucr.edu/ hli/mec/, Dostęp z 10.08.2012.

[27] Ong S. C. W., Ranganath S.: Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):873–891, 2005.

[28] Oszust M., Wysocki M.: Determining Subunits for Sign Language Recognition by Evolutionary Cluster-Based Segmentation of Time Series. Rutkowski L., Scherer R., Tadeusiewicz R., Zadeh L. A., Zurada J. M., redaktorzy, *ICAISC (2)*, wolumen 6114 serii *Lecture Notes in Computer Science*, strony 189–196. Springer, 2010.

[29] Oszust M., Wysocki M.: Recognition of Signed Expressions Using Cluster-Based Segmentation of Time Series. Choras R., redaktor, *Image Processing and Communications Challenges 2*, wolumen 84 serii *Advances in Intelligent and Soft Computing*, strony 167–174. Springer Berlin / Heidelberg, 2010.

[30] Oszust M., Wysocki M.: Recognition of signed expressions using visually-oriented subunits obtained by an immune-based optimization. Martin T. P., Muda A. K., Abraham A., Prade H., Laurent A., Laurent D., Sans V., redaktorzy, *SoCPaR*, strony 41–46. IEEE, 2010.

[31] Oszust M., Wysocki M.: Modelling and Recognition of Signed Expressions Using Subunits Obtained by Data–Driven Approach. Ramsay A., Agre G., redaktorzy, *AIMSA*, wolumen 7557 serii *Lecture Notes in Computer Science*, strony 315–324. Springer, 2012.

[32] Pasaholoudi V. N., Margaritis K. G.: Hidden Marcov Models for Sign Language Recognition: a Review. *2nd Hellenic Conf. on AI, SETN-2002*, strony 343–354. Companion Volume, Thessaloniki, Greece, 2002.

[33] Pitsikalis V., Theodorakis S., Maragos P.: Data-Driven Sub-Units and Modeling Structure for Continuous Sign Language Recognition with Multiple Cues. *LREC Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (LREC-2010), Valletta, Malta*, 2010.

[34] Pitsikalis V., Theodorakis S., Vogler C., Maragos P.: Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. *Gesture11*, strony 1–6, 2011.

[35] Pękalska E., Duin R. P. W.: Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002.

[36] Riad A. M., Elmonier H. K., Shohieb S. M., Asem A. S.: SignsWorld; Deeping Into the Silence World and Hearing Its Signs (State of the Art). *CoRR*, 2012.

[37] Segouat J., Braffort A.: Toward modeling sign language coarticulation. *Proceedings of the 8th international conference on Gesture in Embodied Communication and Human-Computer Interaction*, GW'09, strony 325–336. Springer-Verlag, Berlin / Heidelberg, 2010.

[38] Stokoe W. C.: Sign language structure: an outline of the visual communication systems of the American deaf. *Journal of deaf studies and deaf education — Studies in Linguistics, Occasional Papers*, 1960, 2005.

[39] Suszczańska N., Szmal P., Francik J.: Translating Polish Texts Into Sign Language In The Tgt System. strony 282–287, Innsbruck, Austria, 2002.

[40] Szczepankowski B.: *Język migany w szkole*. WSiP, 1988.

[41] Tadeusiewicz R.: *Systemy wizyjne robotów przemysłowych*. Wydawnictwa Naukowo-Techniczne, 1992.

[42] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997.

[43] Tamura S., Kawasaki S.: Recognition of sign language motion images. *Pattern Recognition*, 21(4):343–353, 1988.

[44] Tang M.: Recognizing Hand Gestures with Microsoft's Kinect. Department of Electrical Engineering, Stanford University, 2011.

[45] Theodorakis S., Pitsikalis V., Maragos P.: Model-level data-driven sub-units for signs in videos of continuous sign language. *ICASSP*, strony 2262–2265. IEEE, 2010.

[46] Theodoridis S., Koutroumbas K.: *Pattern Recognition, Fourth Edition*. Academic Press, 2008.

[47] Tomaszewski P.: *Fonologia wizualna Polskiego Języka Migowego*. Matrix, 2010.

[48] Turunen J. J., Lipping T.: Phoneme analysis based on quantitative and qualitative entropy measurement. *Computer Speech and Language*, 22(4):313–329, 2008.

[49] Vogler C., Metaxas D. N.: Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes. Braffort A., Gherbi R., Gibet S., Richardson J., Teil D., redaktorzy, *Gesture Workshop*, wolumen 1739 serii *Lecture Notes in Computer Science*, strony 211–224. Springer, 1999.

[50] Wang Q., Chen X., Zhang L.-G., Wang C., Gao W.: Viewpoint invariant sign language recognition. *Comput. Vis. Image Underst.*, 108(1-2):87–97, 2007.

[51] Wysocki M., Kapuściński T., Marnik J., Oszust M.: *Rozpoznawanie gestów wykonywanych rękami w systemie wizyjnym*. Oficyna Wydawnicza Politechniki Rzeszowskiej, 2011.

[52] Xu R., Wunsch D.: *Clustering*. Wiley-IEEE Press, 2009.

[53] Young S. J., Kershaw D., Odell J., Ollason D., Valtchev V., Woodland P.: *The HTK Book Version 3.4*. Cambridge University Press, 2009.

[54] Zahedi M., Manashty A. R.: Robust Sign Language Recognition System Using ToF Depth Cameras. *CoRR*, 2011.

[55] Ziółko B., Ziółko M.: *Przetwarzanie mowy*. Wydawnictwa AGH, 2011.