Włodzimierz Kasprzak

Rozpoznawanie Obrazów i Sygnałów Mowy

Konspekt

Wstęp

Celem niniejszej książki jest przybliżenie zagadnienia naukowego, zwanego „rozpoznawaniem wzorców", i jego głównych zastosowań w technice, jakimi są systemy analizy obrazów cyfrowych i sygnałów mowy, stosunkowo szerokiej grupie studentów i inżynierów różnych specjalności, w tym: Informatyki, Automatyki i Robotyki, Optoelektroniki i Inżynierii Biomedycznej. Książka stanowić może materiał pomocniczy do wykładu i ćwiczeń. W części wykładowej kładzie się nacisk na opisy funkcjonalne i algorytmiczne metod analizy. W ramach ćwiczeń następuje praktyczne opanowanie tych metod na drodze ich symulacji z wykorzystaniem przykładowych danych. Oba etapy przygotowują czytelnika do samodzielnego wykonania projektu informatycznego, którego celem jest implementacja programowa wybranego systemu analizy obrazu lub mowy.

Ze względu na duży zakres materiału konieczna była jego selekcja, którą autor wykonał w oparciu o trzy kryteria. Po pierwsze, w naturalny sposób w ramach zagadnienia analizy sygnałów i obrazów możemy wyróżnić trzy poziomy abstrakcji danych: poziom przetwarzania sygnału, poziom segmentacji sygnału i rozpoznawania obiektów oraz poziom rozumienia sygnału. Autor skoncentrował się na pośrednim poziomie analizy (segmentacja sygnału i rozpoznawanie obiektów), gdyż zagadnienia przetwarzania sygnału są przedmiotem wielu podręczników, a poziom rozumienia sygnałów posiada silny związek z dziedziną „sztucznej inteligencji".

Po drugie, ograniczono omawianie algorytmów rozpoznawania obiektów i sekwencji słów w zasadzie do statycznej analizy względem czasu, tzn. do analizy pojedynczych obrazów i do „wsadowej" (nieiteracyjnej, niekontekstowej) analizy wycinków sygnału mowy. Jedynym odstępstwem jest tu rozdział 6 dotyczący sposobów detekcji estymacji ruchu w sekwencji obrazów. Tym samym pominięto analizę danych obrazowych uzyskiwanych metodami stereo-wizji, dalmierzami laserowymi, 3-wymiarowej tomografii i za pomocą innych specjalizowanych urządzeń pomiarowych. W opinii autora ten obszar analizy obrazów odpowiada specjalizowanym wykładom nt. zastosowań wizji komputerowej w robotyce, w medycynie, w nawigacji, itd.

Po trzecie, autor zdaje sobie sprawę z faktu, że nawet w zakresie rozpoznawania obiektów pominął pewne klasy algorytmów, jak np. morfologiczne metody przetwarzania obrazu, modelowanie analizy metodami zbiorów rozmytych lub algorytmami genetycznymi. W tym względzie, z uwagi na wymóg spójności treści, zdecydowało subiektywne odczucie autora o ogólności i skuteczności prezentowanych algorytmów.

Materiał podzielony został na trzy części zatytułowane: rozpoznawanie wzorców, rozpoznawanie obrazów i rozpoznawanie sygnałów mowy.

W pierwszej części przedstawiono pojęcie wzorca, poziomy abstrakcji wzorców, procesy klasyfikacji prostych wzorców i rozpoznawania złożonych wzorców (rozdział 1). Omówiono zagadnienia transformacji przestrzeni cech metodami analizy składowych głównych i linowej analizy dyskryminacyjnej oraz główne rodzaje klasyfikatorów (rozdział 2).

W drugiej części, w zakresie rozpoznawania obrazów omawiane są: zagadnienia reprezentacji obrazów, kalibracji kamery i normowania kształtów w obrazie (rozdział 3), algorytmy segmentacji obrazu, wyznaczania cech tekstur i konturów w obrazie (rozdział 4), problemy rozpoznawania 2-wymiarowych i 3-wymiarowych obiektów (rozdział 5) oraz algorytmy analizy sekwencji obrazów w celu detekcji ruchu, śledzenia obiektów i autonomicznej nawigacji (rozdział 6).

W trzeciej części, w zakresie rozpoznawania mowy prezentowane są zagadnienia: reprezentacji cyfrowego sygnału mowy w dziedzinie czasu i częstotliwości (rozdział 7), przetwarzania i detekcji sygnału mowy w sygnale akustycznym (rozdział 8), segmentacji i wyznaczania cech sygnału mowy w dziedzinie czasu i częstotliwości (rozdział 9), modelowania akustyczno-fonetycznego sygnału mowy (rozdział 10), tworzenia statystycznego modelu słów i rozpoznawania sekwencji słów (rozdział 11).

Punkty z treścią o zaawansowanym charakterze oznaczono w treści za pomocą znaku (*). Literatura podzielona została na podstawową, zamieszczoną na końcu książki, i uzupełniającą, podawaną w częściach po kolejnych rozdziałach.-

Bibliografia podstawowa

[1] J. Benesty, M.M. Sondhi, Y. Huang. *Springer Handbook of Speech Processing*. Springer, Berlin Heidelberg, 2007.

[2] R. Duda, P. Hart, D. Stork. *Pattern Classification*. 2nd edition, John Wiley & Sons, New York, 2001.

[3] H. Niemann. *Klassifikation von Mustern*. 2nd edition, Springer, Berlin, 2003.

[4] Pakiet oprogramowania: *CSLU Speech Toolkit*. Oregon Graduate Institute, `cslu.cse.ogi.edu`, 2001-2005.

[5] D. Paulus, J. Hornegger. *Applied Pattern Recognition. A Practical Introduction to Image and Speech Processing in C++*. Vieweg, Braunschweig, 3d edition, 2001.

[6] I. Pitas. *Digital Image Processing Algorithms*, Prentice Hall, New York, 1993.

[7] L. Rabiner, B.-H. Juang. *Fundamentals of speech recognition*. Prentice Hall, New York, 1993.

[8] E.-G. Schukat-Talamazzini. *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Wiesbaden, 1995.

[9] W. Skarbek. *Metody reprezentacji obrazów cyfrowych*, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993.

[10] R. Tadeusiewicz, P. Korohoda: *Komputerowa analiza i przetwarzanie obrazów*. FPT, Kraków 1997.

I. Rozpoznawanie wzorców

Rozdział 1. Podstawy rozpoznawania wzorców

Rozpoznawanie wzorców (ang. *pattern recognition*) jest zagadnieniem naukowym, które czerpie swoje podstawy z wielu teorii podstawowych, takich jak teoria informacji, analiza matematyczna i statystyka [2, 3, 9, 10]. Ze względu na stosowane narzędzia formalne i implementacyjne ta dziedzina wiedzy lokuje się najczęściej na styku automatyki i informatyki [1.11, 1.12, 1.13]. Rozwiązania teoretyczne przekładane są na praktyczne metody z wykorzystaniem narzędzi opisu i algorytmów stosowanych w innych działach nauk inżynierskich, takich jak sztuczna inteligencja (ang. *artificial intelligence*), lingwistyka obliczeniowa (ang. *computational linguistics*), optymalizacja, sieci neuronowe, algorytmy genetyczne i ewolucyjne, itp. [1.7, 1.9, 2.3, 2.5, 2.8].

Zadaniem systemu rozpoznawania wzorców jest stworzenie **symbolicznego opisu** (w postaci funkcji, segmentów, obiektów, ruchu, wyrazów lub zdań języka, struktur, itd.) dla zawartości rzeczywistego 1-wymiarowego, 2-wymiarowego lub wyżej wymiarowego sygnału cyfrowego (obrazu cyfrowego, sygnału mowy, obrazu z kamery, serii skanów tomograficznych, itd.) i przyporządkowanie opisowi jego **klasy** lub instancji klasy (np. znaku, grupy osób, typu pojazdu, rodzaju choroby, itp.), czyli jego znaczenia (semantyki) w ramach dziedziny zastosowania [2, 3, 1.2, 1.3].
Głównymi obszarami **zastosowania** teorii rozpoznawania wzorców są systemy analizy obrazów cyfrowych i sygnałów mowy [1, 4, 5, 6, 7, 8].

1.1 Pojęcie wzorca

Zakładamy, że realizujemy opis **środowiska U** przy pomocy rodziny funkcji $b_r(\boldsymbol{x})$. Zadania rozwiązywane w **problemie rozpoznawania wzorców** zwykle wymagają ograniczenia uniwersalnego środowiska do **dziedziny** rozpatrywanego problemu **Ω.** Elementy dziedziny **Ω** nazywamy **wzorcami**. Są to więc zbiory konkretnych funkcji wielo-argumentowych wyrażające pewną wielkość fizyczną, obiekt, system, itd.
W ramach dziedziny **Ω** możemy wyróżnić **klasy wzorców** (Ω_1, Ω_2, ... , Ω_m), czyli grupy wzorców o podobnych cechach i pełniących podobną rolę w ramach dziedziny zastosowania.
Pojęcie **rozpoznawania wzorca** ω odnosimy tu do procesu przyporządkowania wzorcowi jego klasy (utożsamiamy to z **klasyfikacją wzorca)**.
W systemach analizy sygnałów i obrazów, stosowanych np. w celach biometrycznych, pojęcie rozpoznawania obejmuje zbliżone pojęcia: **identyfikacja** i **weryfikacja**. W takich rozwiązaniach zwykle system analizy nie dysponuje opisami klas wzorców a jedynie przechowuje szereg wzorców referencyjnych. Celem systemu biometrycznego jest porównanie aktualnego wzorca z bazą referencyjną, określenie odległości pomiędzy parami wzorców i jeśli możliwy jest wybór dokładnie jednego wzorca referencyjnego to mówimy o identyfikacji, a jeśli możemy wybrać przynajmniej jeden ze wzorców referencyjnych to mówimy o weryfikacji.

1.2 Paradygmaty rozpoznawania wzorców

Wyróżnimy różne kategorie wzorców (ze względu na "trudność" ich rozpoznania):
- **proste** wzorce,
- **złożone** wzorce (sekwencja lub struktura prostych wzorców),
- **abstrakcyjne** wzorce (np. dynamiczna scena, zdania mówione).

Naiwny sposób rozpoznawania prostych wzorców polega na dopasowywaniu „punkt-po-punkcie” przebiegów sygnału ze wzorcem referencyjnym (modelem), po ewentualnie wymaganej normalizacji. Takie rozwiązanie jest możliwe tylko w bardzo ograniczonych przypadkach.
Typowym sposobem rozpoznawania prostych wzorców jest **klasyfikacja**, która polega na:
- wyznaczeniu wektora cech wzorca,
- uprzednim zaprojektowaniu (nauczeniu) klasyfikatora cech,
- przyporządkowaniu cech wzorca do jednej ze znanych klas (wymaga istnienia miary odległości i reguły decyzyjnej).

Problem **rozpoznawania złożonych** wzorców wymaga stworzenia modeli wzorców z pewnej dziedziny i zdefiniowania algorytmu dopasowującego wzorce z ich modelami. Zwykle taki model wyrażony jest za pomocą sekwencji lub struktury prostych wzorców. Tym samym wyróżnimy dwa główne zagadnienia podczas projektowania klasyfikatora złożonych wzorców:
- Stworzenie **modeli** (opisów symbolicznych) klas wzorców, wyrażonych poprzez ich prostsze części (**dekompozycja** modelu na prostsze części);

- Zaprojektowanie algorytmu **dopasowania** (**pasowania**) opisu ze wzorcem (różne miary odległości i funkcje decyzyjne).

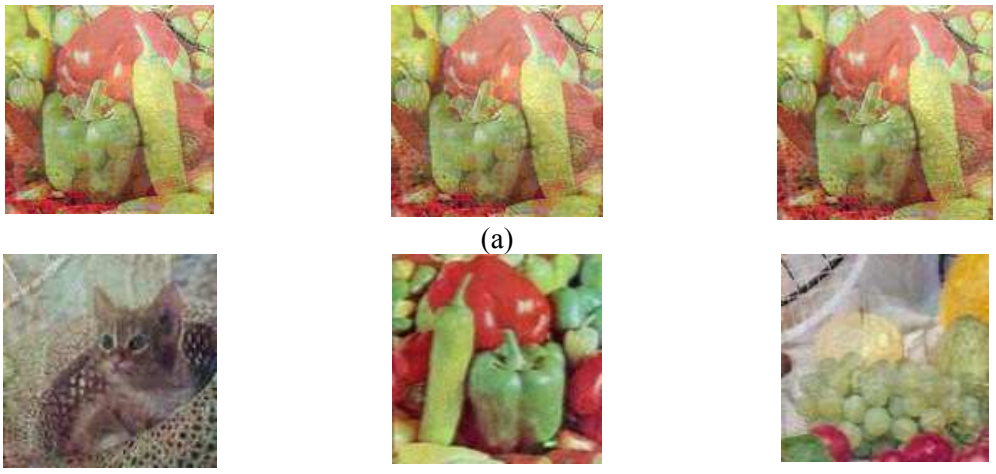
Przykłady rozpoznawania złożonych wzorców:

- rozpoznawanie napisów - sekwencji znaków (prostych wzorców),
- rozpoznawanie słów (poleceń) w sygnale mowy,
- rozpoznawanie 2- i 3-wymiarowych obiektów w obrazach cyfrowych.

Proste i złożone wzorce utożsamiamy zwykle z fizycznymi obiektami, „widocznymi" w obrazie lub sygnale mowy. Inną interpretację nadamy **wzorcom abstrakcyjnym** – są nimi zwykle zdania w języku naturalnym, które wyrażają „wysoko-poziomowy" opis sceny lub przekaz informacji werbalnej od innego człowieka [1.4, 1.6, 1.8, 1.10]. Np. rozumienie zapytań w systemie dialogowym mowy, interpretacja sekwencji obrazów medycznych, interpretacja akcji pojazdu w systemie analizy sceny ruchu drogowego w oparciu o sekwencję obrazów cyfrowych.

1.3 Analiza cyfrowych obrazów

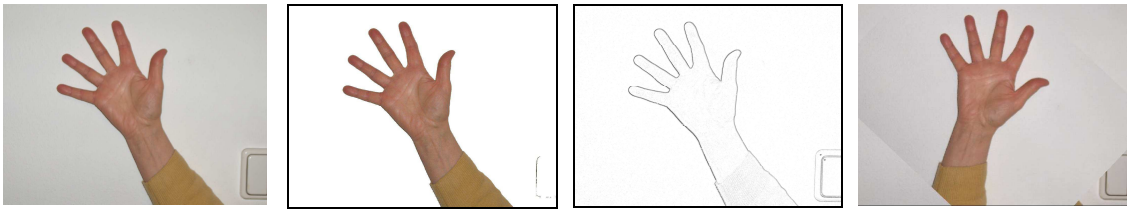
System analizy obrazów może realizować procesy na różnych **poziomach abstrakcji danych (opisu)** [1.5]: **poziom sygnału** : np. usuwanie szumu, separacja obrazów źródłowych z ich mieszanin (rys. 1.11); **poziom ikoniczny**: np. rejestracja obrazu względem obrazu tła, detekcja obrazu krawędziowego, klasyfikacja całego obrazu (rys. 1.12); **poziom segmentacji**: np. detekcja linii, obszarów jednorodnych, tekstur, konturów obiektu (rys. 1.13), ruchomych obszarów (rys. 1.14); p**oziom obiektów**: rozpoznawanie obiektu w oparciu o 2- lub 3-wymiarowy model klasy obiektu (rys. 1.15), rekonstrukcja znanego obiektu (rys. 1.16); p**oziom opisu sceny**: interpretacja scen zawierających ruchome obiekty.



(a)

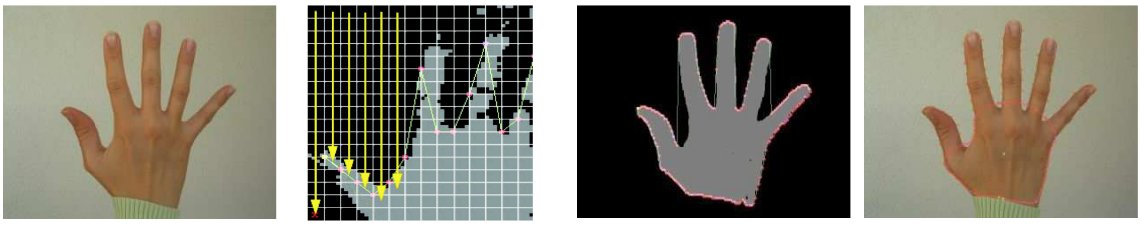
(b)

Rys. 1.11. Przykład operacji na poziomie sygnału - separacja obrazów źródeł z ich mieszanin: (a) trzy mieszaniny na wejściu, (b) trzy odseparowane obrazy źródeł.



(a) (b) (c) (d)

Rys. 1.12. Przykłady operacji na poziomie ikonicznym: (a) obraz wejściowy, (b) rejestracja obrazu względem obrazu tła, (c) wyznaczenie obrazu krawędziowego, (d) normalizacja obrazu – obrót do osi głównej dla obrazu krawędziowego.

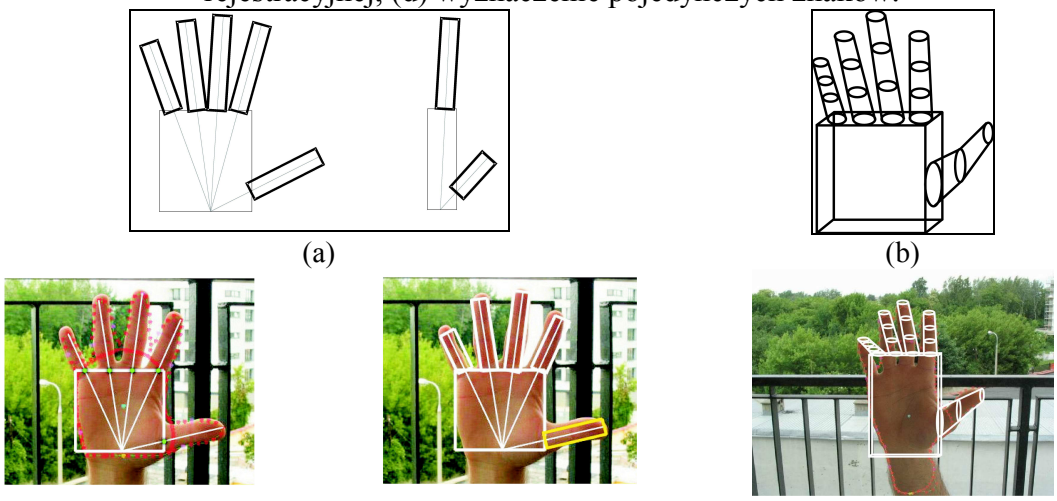


(a) (b) (c) (d)

Rys. 1.13. Przykład segmentacji obrazu - detekcja konturu obiektu: (a) obraz wejściowy, (b) wyznaczanie kolejnych punktów konturu w obrazie krawędziowym, (c) znaleziony kontur początkowy, (d) wyznaczone dwa kontury obejmujące różne obszary obiektu.

(a) (b) (c) (d)

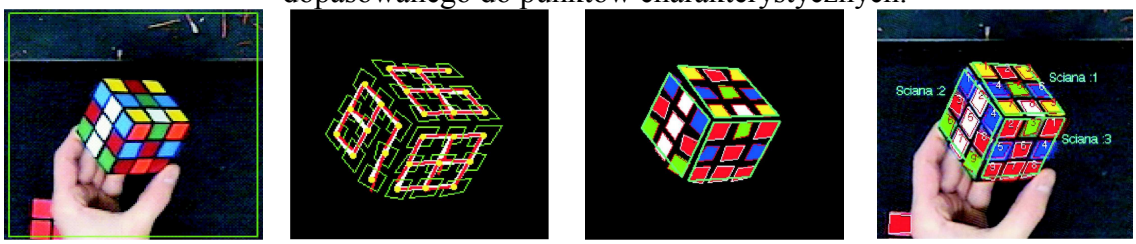
Rys. 1.14. Przykład segmentacji obrazu - wyznaczenie tablicy rejestracyjnej: (a) detekcja ruchomego obiektu, (b) określenie obszaru zainteresowania, (c) wyznaczanie obszaru tablicy rejestracyjnej, (d) wyznaczenie pojedynczych znaków.



(a) (b)

(c) (d) (e)

Rys. 1.15. Rozpoznawanie obiektu w oparciu o 2- lub 3-wymiarowy model klasy obiektu: (a) 2-wymiarowe modele dłoni, (b) 3-wymiariwy model dłoni, (c) wyznaczenie punktów charakterystycznych na podstawie pary konturów dla dłoni, (d) widok 2-wymiarowego modelu dopasowanego do punktów charakterystycznych, (e) widok modelu 3-wymiarowego dopasowanego do punktów charakterystycznych.



(a) (b) (c) (d)

Rys. 1.16. Rekonstrukcja znanego obiektu jako przykład operacji na poziomie obiektów: (a) obraz wejściowy zawierający kostkę Rubika, (b) grupowanie jednorodnych obszarów prostokątnych, (c) aproksymacja powierzchni dla każdej grupy, (c) wyznaczenie ścianek i ich elementów – rekonstrukcja kostki Rubika.

1.4 Analiza sygnałów mowy

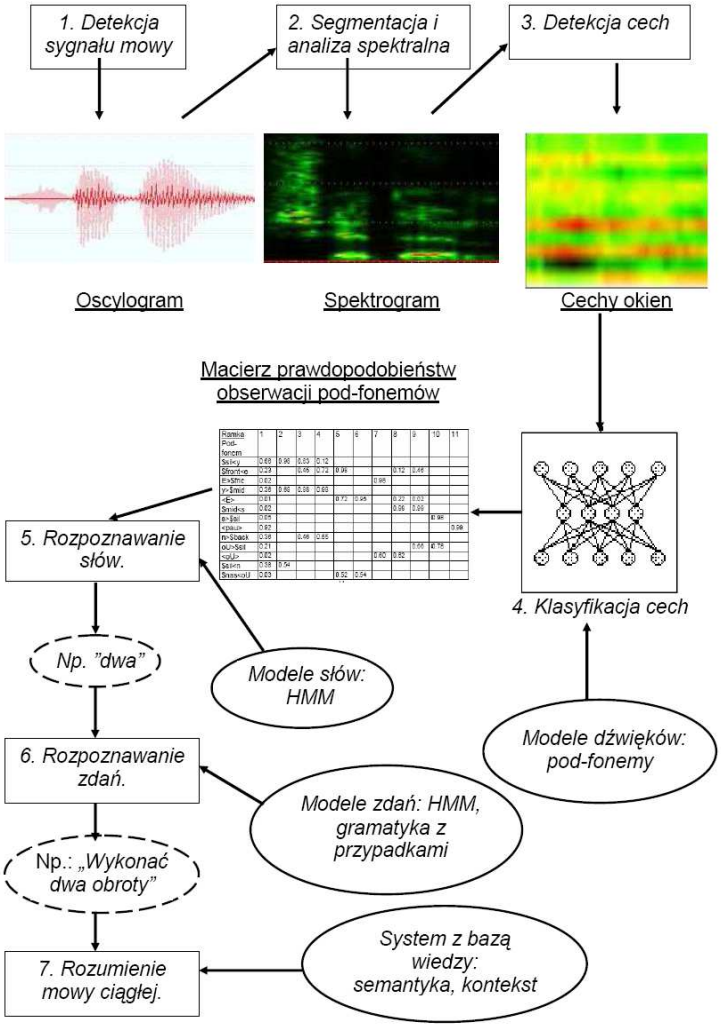
Dialog za pomocą głosu pomiędzy człowiekiem i urządzeniem może odbywać się w różnych sytuacjach [1, 4, 7, 8, 1.3]. Struktura systemu rozpoznawania mowy, podana na rys. 1.17, obejmuje część sterowaną danymi (kroki 1-4) i część sterowaną modelem (kroki 5-7).

Sygnał mowy powstaje jako seria czasowa oddająca wartości ciśnienia środowiska rejestrowane w czujniku. **Oscylogram** to wykres przebiegu zmierzonego sygnału w dziedzinie czasu.

Artykułowane dźwięki, które utożsamiamy z ludzką mową, różnią się od dźwięków nieartykułowanych, będących krzykiem, tym, że posiadają one charakterystyczną strukturę częstotliwościową. Dlatego też będziemy poszukiwać opisu dla podstawowych jednostek fonetycznych, z których składa się mowa, w wyniku uprzedniej analizy sygnału w przestrzeni czas-częstotliwość (**spektrogram**). Taki obraz sygnału uzyskamy dzięki zastosowaniu **okienkowej transformaty Fouriera**.

Podstawowa częstotliwość (F0) – dominująca częstotliwość w sygnale mowy w danej chwili, której harmoniczne częstotliwości tworzą aktualny sygnał. Ta częstotliwość zależy od cech mówcy i intonacji wypowiedzi.

Formanty są to duże kolejne koncentracje energii w spektrogramie dla coraz większych częstotliwości (oznaczane F1, F2, F3 itp.). Położenia tych formantów są charakterystyczne dla rodzaju dźwięku – jednostki fonetycznej.



Rys. 1.17. Typowa struktura systemu rozpoznawania mowy z wykorzystaniem ukrytych modeli Markowa (HMM) na etapie rozpoznawania słów i zdań.

Omawiamy typowy zestaw **26 cech numerycznych** dla każdego okna sygnału. Przedstawiamy przykładową realizację **klasyfikatora cech** okna w postaci sieci neuronowej (wielowarstwowy perceptron – MLP). Wyjście sieci podaje prawdopodobieństwa wystąpienia każdej jednostki fonetycznej w aktualnym oknie. Zbierając wyniki klasyfikatora dla wszystkich okien analizowanego sygnału uzyskamy macierz prawdopodobieństw dla jednostek fonetycznych i okien.
Fonemy to lingwistyczne klasy dźwięków, częściowo zależne od języka. Samogłoski mogą być dzielone na monoftongi i dyftongi (dwugłoski). Pozostałe fonemy to spółgłoski, które dzielimy na: zbliżeniowe, nosowe, szczelinowe (tnące), zwarte (wybuchowe), afrykaty (zwarto-szczelinowe).
Pod-fonemy. Fonem ma duży wpływ na sąsiednie fonemy (np. /E/ po /s/ może wyglądać całkiem inaczej niż /E/ po /b/). Dlatego stosuje się podział każdego fonemu na jedną, dwie lub trzy części zależnie od czasu trwania fonemu i jego wpływu na sąsiadów.
HMM (ukryty model Markowa) to typowy schemat reprezentacji słów i (również) zdań, złożony ze struktury w postaci skierowanego grafu i rozkładów prawdopodobieństw przejść (związanych z łukami) i wyjść (związanych z węzłami - stanami). Taki model umożliwia reprezentację różnych transkrypcji tego samego słowa, różnych długości czasu trwania wypowiedzi tego samego słowa i tolerowanie pojedynczych przekłamań podczas klasyfikacji ramek sygnału.
Przeszukanie Viterbiego to algorytm poszukiwania najlepszej ścieżki przejścia przez macierz prawdopodobieństw trzy-fonów względem modelu każdego słowa (zdania).
Czasem wystarczy nam **prosty system klasyfikacji spektrogramów** o znormalizowanym rozmiarze zamiast generalnego systemu rozpoznawania mowy. Przedstawiamy strukturę takiego systemu klasyfikacji spektrogramów.

1.5 Statystyka sygnału / obrazu cyfrowego

Odpowiadamy na pytanie: „dlaczego statystyka jest potrzebna dla rozpoznawania wzorców?” Przypominamy pojęcia: dyskretna zmienna losowa, momenty rozkładu prawdopodobieństwa, wektor zmiennych losowych, niezależność i gęstość warunkowa,

1.6 Próbkowanie i digitalizacja sygnału analogowego

W procesie akwizycji sygnału cyfrowego, z punktu widzenia cyfrowego systemu obliczeniowego, powstają dwa zasadnicze problemy: 1) **próbkowanie** sygnału w czasie - jak zapewnić **bezstratną** postać cyfrową sygnału w procesie próbkowania sygnału w ciągłym czasie lub przestrzeni i 2) jak zapewnić **optymalną** digitalizację amplitudy sygnału analogowego.
Omawiamy **twierdzenie o próbkowaniu i z**asadę **kwantyzacji** amplitudy czyli cyfrowego kodowania amplitudy sygnału. Wprowadzamy **stosunek sygnału do szumu** jako miarą dokładności kwantyzacji. Omawiamy **kodowanie PCM** (ang. *pulse code modulation*) - modulację kodowo-impulsową – jako typowy sposób optymalnego wyznaczania przedziałów kwantyzacji dla amplitudy sygnału.

1.7 Wybrane probemy optymalizacji

Omawiamy metodę najmniejszych kwadratów, algorytm Gaussa dla rozwiązania układu równań i metode optymalizacji kwadratowej wypukłej przy zadanych ograniczeniach.

1.8 Zadania

Zadania dotyczą: 1) projektowania klasyfikatora spektrogramów, 2) obliczania momentów rozkładu zmiennej losowej na podstawie z histogramu obrazu, 3) definiowania funkcji w języku C++ dla obliczania wartości prawdopodobieństwa dla 1-wymiarowego rozkładu Gaussa, 4) rozwiązywanie układu równań, 5) badania miary SNR w przypadku dyskretnej

reprezentacji amplitudy za pomocą B bitów.

Literatura uzupełniająca do rozdziału 1

[1.1] K. Fukunaga: *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.
[1.2] R.C. Gonzalez, P.Wintz: *Digital Image Processing*. Addison-Wesley, Reading MA, 1987.
[1.3] R. Gubrynowicz i inni: *Simplified system for isolated word recognition*. Archives of Acoustics, IFTR PAS Warszawa, vol. 15(1990), 287-300.
[1.4] A. Hanson, E. Riseman: *VISIONS, A Computer System for Interpreting Scenes*. [A. Hanson, E. Riseman (Eds.), *Computer Vision Systems*, Academic Press, New York, 1978], 303–333.
[1.5] W. Kasprzak: *Adaptive computation methods in digital image sequence analysis*. Prace naukowe, Elektronika, z. 127, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.
[1.6] E. Riseman, A. Hanson A.: *The VISIONS Image Understanding System*. [C. Brown (ed.), *Advances in Computer Vision*, Lawrence Erlbaum Ass. Pub., Hillsdale, N.J., 1988], 6–103.
[1.7] S. Russell, P. Norvig: *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River NJ, 2002.
[1.8] G. Sagerer, H. Niemann: *Semantic Networks for Understanding Scenes.* Advances in Computer Vision and Machine Intelligence. Plenum Press, New York and London, 1997.
[1.9] A. Stachurski, A.P. Wierzbicki: *Podstawy optymalizacji*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1999.
[1.10] R. Tadeusiewicz, M.R. Ogiela: *Medical Image Understanding Technology*, Series: Studies in Fuzziness and Soft Computing, vol. 156, Springer-Verlag, Berlin - Heidelberg - New York, 2004.
[1.11] `www.iapr.org`: oficjalna strona *The International Association for Pattern Recognition*, 2007.
[1.12] `www.tpo.org.pl`: oficjalna strona polskiego *Towarzystwa Przetwarzania Obrazów*, 2007.
[1.13] `cslu.cse.ogi.edu`: strona poświęcona rozpoznawaniu mowy, 2007.

Rozdział 2. Klasyfikacja prostych wzorców

Jakość klasyfikacji prostego wzorca zależy w ogólności od:

1. **jakości cech dla próbek trenujących** klasyfikator – przykłady przekształceń odpowiadających tym kryteriom to PCA i LDA, omawiane w pkt. 2.1;
2. prawdopodobieństwa i wartości **ryzyka** błędnej klasyfikacji – pkt. 2.2;
3. spodziewanego **błędu klasyfikacji** – to zależy od rodzaju klasyfikatora – typowe klasyfikatory omawiane są w pkt. 2.3-2.8.

2.1 Przekształcenia wzorca zależne od próbek uczących (*)

Omawiamy przekształcenia przestrzeni cech (**PCA, LDA**) optymalizowane ze względu na dane trenujące klasyfikatora, tzn. spełniające analitycznie zadane kryteria [1, 2.1, 2.3, 2.4, 2.7, 2.9]. Podajemy trzy kryteria optymalizacji liniowego przekształcenia wzorca w przestrzeń cech: 1) średni odstęp kwadratowy jednej wartości cechy od każdej innej (**PCA**), 2) średni kwadratowy odstęp cech jednej klasy od cech innych klas, 3) średni kwadratowy odstęp cech jednej klasy. Kombinacja s2) i 3) to tzw. **dyskryminant Fishera** stosowany w **LDA** – liniowej analizie dyskryminacyjnej**.**

2.2 Problem klasyfikacji prostego wzorca

Klasyfikacja wzorca wymaga podjęcia *decyzji* - w wyniku obserwacji aktualnego wektora cech i wiedzy nabytej uprzednio na podstawie zbioru uczącego (w procesie **uczenia**) [2, 1.1, 2.6]. Omawiamy miarę jakości klasyfikatora oparta o pojęcia prawdopodobieństwa i wartości **ryzyka** błędnej klasyfikacji.

2.3 Klasyfikator według funkcji potencjału

W tym podejściu zakładamy, że nie mamy dostępu do statystycznych danych opisujących klasy, tzn. nie dysponujemy funkcjami gęstości prawdopodobieństw warunkowych cech względem klas. Dalszymi założeniami są:

- dla każdej klasy istnieje parametryczna funkcja (tzw. funkcja *potencjału*) zdefiniowane nad przestrzenią cech, charakteryzująca stopień przynależności danego punktu przestrzeni do zadanej klasy;
- funkcje potencjału należą do jednej, zadanej rodziny parametrycznych funkcji.

2.4 Klasyfikator statystyczny Bayesa

Zakładamy istnienie statystyki klas w przestrzeni cech w postaci znanych nam:

- *a priori* prawdopodobieństw klas $p(\Omega_\kappa)$,
- a priori gęstości prawdopodobieństw warunkowych $p(\boldsymbol{c} \mid \Omega_\kappa)$,

dla wszystkich $\Omega_\kappa \in \boldsymbol{\Omega}$. Klasyfikator Bayesa poszukuje maksymalnego prawdopodobieństwa a posteriori.

Dla jednorodnego rozkładu klas ($p(\Omega_\kappa) = p(\Omega_\lambda)$ dla wszystkich par klas) reguła decyzyjna klasyfikatora Bayesa sprowadza się do reguły decyzyjnej *największej wiarygodności* (ang. *maximum likelihood*).

Prawdopodobieństwo błędu klasyfikatora Bayesa (oznaczenie: p_B) stanowi **dolną granicę** błędu dowolnego klasyfikatora, wtedy gdy funkcja kosztu wynosi: $\boldsymbol{r_{ii}} = 0$ (dla prawidłowej decyzji), $\boldsymbol{r_{ij}} = 1$ (dla błędnej decyzji). W takiej sytuacji jest to **optymalny klasyfikator**.

Uczenie rozkładów prawdopodobieństwa dla klasyfikatora Bayesa:

1. A priori prawdopodobieństwa klas $p(\Omega_k)$, $1 \leq k \leq K$, można wyliczyć na podstawie relatywnej częstości klas w zbiorze próbek.
2. Modele funkcji gęstości prawdopodobieństwa $p(\boldsymbol{c} \mid \Omega_k)$:

- **nieparametryczne:** zastosuj dyskretny rozkład prawdopodobieństwa w jawnej postaci (np. histogram);
- **parametryczne:** założenie istnienia rodziny funkcji gęstości i szacowanie ich parametrów na podstawie próbek uczących - zwykle przy założeniu statystycznej niezależności elementów wektora cech.

Dla rozkładu Gaussa wartości parametrów μ_k i $\boldsymbol{\Sigma}_k$ (dla każdej klasy Ω_k) mogą być oszacowane zgodnie z zasadą estymatora *największej wiarygodności* ML (ang. *maximum likelihood*). Dla innych postaci rozkładów niż rozkład normalny często nie jest możliwe uzyskanie dokładnych analitycznych postaci dla estymatorów największej wiarygodności. Wtedy należy zastosować iteracyjny sposób szacowania parametrów rozkładu, czyli rozwiązać „równania wiarygodności" *metodą Newtona* lub inną *gradientową metodą* iteracyjną. Przy takim podejściu wyznaczany jest w każdej iteracji optymalny kierunek w przestrzeni parametrów dla poszukiwania lokalnego maksimum „funkcji wiarygodności". Przykładem prostego iteracyjnego algorytmu, który nie posługuje się kierunkiem w przestrzeni parametrów a poszukuje rozwiązania problemu metodą kolejnych przybliżeń jest algorytm *maksymalizacji oczekiwań* EM (ang. *expectation maximization*).

2.5 Klasyfikator według minimalnej odległości

Jest to pewna specjalizacja klasyfikatora Bayesa przy zastosowaniu uproszczeń rozkładów priori i wyrażenia miary odległości prawdopodobieństwa przez **odległość Euklidesa** lub **Mahalanobisa** w przestrzeni cech.

2.6 Klasyfikator według „k sąsiadów"

Zakładamy istnienie zbioru próbek uczących i wcześniej klasyfikowanych wektorów cech, z których każda próbka względnie wektor cech należy do klasy zgodnej z regułą decyzyjną

tego klasyfikatora. Każda próbka względnie każdy kolejny wektor cech staje się reprezentantem swojej klasy. **Reguła decyzyjna** klasyfikatora według k sąsiadów przyporządkowuje nowemu wektorowi cech tę klasę, do której należy jego najbliższy sąsiad, względnie większość spośród *k* najbliższych sąsiadów.

2.7 Maszyna wektorów wspierających SVM (*)

W klasyfikatorze statystycznym zakłada się, że dysponujemy reprezentatywną próbką uczącą dla wszystkich klas. W SVM („Support Vector Machine") przyjmuje się skończony charakter próbek uczących i sprowadza problem do wielokrotnej decyzji pomiędzy dwiema klasami (lub grupami klas) (oznaczanymi zwykle jako „+1", „-1").
Podczas uczenia maszyny poszukuje się optymalnej hiper-płaszczyzny (dla przypadku liniowego) lub hiper-powierzchni (dla „nieliniowej" maszyny).
Klasyfikacja *k* klas przy użyciu binarnej maszyny SVM może być zrealizowana na różne sposoby. Wyróżnimy tu 3 strategie korzystania z binarnych klasyfikatorów. **„**jedna klasa przeciw wszystkim innym", „jedna przeciw jednej" i „jedna przeciw pozostałym".
Rozmiar Vapnika–Czervonenkisa (rozmiar VC) *h* jest miarą zbioru funkcji rozdzielających. Dla problemu dwóch klas *h* wyznacza maksymalną liczbę wzorców, które mogą być rozdzielone we wszystkie możliwe sposoby – liczba takich podziałów wynosi 2^h. Specyficznym zbiorem funkcji rozdzielających jest zbiór zorientowanych hiper-płaszczyzn. Rozmiar Vapnika-Chervonenkisa zbioru zorientowanych hiper-płaszczyzn w przestrzeni $\Re^n$ wynosi $h = n + 1$.
Szczegółowo wyprowadzamy **klasyfikator SVM dla liniowo separowalnego** zbioru próbek. Odwołujemy się przy tym do problemu optymalizacji metodą **wypukłego programowania kwadratowego** z nierównościowymi ograniczeniami liniowymi. Niezerowa wartość warunków dodatkowych w pewnym punkcie **a** przestrzeni parametrów szukanej hiperpłaszczyzny oznacza, że możliwe jest wykonanie małego przesunięcia z punktu **a** w dowolnym kierunku bez naruszenia tego ograniczenia. Okazuje się, że wektor pochodnych funkcji celu w punkcie optymalnym daje się zapisać jako liniowa kombinacja pochodnych ograniczeń. Dlatego też uzupełnimy funkcję celu *U* o liniową kombinację warunków dodatkowych – przemnożymy ograniczenia przez dodatnie mnożniki Lagrange'a ϑ i odejmiemy od kryterium optymalizacji *U*. Uzyskujemy zmodyfikowaną funkcję celu o postaci tzw. **funkcjonału Lagrange'a** *L*. Tę funkcję celu należy minimalizować względem **a** aż do chwili, gdy nie zanikną pochodne względem ϑ. Znane są w**arunki konieczne Karusha-Kuhna-Tuckera** dla optymalizacji funkcjonału *L*. Wykorzystujemy dalej przekształcenie problemu minimalizacji funkcjonału *L* w dualne zadanie Lagrange'a, które polega na poszukiwaniu maksimum dualnej funkcji Lagrange'a, czyli takiej postaci funkcjonału Lagrange'a, dla której zanikają pochodne względem szukanych parametrów hiperpłaszczyzny. Uzyskamy **dualny problem o postaci Wolfe'a**, który daje się już rozwiązać bezpośrednio dla wszystkich *h*-elementowych (*h* - rozmiar VC) podzbiorów próbek (wektorów cech), takich, że *n* (*n*=*h*-1) próbek należy do jednej klasy a pozostała próbka - do drugiej klasy. Po znalezieniu optymalnego wektora mnożników ϑ, dla każdego *h*-elementowego podzbioru próbek, wartości parametrów hiperpłaszczyzny (**a**) uzyskujemy z równań dla pochodnych funkcjonału *U*. Wybieramy rozwiązanie, dla którego wartość $2 / |a|$ jest maksymalna i zachodzi separacja wszystkich próbek za pomocą znalezionych hiper-płaszczyzn. Zauważamy też, że do wyznaczenia hiper-płaszczyzny wystarczą nam **produkty skalarne** wybranych wektorów cech (tzw. **wektory wspierające**). Ta obserwacja pozwala na uogólnienie rozważań do nieliniowej maszyny SVM opartej o hiper-powierzchnie wyznaczone przez dowolne wielomiany współczynników wektorów cech.

2.8 Klasyfikacja neuronowa

Omawiamy sieci typu **perceptron** (**jednokierunkowa** sieć, ang. „*feed-forward*"), w której sygnały wejściowe są przesyłane od wejścia do warstwy wyjściowej poprzez *połączenia pobudzające ,* co reprezentuje macierz wag połączeń $\boldsymbol{W}$.

Uczenie wag perceptronu - określamy sumaryczny kwadrat błędu perceptronu U (błędem jest różnica pomiędzy żądanym i rzeczywistym wyjściem sieci) - uczenie perceptronu polega na optymalizacji wartości U, czyli na poszukiwaniu jej minimalnej wartości metodą *najszybszego spadku w kierunku gradientu* U (ang. *gradient descent*) względem każdej z wag w_{ij} . Stąd wyprowadzamy regułę modyfikacji wagi stosowana podczas uczenia sieci.

Wielowarstwowy "perceptron" (MLP)

Taka sieć ma warstwę wejściową, warstwę ukrytą (jedną, lub więcej) i neurony warstwy wyjściowej. Można pokazać (Cybenko, 1989), że już 3-warstwowy perceptron o sigmoidalnej funkcji aktywacji i n^{hidden} neuronach w warstwie ukrytej, przy $n^{\text{hidden}} \rightarrow \infty$, może aproksymować dowolne zbiory w przestrzeni $\Re^n$, albo dowolną funkcję ciągłą zdefiniowaną w tej przestrzeni.

Uczenie z nadzorem

Omawiamy regułę Widrof-Hoffa (reguła „delta") i jej uogólnienie dla MLP - regułę **wstecznej propagacji błędu** (ang. *error backpropagation rule*) stosowaną w procedurze uczenia wag perceptronu MLP. Modyfikacja wag rozpoczyna się od ostatniej warstwy i przemieszcza się wstecz warstwa po warstwie, aż zakończy się na warstwie o indeksie 1. Podczas kroku modyfikacji propagowane są „wstecz" wartości korekty, obliczone początkowo dla najwyższej warstwy.

Uczenie bez nadzoru

Omawiamy reguły „Hebb" i „anty-Hebb".

Uczenie bez nadzoru w warunkach konkurencji (ang. „*competitive learning*")

Opisujemy *sieć Kohonena* i sieci *samoorganizujące się* (grupowanie, tworzenie klastrów) jako przykłady sieci stosujących **nienadzorowane uczenie w warunkach konkurencji.**

Uczenie z nadzorem w warunkach konkurencji (kwantyzacja wektorowa Kohonena)

Kohonen zaproponował także odmianę reguły uczenia w warunkach konkurencji dla uczenia z nadzorem. W tym przypadku każda próbka ucząca zaopatrzona jest w etykietę wyrażającą przynależność do określonej swojej klasy. Zastosowanie tej reguły uczącej odpowiada procesowi kwantyzacji wektorowej (ang. LVQ, *learning vector quantization*).

2.9 Zadania

Zadania dotyczą: definiowania i rozwiązywania problemu optymalizacyjnego występującego podczas projektowania klasyfikatorów dla zadanych próbek (cech), projektowania klasyfikatorów różnych rodzajów, wykonania (krótkiej) symulacji procesu uczenia 3-warstwowego perceptronu (jedna warstwa ukryta), wykonania symulacji procesu uczenia warstwy jedno-kierunkowej sieci stosującej regułę uczenia w warunkach konkurencji.

Literatura uzupełniająca do rozdziału 2

[2.1] N. Ahmed, K. Rao:. *Orthogonal Transforms for Digital Signal Processing*. Springer, Berlin, Heidelberg, New York, 1975.

[2.2] C. Burges: *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, vol. 2(1998), No. 2, 121-167.

[2.3] A. Cichocki, R. Unbehauen: *Neural Networks for Optimization and Signal Processing*. J. Wiley, New York, 1994.

[2.4] A. Hyvarinen, J. Karhunen, E. Oja: *Independent Component Analysis*, John Wiley & Sons, New York etc., 2001.

[2.5] S. Osowski: *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa 1996.

[2.6] J. Schurmann: *Pattern classification. A unified view of statistical and neural approaches*. John Wiley & Sons, New York 1996.
[2.7] W. Skarbek, K. Kucharski, M. Bober: *Dual Linear Discriminant Analysis for Face Recognition.* Fundamenta Informaticae, vol. 61, (2004), No.1, 303-334.
[2.8] R. Tadeusiewicz: *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.
[2.9] M. Turk, A. Pentland: *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, vol. 3 (1991), 71-86.

II. Rozpoznawanie obrazów

Rozdział 3. Reprezentacja obrazu cyfrowego

3.1 Akwizycja obrazu dla 3-wymiarowej sceny

Omawiamy **modele rzutowania** (projekcji sceny): rzut zbieżny (perspektywa) i rzut równoległy. **Współrzędne jednorodne** punktu przestrzeni dają nam możliwość zapisania podstawowych przekształceń układu współrzędnych i (dualnie) punktów przestrzeni w postaci jednej macierzy przekształceń. Podajemy macierze dla podstawowych **przekształceń** – przesunięcia, slakowanie, obroty. Wprowadzamy macierzowe przekształcenie dla **rzutu zbieżnego**.
Omawiamy zagadnienie **kalibracji kamery.** Przekształcenie między globalnym układem a układem kamery uzyskamy dzięki: a) **jawnej kalibracji** – gdy są nam znane wszystkie elementarne przekształcenia, b) **auto-kalibracji** kamery – w wyniku obserwacji obrazu znanego nam obiektu (wzorca) umieszczonego w znanym miejscu sceny. Podajemy szczegółowo **procedurę auto-kalibracji kamery**.

3.2 Wewnętrzna reprezentacja obrazu

Zdefiniujemy **klasę w języku C++** przeznaczoną do reprezentacji i działań na 2-wymiarowej tablicy o elementach sparametryzowanego typu. Będzie to klasa wzorcowa (szablonowa) **Matrix<T>**. **Cyfrowy obraz** jest obiektem klasy wzorcowej, np. IMAGE<T>, która korzysta z obiektu klasy Matrix<T>.
Omawiamy **przestrzenie reprezentacji barw HLS** (Hue, Lightness, Saturation) i **HSV** (Hue, Value, Saturation), które oparte są na badaniach **percepcji** kolorów przez **człowieka**. **W technice** przyjęły się modele reprezentujące kolor jako **mieszaninę** trzech kolorów **podstawowych**: typowy dla prezentacji na monitorach komputerowych - **RGB** (Red, Green, Blue) - model addytywny ("biel" jest sumą składowych); typowy dla wydruku na papierze - **CMY** (Cyan, Magenta, Yellow) - model negatywny ("czerń" jest sumą składowych), typowy dla techniki telewizyjnej - **YIQ** (Y - luminancja, I, Q - chrominancje wzgl. czerwonego i czerwono-niebieskiego); w reprezentacji obrazów cyfrowych w TV cyfrowej typową przestrzenią jest **YUV** (lub **Y Cb Cr**), gdzie U (Cb) i V (Cr) to miary odległości koloru szarego od koloru niebieskiego względnie czerwonego.

3.3 Zewnętrzna reprezentacja obrazu

W tym punkcie wskazujemy na podstawowe formaty reprezentacji pojedynczych obrazów cyfrowych i sekwencji obrazów cyfrowych przeznaczone do ich przechowywania na zewnętrznych nośnikach danych. Wybrane formaty plików graficznych: TIFF (ang. *Tagged Image File Format*), GIF (ang. *Graphics Interchange Format*), PBM (ang. *Portable Bitmap Format*).
Omawiamy **algorytmy JPEG** dla kompresji obrazu - tryby kompresji obrazu: 1) tryb **sekwencyjny** jest podstawowym trybem kompresji JPEG, 2) tryb **niestratny**, 3) tryb **progresywny**, 4) tryb **hierarchiczny**, 5) tryb "***Motion JPEG***", który jest podstawowym sposobem kompresji ramek typu I w sekwencji obrazów.

Omawiamy jedynie pod względem użytkowym cyfrowe formaty **audio/wideo** z rodziny **MPEG** (ang. *Motion Picture Expert Group*) - oznacza rodzinę algorytmów kompresji i dekompresji dla sekwencji cyfrowych danych audio i wideo.
Grupowanie kolorów pikseli w JPEG i MPEG polega na tym, że obraz kolorowy RGB przekształcany jest na przestrzeń YUV (lub YC_bC_r) i składowe chrominancji U i V (C_b i C_r) zapisuje się wspólnie dla kilku (2 lub 4) sąsiednich pikseli.

3.4 Zadania

Zadania dotyczą: procesu auto-kalibracji kamery, zdefiniować klasy w języku C++ dla reprezentacji obrazów równych typów z wykorzystaniem klasy **Matrix<T>** (w tym zdefiniowania operatora indeksowania **[ ]** w tych klasach, zdefiniowania funkcji w języku C++ obliczająca wartość średnią w zadanym fragmencie macierzy obrazu, analizowania procesu kalibracji kolorów.

Literatura uzupełniająca do rozdziału 3

[3.1] C.-S. Fu, W. Cho, S. K. Essig. *Hierarchical colour image region segmentation for content-based image retrieval system*. IEEE Transactions on Image Processing, vol. 9 (2000), No. 1, 156-162, 2000.
[3.2] Haskell B.G., Puri A., Netravali A.N.: *Digital video: an introduction to MPEG-2*, New York, Chapman & Hall, September 1996.
[3.3] Held G., Marshall T. R.: *Data and image compression: tools and techniques*, John Willey and Sons Ltd., NY, 1996.
[3.4] ISO/IEC JTC1/SC29/WG11 N3908: *MPEG-4 Video Verification Model*, ver. 18.0, Pisa, January 2001.
[3.5] B.S. Manjunath, P. Salembier, T. Sikora: *Introduction to MPEG-7: Multimedia Content Description Interface,* John Wiley & Sons, Ltd., 2002.
[3.6] I. E.G. Richardson: *H.264 and MPEG-4 video compression*. John Wiley & Sons, Chichester UK, 2005.
[3.7] `www.chiariglione.org`: oficjalna strona "Motion Picture Experts Group", 2007.
[3.8] `www.itu.org` : oficjalna strona "Internatonal Telecommunication Union", 2007.

Rozdział 4. Segmentacja obrazu i detekcja cech

4.1 Przekształcenia początkowe obrazu

Problem **binaryzacji obrazu** rozwiązywany jest poprzez dopasowanie rozkładu mieszanego z dwóch rozkładów Gaussa do histogramu obrazu.
Problem **normowania 2-wymiarowego kształtu** w obrazie rozwiązywany jest dzięki przekształceniom obrazu wyznaczonym przez odpowiednie funkcje momentów geometrycznych. Wyróżniamy kroki przekształceń: przesunięcie do środka masy i normalizacja amplitudy, normalizacja rozmiaru – skalowanie osi, obrót na główną oś, odbicie lustrzane względem osi Y.

4.2 Obraz krawędziowy

Omawiana jest dwu-stopniowa detekcja **cech liniowych** w obrazie: 1. wyznaczenie obrazu krawędziowego, 2. wyznaczenie segmentów liniowych w obrazie krawędziowym. **Wierzchołki** mogą być wykrywane na dwa sposoby: w procesie tworzenia obrazu krawędziowego, dzięki właściwym operatorom krawędziowym, lub jako przecięcia przynajmniej dwóch uprzednio znalezionych segmentów liniowych.
Omawiamy typowe dwie grupy **filtrów obrazu**: przeznaczone do usunięcia górnych częstotliwości w obrazie (wygładzanie obrazu, usunięcie szumu) i do usunięcia dolnych

częstotliwości (detekcja krawędzi, wyostrzanie obrazu). Definiujemy pojęcie filtru **liniowego** i **nieliniowego**. Podajemy przykłady ważnych filtrów dla przetwarzania cyfrowego obrazu: *filtr uśredniający*, *filtr mediany*, *filtr Gaussa* (wygładzanie, redukcja szumu), *gradientowe filtry* (detekcja krawędzi – pierwsza pochodna funkcji obrazu), *filtr Laplace'a* (detekcja krawędzi – druga pochodna funkcji obrazu).

Omawiamy szczegółowo kilka znanych filtrów gradientowych aproksymowanych w postaci **operatorów krawędziowych**. Są to: *krzyż Robertsa*, operator różnic centralnych, *operator Sobela*, *Prewitta* i inne. Podajemy aproksymacje *filtru Laplace'a* przy pomocy operatorów krawędziowych. Uogólniamy operator krawędziowy Sobela na operator **krawędziowy kolorowy**.
Wskazujemy na inne podejścia niż stosowanie odpowiednich operatorów krawędziowych również przeznaczone do wyznaczania obrazu krawędziowego: dopasowanie otoczenia piksela z parametrycznym **modelem krawędzi** lub **kombinacje** 2 lub więcej metod w celu zwiększenia jakości wyniku lub podniesienia efektywności obliczeń.
Następnym krokiem jest **pocienianie krawędzi** w obrazie krawędziowym. Podajemy trzy sposoby realizacji tego kroku przetwarzania obrazu, w szczególności wykorzystujące cechy **lokalnego otoczenia** do modyfikacji siły krawędzi zadanego piksela.

4.3 Segmenty liniowe

Podajemy dwa alternatywne sposoby wyznaczania cech liniowych w oparciu o obraz krawędziowy: detekcja **segmentów liniowych** (połączyć punkty w łańcuchy krawędzi, aproksymować łańcuchy odcinkami prostych), detekcja linii i okręgów dzięki **transformacji Hough'a**.
Dla rozwiązania pierwszego zagadnienia podajemy najpierw szczegółowe opisy dwóch algorytmów **detekcji łańcucha krawędzi**: *algorytm Nevatia-Babu* i algorytm „*progów histerezy*".
Omawiamy tzw. **operator Canny'ego** – całościowy algorytm łączący w sobie kolejno etapy wygładzania obrazu, detekcji obrazu krawędziowego, cieniowania krawędzi i detekcji łańcuchów krawędziowych.
Następnie podajemy algorytm **aproksymacji łańcucha krawędzi** zbiorem odcinków prostych.

4.4 Przekształcenia Hough'a dla wykrycia linii

Poszukiwanie linii w obrazie krawędziowym może też realizowane bezpośrednio w oparciu o elementy krawędziowe dzięki zastosowaniu jednej z odmian tzw. przekształcenia Hough'a (HT). Przedstawiamy **podstawową procedurę** przekształcenia Hough'a dla detekcji linii w obrazie a także sposoby wykorzystania tego przekształcenia do **detekcji środków okręgów** w obrazie i do detekcji samych **okręgów**. Podajemy wreszcie uogólnienie tego przekształcenia (GHT) dla wykrywania dowolnego **2-wymiarowego konturu** w obrazie.

4.5 Obszary jednorodne obrazu

Wprowadzamy **kryteria jednorodności** obszaru w obrazie, które obok wymogu spójności pikseli pozwalają na wyznaczanie obszarów jednorodnych. Omawiamy zasadnicze **elementy** takich **algorytmów**: rozrastanie obszaru (ang. *growing*), łączenie obszarów (ang. *merging*) i technikę "dziel i łącz" (ang. *split-and-merge*). Na koniec formułujemy **szczegółowy algorytm** wyznaczania obszarów jednorodnych w oparciu o omawiane wcześniej elementy i kryteria jednorodności.

4.6 Tekstura w obrazie

Obszary niejednorodne w obrazie charakteryzują się określoną teksturą, czyli pewnymi regularnymi cechami powierzchni obrazowanego obiektu. Wyznaczanie tekstur w obrazie ma charakter dwu-etapowy: najpierw wyznaczamy **wektor cech** dla bloku obrazu a następnie **klasyfikujemy** go do jednej z zadanych klas tekstur. Podstawowy sposób wyznaczania cech tekstur to wielokrotna **filtracja bloków obrazu** realizowana dla zbioru uzupełniających się (zwykle od kilku do kilkudziesięciu) **masek**. Uzyskuje się w ten sposób dla każdego bloku w obrazie wektor cech obszaru obrazu względem każdej z funkcji bazowych (reprezentowanych kolejnymi maskami).
Podajemy zbiory masek aproksymujących **ważone sumy** i **różnice** kilku **funkcji Gaussa**, które przyjmują postać charakterystycznych punktów (ang. *spots*) i pasków (ang. *bars*).
Drugim rodzajem masek omawianym szczegółowo są maski **filtrów Gabora**, czyli aproksymacje funkcji Gabora tworzonych z elementów funkcji bazowych transformaty Fouriera pomnożonych przez funkcje Gaussa. Filtry Gabora definiowane są parami – jeden wykrywa **symetryczne** własności w określonym kierunku, a drugi - **anty-symetryczne** własności.
Szczegółowo omawiane są także dwie dalsze metody pozyskiwania cech tekstur: metoda **macierzy spójności wartości jasności** i metoda **histogramów sum i różnic** par pikseli.

4.7 2-wymiarowe kształty

Podajemy sposoby charakteryzowania 2-wymiarowych kształtów w obrazie, ważne niezależnie od tego czy kształt zadany jest w postaci zbioru konturów czy też w postaci jednego obszaru. Do charakterystyk określanych mianem „**liczb znamionowych”** zaliczamy wielkości będące wynikiem specyficznych "procesów pomiarowych", a nie opartych na określonym analitycznie zadanym przekształceniu przestrzeni reprezentacji (np. proces zliczania punktów przecięć kształtu z odpowiednio dobranymi prostymi, współczynniki wypełnienia kształtu). Innym sposobem charakteryzowania kształtów jest skorzstanie z momentów geometrycznych (czyli potraktowania obszaru obiektu jako 2-wymiaorowej funkcji gęstości rozkładu prawdopdobieństwa). Podajemy siedem **funkcji opartych o momenty geometryczne** róznych rzędów, które są **niezmiennicze** względem podstawowych przekształceń przesztrzeni reprezentacji kształtu.
Omawiamy też inny popularny sposób charakteryzowania kształtu zadanego przy pomocy otaczającego go konturu. Jest nim wyznaczenie **1-wymiarowej funkcji odległości** punktów konturu od środka masy konturu a następnie analizowanie przebiegu tej funkcji – tzn. normalizacja i wyznaczanie cech takich, jak położenie minimów i maksimów, cechy statystyczne i częstotliwościowe.
Na koniec rozdziału omawiamy tzw. **metodę “aktywnego konturu”**, czyli algorytm iteracyjnego modyfikowania krzywej, reprezentującej kontur obiektu w obrazie, sterowany minimalizacją energii związanej z tą krzywą i jej otoczeniem w obrazie.

4.8 Zadania

Zadania dotyczą analizy omawianych algorytmów segmentacji obrazu poprzez prześledzenie wyników ich pracy dla obrazów o małej rozdzielczości (np. o rozmiarze 5x4 piksele). Wyznaczane i porównywane ze sobą są wyniki operatorów krawędziowych, algorytmów detekcji łańcuchów krawędziowych, przekształceń Hough'a przeznaczonych do detekcji linii prostych i środków okręgów, uogólnionego przekształcenia Hough’a do detekcji dowolnego konturu, algorytmu wyznaczania obszarów jednorodnych i cech tekstur.

Literatura uzupełniająca do rozdziału 4

[4.1] A. J. Abrantes, J. S. Marques. *A class of constrained clustering algorithms for object boundary extraction.* IEEE Trans. on Image Processing, 5(1996), No. 11, 1507-1521.
[4.2] J. Canny: *A computational approach to edge detection*. IEEE Trans Patt. Anal. Mach. Intell, vol. 6 (1986), 679-698.
[4.3] M. Kass, A. Witkin, Terzopoulos D.: Snakes: *Active Contour Models*. International Journal of Computer Vision, 18:321{331, 1988, No. 1.
[4.4] H. Kauppinen, T. Seppanen, M. Pietikainen: *An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification*. IEEE Trans. Patt. Anal. Mach. Intell., vol.17(1995), No. 2, 201-206.
[4.5] W. Malina, M. Smiatacz: *Metody cyfrowego przetwarzania obrazów*. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2005.
[4.6] B.S. Manjunath, W.Y. Ma: *Texture features for browsing and retrieval of image data.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18(1996), No. 8, 837-842.
[4.7] S. Osher, N. Paragios, Editors. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer Verlag, New York, 2003.
[4.8] T. Raden, J.H. Husoy, *Filtering for texture classification: A comparative study*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21 (1999), No. 4, 291-310.
[4.9] K.R. Rao, R. Yip: *Discrete cosine transform – algorithms, advantages and applications*. Academic Press Inc, San Diego, 1990. .
[4.10] C-Y. Xu, J. L. Prince. *Snakes, Shapes and Gradient Vector Flow*. IEEE Trans. on Image Processing, 7(1998), No. 3, 359-369.

Rozdział 5. Rozpoznawanie 2- i 3-wymiarowych obiektów

5.1 Sekwencja wzorców i programowanie dynamiczne

Omawiamy proces statystycznej klasyfikacji **sekwencji prostych wzorców** jako procesu Markowa 1-szego rzędu. Wprowadzamy strategię **przeszukiwania** przestrzeni dopasowania modelu złożonego wzorca do sekwencji (pomiarowej) prostych wzorców zwaną **programowaniem dynamicznym**. Pokazujemy, jak definiować model złożonego wzorca tak, aby można było zastosować programowanie dynamiczne w warunkach błędów sekwencji pomiarowej - występowania **pojedynczych lub wielokrotnych przekłamań** o różnym charakterze.

5.2 Rozpoznawanie znanego 2-wymiarowego lub 3-wymiarowego obiektu

Rozpoznawanie znanego (i sztywnego) **pojedynczego obiektu** w obrazie jest rozszerzeniem problemu klasyfikacji sekwencji wzorców prostych, gdyż wymaga ona **dodatkowo** rozwiązania dwóch problemów: dopasowania (pod-)zbioru segmentów z (pod) zbiorem elementów reprezentacji modelu obiektu i określenie położenia w przestrzeni wykrytego obiektu. Jeśli model problemu zawiera **wiele** alternatywnych **obiektów** to w procesie rozpoznawania potrzebna nam jest **dodatkowo strategia generowania / weryfikowania** i **porównywania** hipotez - alternatywnych wyników dla instancji różnych obiektów. Podajemy s**trategie optymalne** i **heurystyczne** dla rozwiązania tego problemu. **Strategia optymalna** polega na systematycznym przeszukiwania przestrzeni dopasowania wszystkich możliwych instancji wszystkich możliwych modeli z aktualnym zbiorem segmentów, oceną jakości tych instancji i wyborem najlepszej z nich. Jest ona w praktyce trudna lub niemożliwa do zrealizowania w większości zastosowań. Dlatego też w praktyce stosuje się **strategie heurystyczne**, które prowadzą do znalezienia „dobrych" rozwiązań po ograniczonym czasie obliczeń. Omawiamy **strategie heurystyczne** o generalnym charakterze: strategia „*dopasowanie sterowane danymi*", strategia „*dopasowanie sterowane modelem*", strategia „*generuj-weryfikuj hipotezy*". Podajemy szczegółowy **algorytm rozpoznawania 3-wymiarowej bryły** jako przykład strategii "*generuj-weryfikuj hipotezy*".

5.3 Przeszukiwanie przestrzeni rozwiązań

Przedstawiamy **algorytm A*** (znaną strategię optymalnego przeszukiwania grafu decyzji) i pokazujemy jego **adaptację jako generalną strategię rozpoznawania** wielo-obiektowych obrazów.

5.4 Rozpoznawanie 3-wymiarowego obiektu o parametrycznym modelu(*)

Jeśli model reprezentuje pewną klasę 3-wymiarowych obiektów (tzw. *generyczny model*) to zwykle przyjmuje on postać wektora parametrów (tzw. wektor stanu) przeznaczonych do reprezentacji rozmiaru, położenia i ewentualnie dalszych zindywidualizowanych cech instancji takiej klasy, czyli konkretnego obiektu sceny. Rozpoznawanie takich obiektów wyrazimy w terminach **nadążnej estymacji wektora stanu** (określanej też w teorii estymacji mianem *identyfikacji modelu*), gdzie nieznane parametry stanu są szacowane (lub identyfikowane) na podstawie zbioru obserwacji (pomiarów) przy spełnieniu pewnego kryterium optymalizacji.
Omawiamy główne **rodzaje estymatorów stanu** systemu stacjonarnego: statystyczne estymatory ML i MAP oraz niestatystyczne estymatory LSE i MMSE.
Podajemy przykład estymacji MAP (maksymalizacja rozkładu prawdopodobieństwa *a posteriori*) dla stanu 3-wymiarowego obiektu dłoni (przy określonym ułożeniu palców).

5.5 Zadania

Zadania dotyczą definiowania modeli złożonych wzorców (sekwencji liter pisanych) tolerujących przekłamania w postaci grafów o sekwencyjnym „lewo-prawym" charakterze i zastosowania dla nich metod przeszukiwania: programowania dynamiczne i A*.
Zadania dotyczą też projektowania algorytmów dla realizacji strategii „generuj-weryfikuj" i estymacji wektora stanu metodą MAP dla pojedynczych brył (ostrosłup, sześcian).

Literatura uzupełniająca do rozdziału 5

[5.1] P.J. Besl, R.C. Jain: *Three-dimensional object recognition*. ACM Computing Surveys, vol.17 (1985), 75-145.
[5.2] R. Bellman, R. Kalaba: *Dynamic Programming and Modern Control Theory*. Academic Press, New York, 1965.
[5.3] W.E.L. Grimson: Object *Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, Cambridge, MA, 1990.
[5.4] O. Faugeras: *Three-dimensional computer vision. A geometric viewpoint*. The MIT Press. Cambridge, Mass. 1993.
[5.5] J. Pearl: *Heuristics. Intelligent Search Strategies for Computer Problem Solving*. Addison–Wesley, Reading, Mass., 1984.
[5.6] J.M. Rehg, T. Kanade. *Digit Eyes: Vision-Based Human Hand Tracking*. Report CMU-CS-93-220, School of Computer Science, Carnegie Mellon University, 1993.

Rozdział 6. Estymacja ruchu w sekwencji obrazów

6.1 Detekcja i estymacja ruchu w obrazie

Na początek omawiamy metody **detekcji binarnej mapy** ruchu dla sekwencji obrazów czasie. Następnie wprowadzamy pojęcie **optycznego potoku** jako aproksymacji ruchu (jego wielkości i kierunku) w obrazie wykonanej dla pojedynczych pikseli obrazu. Wspominamy o problemie **apertury** wynikającym z lokalnego charakteru badanego otoczenia piksela.

6.2 Optyczny potok

Definiujemy **kryterium optymalizacji** dla wyznaczenia optycznego potoku w obrazie a następnie podajemy algorytm Horna-Schuncka iteracyjnego sposobu aproksymacji optycznego potoku.

6.3 Ruch dyskretnych cech obrazu

W tym punkcie podajemy algorytmy wyznaczania ruchu segmentów obrazu poprzez poszukiwanie korespondujących ze sobą segmentów w kolejnych obrazach zadanej sekwencji. Podajemy **strategię hierarchicznego dopasowywania** ze sobą **bloków** w kolejnych obrazach. Omawiamy problematykę **wyznaczania punktów charakterystycznych** operatorem **Moravca** i wyznaczania **cech segmentów** liniowych, dogodnych do pasowania par tych segmentów. Podajemy **strategie pasowania** ze sobą par takich segmentów.

6.4 Zadania

Zadania polegają na prześledzeniu działania różnych algorytmów wyznaczania ruchu w obrazie przy operowaniu na parach obrazów o małych rozmiarach. Badana jest metoda gradientowa Horna-Schuncka dla aproksymacji optycznego potoku, metoda hierarchicznego pasowania bloków obrazu i wyznaczania oraz pasowania kilku punktów charakterystycznych według „strategii najbliższego sąsiada".

Literatura uzupełniająca do rozdziału 6

[6.1] J.K. Aggarwal, N. Nandhakumar: *On the computation of motion from sequences of images - a review*. Proceedings of the IEEE, vol. 76(1988), No. 8, 917-935.
[6.2] J.L. Barron, D.J. Fleet, S.S. Beauchemin: *Performance of optical flow techniques. systems and experiment.* International Journal of Computer Vision, vol. 12(1994), No.1, 43-77.
[6.3] I.J. Cox: *A review of statistical data association techniques for motion correspondence*. International Journal of Computer Vision, vol. 10(1993), No. 1, 53-66.
[6.4] R. Deriche, O.Faugeras: *Tracking line segments*. Image and Vision Computing, vol. 8(1990), No. 4, 261-270.
[6.5] B. K. P. Horn, B. G. Schunck. *Determining optical flow*. Artificial Intelligence, vol. 17 (1981), 185-203.
[6.6] H. Kirchner: *Bewegungserkennung in Bildfolgen: Ein mehrstufiger Ansatz*, Deutscher Universitatsverlag, Wiesbaden, 1993.
[6.7] S.-P. Liu, R.~Jain: *Motion Detection in Spatio-Temporal Space*, Computer Vision Graphics and Image Processing, vol. 45(1989), 227-250.
[6.8] H.-H. Nagel: *On a constraint equation for the estimation of displacement rates in image sequences,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11 (1989), 13-30.
[6.9] B. Schunck: *The image flow constraint equation*, Computer Vision Graphics and Image Processing, vol. 35 (1986), 20-46.
[6.10] Z. Zhang: *Token tracking in a cluttered scene*. Image and Vision Computing, vol. 12(1994), No. 2, 110-120.

III. Rozpoznawanie sygnałów mowy

Rozdział 7. Reprezentacja sygnału mowy

7.1 Reprezentacja cyfrowego sygnału mowy

Przypominamy **pojęcie PCM** ("modulacja impulsowo-kodowa") jako typowy format cyfrowej reprezentacji (wewnętrznej w systemie cyfrowym) sygnałów dźwięku.

Podajemy **strukturę pliku WAV** jako typowego format zewnętrznej reprezentacji sygnału dźwięku.

7.2 Układ słuchu człowieka

Poznajemy zasady budowy układu słuchu człowieka i wyciągamy wnioski co do potrzeby częstotliwościowej dekompozycji sygnału mowy.

7.3 Transformata Fouriera

Definiujemy **szereg Fouriera** jako reprezentację dowolnej funkcji w czasie poprzez spektrum zawartych w niej częstotliwości. W praktyce do wykonania dekompozycji funkcji dyskretnej definiowanej w skończonym okresie czasie służy **cyfrowa transformata Fouriera.** Szczegółowo omawiamy algorytm **szybkiej Transformaty Fouriera (FFT)**.

7.4 Transformata falkowa

Podajemy algorytm dekompozycji sygnału według transformaty falkowej jako alternatywny do transformaty Fouriera sposób uzyskiwania **reprezentacji czasowo-częstotliwościowej** sygnału.

7.5 Zadania

W ramach zadań badane są najpierw podstawowe własności transformaty Fouriera dla sygnału o wartościach rzeczywistych. Krok po kroku obliczamy wyniki pośrednie szybkiej transformaty FFT i transformaty falkowej dla sygnału o długości 8 próbek.

Literatura uzupełniająca do rozdziału 7

[7.1] B. Adamczyk, K. Adamczyk, K. Trawiński: *Zasób mowy ROBOT*. Biuletyn IAiR nr 12, Wojskowa Akademia Techniczna, Warszawa, 2000.
[7.2] S. Grocholewski: *Wykorzystanie bazy nagrań CORPORA do automatycznej segmentacji zbiorów nagrań*. [W. Jassem: *Speech and Language Technology*, vol. 4 (2000)], 43-55.
[7.3] J.-C. Junqua, J.-P. Haton: *Robustness in automatic speech recognition*. Kluwer Academic Publications, Boston etc. (1996)
[7.4] S. W. Smith: *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, California 1999.
[7.5] R. Tadeusiewicz: *Sygnał mowy*. WKiŁ, Warszawa 1988.

Rozdział 8. Detekcja sygnału mowy

8.1 Usuwanie szumu

Omawiamy filtrację sygnału **filtrem dolno-przepustowym** jako sposób na usunięcie szumu Gaussa w sygnale pomiarowym. Dla szumu o innej charakterystyce proponujemy metodę tzw. **spektralnego odejmowania**. Wprowadzamy pojęcie **filtru "preemfazy"** stosowanego dla wstępnego wzmocnienie składowych o wyższych częstotliwościach.

8.2 Cechy sygnału w dziedzinie czasu

Cechy sygnału mowy w dziedzinie czasu nie odgrywają zasadniczej roli w procesie detekcji cech sygnału mowy. Pełnią jedynie pomocniczą rolę w procesie detekcji samego sygnału, jego wstępnej segmentacji na głoski i ewentualnej normalizacji sygnału. **Cechami** tymi są: 1) energia w oknie sygnału, 2) auto-korelacja sygnału, 3) znormalizowana korelacja wzajemna. Podajemy algorytm **wstępnej segmentacji** sygnału mowy, tzn. jego podziału na ramki o identycznym czasie trwania.

8.3 Zadania

Zadania dotyczą wyznaczania cech w oknach sygnału i wstępnej segmentacji sygnału w oparciu o te cechy.

Literatura uzupełniająca do rozdziału 8

[8.1] L.F. Lamel i inni: *An Improved Endpoint Detector for Isolated Word Recognition*. IEEE Trans. Acoustics, Speech and Signal Processing, vol. ASSP–29, No. 4, August 1981.
[8.2] M. Piasecki, S. Zyśko: *Rozpoznawanie granic słowa w systemie automatycznego rozpoznawania izolowanych słów*. Raport, Wydziałowy Zakład Informatyki Politechniki Wrocławskiej, Wrocław, 1999.
[8.3] U. Glavitsch: *Speaker normalization with respect to F0: a perceptual approach*. TIK-Report nr 185, Eidgenössische Technische Hochschule Zürich, Institut für Technische Informatik und Kommunikationsnetze, December 2003.

Rozdział 9. Wyznaczanie cech sygnału mowy

W tym rozdziale przedstawiamy dwa alternatywne podejścia do wyznaczania cech dla okien sygnału: 1) współczynniki "**mel-cepstralne**" (MFCC) uzyskane w oparciu o uprzednią transformatę Fouriera , 2) cechy według **liniowej predykcji** (LPC).

9.1 Cechy *mel-cepstralne* sygnału mowy

Definiujemy poszczególne przekształcenie w procesie pozyskiwania cech mel-cepstralnych: 1) podział sygnału na okna, 2) okienkowa transformata Fouriera w oparciu o okno Hamminga, 3) amplitudy współczynników Fouriera dla okna sygnału, 4) filtry pasmowe trójkątne rozmieszczone według Mel-skali i uzyskanie cech mel-spektralnych, 5) transformacja do cech mel-cepstralnych, 6) „liftrowanie" cech MFCC.
Uzupełniamy wektor cech o sumaryczną energię w oknie i o cechy różnicowe dla wektora MFCC, liczone względem 5 kolejnych okien sygnału.

9.2 Cechy według liniowej predykcji (LPC)

Wprowadzamy model **filtru liniowego FIR** dla modelowania układu generacji (syntezy) mowy. Odpowiada temu stosowanie modelu **LPC** (kodowania według liniowej predykcji) dla analizy sygnału mowy - problem poszukiwania współczynników tego filtra dla pobudzenia impulsowego deltą Diraca (poszukiwanie odpowiedzi impulsowej).
Omawiamy schemat wyznaczania parametrów LPC metodą **auto-korelacji** sygnału i według **iteracyjnej metody Levinsona**. Dyskutujemy **sposoby odwzorowania** tak znalezionych współczynników LPC na cechy okna sygnału.

9.3 Klasyfikacja cech ramki

Do klasyfikacji wektora cech okna sygnału w terminach podstawowych jednostek fonetycznych możemy alternatywnie:

1. zastosować wcześniej nauczony **klasyfikator** (np. neuronowy) wtedy, gdy próbki uczące są w pełni opisane w terminach klas – jednostek fonetycznych, lub
2. zastosować **klasteryzację** połączoną z **kwantyzacją wektorową** – gdy nie dysonujemy w pełni etykietowanymi próbkami uczącymi.

Klasyfikatory omówione zostały w rozdziale 2. Tutaj przedstawiamy podstawowe algorytmy **klasteryzacji** i **kwantyzacji wektorowej**.

9.4 Częstotliwość podstawowa mówcy

Omawiamy podstawowe elementy **prozodii**, czyli analizy metryczno-intonacyjnej sygnału mowy, która zajmuje się **brzmieniową informacją**, taką jak akcent, intonacja, "*melodia mowy*", obejmująca **więcej niż jeden** segment-fonem.

Dla wyznaczenia cech prozodii badamy zmienność **akustycznych parametrów**, takich jak: podstawowa częstotliwość mowy, energia sygnału, czas trwania, struktura spektralna i przerwy w mowie. Podajemy **algorytm** automatycznego wyznaczania **częstotliwości podstawowej sygnału** mowy. Proponujemy algorytm **normowania cech** okna sygnału mowy w dziedzinie częstotliwości zależnie od rodzaju głoski i aktualnej częstotliwości podstawowej F0.

9.5 Zadania

Zadania polegają na obliczaniu wektorów cech typu MFCC i LPC, dla zadanych krótkich okien sygnału o krótkim czasie trwania, i na symulacji procesu klasteryzacji i kwantyzacji wektorowej dla krótkiego zestawu wybranych cech. .

Literatura uzupełniająca do rozdziału 9

[9.1] P. Alexandre, Lockwood, P. Root cepstral analysis: A unified view, applications to speech processing in car environments. *Speech Communication*, 12(1993), 277-288.
[9.2] S.B. Davis, P. Mermelstein. *Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences*. IEEE Trans. on Acoustics, Speech, and Signal Processing, 28(1980), 357-366.
[9.3] C. Lee, i inni: *Optimizing feature extraction for speech recognition*. IEEE Trans. on Speech and Audio Processing, 11(2003), 80-87.
[9.4] J.D. Markel, A.H. Gray Jr: *Linear prediction of speech*, Springer, Berlin, 1976.

Rozdział 10. Akustyczno-fonetyczny model mowy

10.1 Fonetyczne kategorie dźwięków

Wprowadzamy podstawowe pojęcia fonetyczne: f**on** (lub dźwięk), **fonem** (głoska), układ mowy (wytwarzanie i miejsca **artykulacji** dźwięków) i podział zbioru fonemów na 7 grup.
Omawiamy poszczególne **grupy fonemów**: monoftongi, dyftongi, spółgłoski ustne (pół-samogłoski i sonanty), nosowe, spiranty, zwarte i afrykaty.
Omawiamy notację według **międzynarodowego alfabetu fonetyki (IPA)** i jego podzbioru komputerowego – alfabetu **Worldbet.**

10.2 Typowe spektrogramy dla grup fonemów

Omawiamy pojęcie **formantów** jako obszarów w spektrogramie **o d**użej koncentracji energii. Podajemy podstawowe **charakterystyki spektralne** dla 7 wcześniej wyróżnionych grup fonemów.

10.3 Dekompozycja fonemu zależna od kontekstu

Wyjaśniamy potrzebę podziału fonemów na części, tzw. **trzy-fony**. Są to modele części głosek zależne od kontekstu lewego (poprzedzającej głoski) i prawego (następnej głoski w wypowiedzi). Wyróżniamy osiem **kategorii kontekstowych** dla głosek. Omawiamy etapy procesu **projektowania klasyfikatora** dla trzy-fonów.

10.4 Zadania

Zadania mają na celu opanowanie fonetycznych transkrypcji słów w terminach fonemów, analizę postaci spektrogramów dla wybranych fonemów i projektowanie zbiorów trzy-fonów potrzebnych dla systemu rozpoznawania słów z ograniczonego słownika.

Literatura uzupełniająca do rozdziału 10

[10.1] R. J. Gubrynowicz: *Komputerowe modelowanie artykulacji głosek języka polskiego*. Monografia. Prace IPPT PAN, Warszawa, 2001, nr. 4.

[10.2] W. Jassem: *Podstawy fonetyki akustycznej*. PWN, Warszawa, 1973.
[10.3] T. Lander, T. Carmell: *Structure of Spoken Language: Spectrogram Reading*, Centre for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, USA, February 1999.
[10.4] M. Steffen-Batogowa: *Automatyzacja transkrypcji fonematycznej tekstów polskich*. Warszawa, PWN 1975.

Rozdział 11. Rozpoznawanie słów i zdań

W tym rozdziale omawiamy s**tatystyczne podejście** do rozpoznawania sekwencji (fonemów, sylab lub słów), w którym zmienność sygnału mowy modelowana jest statystycznie (ukryte modele Markowa) a parametry model wyznaczane są dzięki automatycznym procedurom uczenia [11.1]-[11.3].

11.1 Dopasowanie z reprezentantem wzorca metodą „marszczenia czasu"

Dynamiczne dopasowanie wzorca w czasie, tzw. „*marszczenie czasu*", to początkowa metoda rozpoznawania mowy, w której zwrócono uwagę na potrzebę dopasowywania ze sobą sekwencji o różnych długościach.

11.2 Rozpoznawanie jako statystyczne wnioskowanie

Modelujemy rozpoznawanie mowy jako system **podejmowania decyzji w warunkach niepewności**, wymagający mechanizmu **statystycznego wnioskowania.** Korzystamy z reguły Bayesa co oznacza potrzebę stworzenia dwóch statystycznych modeli apriori: **model fonetyczny** – określa prawdopodobieństwo warunkowe pomiaru sygnału dla wypowiedzi zdania i **model języka** – określa prawdopodobieństwa wystąpienia zadanej sekwencji słów. W probabilistycznym modelu języka stosujemy przybliżenie rzeczywistego świata, zakładając uwarunkowanie słowa jedynie od bezpośrednio poprzedniego słowa sekwencji (tzw. *model bigramowy*). Model fonetyczny oparty jest na reprezentacji **ukrytych modeli Markowa** (HMM). Wprowadzamy definicję HMM, różne **struktury lewo-prawych** modeli HMM (pełny, model Bakisa, liniowy) i różne sposoby **modelowania obserwacji** w stanach modelu (dyskretne, *pół-ciągłe* lub ciągłe oraz pojedyncze symbole lub wektory symboli). Hierarchiczny model HMM posiada warstwy słów, fonemów i trzy-fonów.

11.3 Przeszukiwanie Viterbiego

Odmianą programowania dynamicznego jest **przeszukiwanie Viterbiego** zaproponowane dla wyznaczania najlepszej ścieżki przejścia w lewo-prawym modelu HMM, przy zadanej sekwencji pomiarowej. Podajemy szczegółowy opis tego algorytmu i wyjaśniamy jego cel w terminach rachunku prawdopodobieństwa.

11.4 Uczenie modelu HMM (*)

Omawiamy algorytmy uczenia elementów macierzy przejść i wyjść (**A** i **B)** modelu HMM: określenie prawdopodobieństw obserwacji cech - **metoda "wprzód-wstecz**"; wyznaczenie prawdopodobieństw przejść pomiędzy stanami - **metoda Bauma-Welcha** (uczenie nienadzorowane) lub **trening Viterbiego** (uczenie nadzorowane).

11.5 Zadania

Należy zdefiniować modele HMM dla transkrypcji różnych słów mówionych i zrealizować przeszukiwanie Viterbiego przy zadanych w zadaniu macierzy prawdopodobieństw występowania pod-fonemów w oknach sygnału.

Literatura uzupełniająca do rozdziału 11

[11.1] A. M. Wiśniewski: *Automatyczne rozpoznawanie mowy bazujące na ukrytych modelach Markowa – problemy i metody*, Biuletyn IAiR nr 12, Wojskowa Akademia Techniczna, Warszawa, 2000.
[11.2] S. Grocholewski: *Statystyczne podstawy systemu ARM dla języka polskiego*. Rozprawy, nr.362, 2001, Politechnika Poznańska.
[11.3] R. Gubrynowicz i inni: *Simplified system for isolated word recognition*. Archives of Acoustics, IFTR PAS, Warszawa, vol. 15(1990), 287-300.
[11.4] L. Rabiner: *A tutorial on hidden Markow models and selected applications in speech recognition.* Proceedings of IEEE, vol. 7 (1989), No. 2, 257-286.
[11.5] A. Wrzoskowicz: *Analiza efektywności niejawnych modeli Markowa w rzeczywistych systemach automatycznego rozpoznawania mowy polskiej*. Rozprawa doktorska, IPPT PAN, Warszawa, 1994.
[11.6] M.J.F. Gales: *Semi-Tied Covariance Matrices for Hidden Markov Models*. IEEE Transactions on Speech and Audio Processing, Vol. 2, May 1999.
[11.7] J. Hamilton: *Analysis of time series subject to changes in regime*. J. Econometrics, vol. 45(1990), 39-70.
[11.8] J.A. Bilmes: *Buried Markov Models for Speech Recognition*. Proc. ICASSP, II, March 1999, 713-716.
[11.9] L. Saul, M. Jordan: *Mixed memory Markov Models: Decomposing complex stochastic processes as mixture of simpler ones*. Machine Learning, 37(1), 1999, 75–87.
[11.10] Z. Ghahramani, M. Jordan: *Factorial hidden Markov models*. Machine Learning, 29, 1997, 245–273.
[11.11] Saul L., Jordan M.: *Boltzmann chains and hidden Markov models.* NIPS-7, 1995. In: G. Tesauro, D. S. Touretzky & T. K. Leen, (Eds.), Advances in Neural Information Processing Systems 7, MIT Press, Cambridge MA. pp.435-442.
[11.12] K.P. Murphy: *Dynamic Bayesian Networks: Representation, Inference and Learning.* Ph.D. Thesis, UC Berkeley, 2002.
[11.13] S. Fine, Y. Singer, N. Tishby: *The hierarchical Hidden Markov Model: Analysis and applications*. Machine Learning, 1998, 32-41.
[11.14] L.E. Baum, T. Petrie, G. Soules, N. Weiss: *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Statistics, 41 (1), 1970, 164-171.