Informatyka, studia dzienne, mgr II st. semestr I
Przetwarzanie obrazu i dźwięku **2011/2012**
Prowadzący: dr inż. Arkadiusz Tomczyk środa, 8:30

Data oddania: _____ Ocena: _____

Paweł Musiał 178726
Łukasz Michalski 178724

Zadanie 4:
Rozpoznawanie izolowanych słów w sygnale mowy.*

Spis treści

1. Cel 2
2. Wprowadzenie 2
2.1. MFCC (*Mel-frequency cepstral coefficients*) 2
2.2. Algorytm DTW (*Dynamic Time Warping*) 2
3. Opis implementacji 4
3.1. MFCC (*Mel-frequency cepstral coefficients*) 4
3.2. Algorytm DTW (*Dynamic Time Warping*) 4
4. Wyniki 5
5. Dyskusja 6
6. Wnioski 7
Literatura 7

* SVN: `https://serce.ics.p.lodz.pl/svn/labs/poid/at_sr0830/lmpm@`

1. Cel

Realizacja zadania polega na stworzeniu aplikacji umożliwiającej obliczanie reprezentacji sygnału audio w postaci ciągów wektorów współczynników MFCC i porównywanie ich za pomocą algorytmu DTW. Należy stworzyć bazę zawierającą przynajmniej 10 różnych słów (przykładowo: „zero", „jeden", ..., „dziewięć") i wykorzystać ją do rozpoznawania słowa wypowiedzianego przez użytkownika. W celu poprawy wyników każde słowo zawarte w bazie może być reprezentowane przez kilka wzorców (np. nagranych przez różne osoby, albo w różnych warunkach akustycznych).

Oprócz ostatecznego wyniku rozpoznania należy zaprezentować wyniki porównań dla wszystkich słów z bazy oraz tablice g (preferowana metoda – w postaci obrazu, reprezentującego wartości $g[i, j]$ za pomocą np. odcieni szarości). Należy rozważyć metodę modyfikacji algorytmu DTW pozwalającą na dopasowanie fragmentu słowa zamiast całości. Należy zaimplementować ograniczenia globalne zgodnie z przydzielonym wariantem, przy czym powinna też istnieć możliwość wyłączenia tych ograniczeń, tak aby ścieżka mogła mieć dowolny kształt.

— Ograniczenie globalne typu Sakoe and Chiba band
— **Ograniczenie globalne typu Itakura parallelogram**

2. Wprowadzenie

Rozpoznawanie mowy jest klasycznym problemem przetwarzania dźwięku, dla którego w ciągu minionych dziesięcioleci zaproponowano wiele rozwiązań. Wyróżniamy tu zasadniczo problem prostszy, polegający na rozpoznawaniu izolowanych słów oraz zadanie rozpoznawania mowy ciągłej. W obu przypadkach należy przyjąć założenia odnośnie sposobu reprezentacji i parametryzacji sygnału mowy oraz odnośnie metod dopasowania danych do wzorca w sposób niezależny od czasu trwania i zmian szybkości analizowanej wypowiedzi.

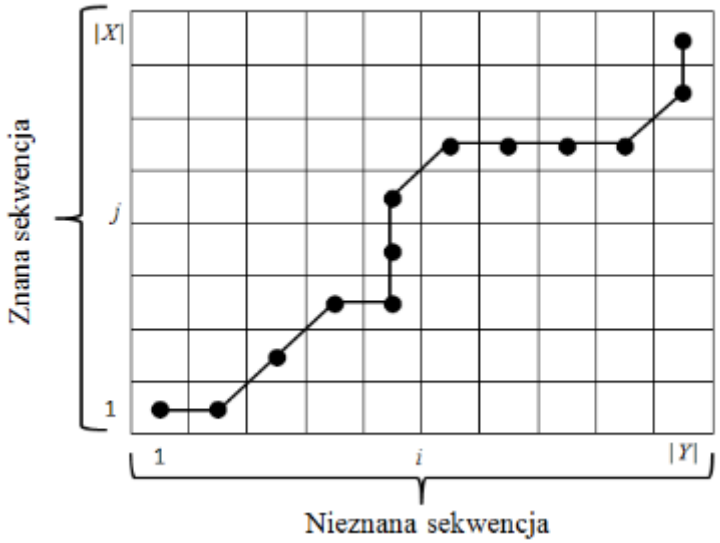
2.1. MFCC (*Mel-frequency cepstral coefficients*)

W trakcie procesu rozpoznawania mowy stosuje się różne metody reprezentowania widma amplitudowego sygnału wejściowego. Często wykorzystuje się tutaj reprezentacje w postaci tzw. cepstrum,i jego ponowne przekształcenie transformatą Fouriera lub cosinusową(„widmo widma"). Z tak powstałych współczynników wybiera się od kilku do kilkunastu (współczynniki cepstralne) , które opisują ogólną informacje o kształcie widma dźwięku. Stanowią one doskonałe narzędzie do późniejszej analizy i porównań dwóch sygnałów. Z uwagi na logarytmiczną zależność między częstotliwością dźwięku, a jego subiektywnie postrzeganą wysokością, dziedzinę częstotliwości, w której rozpatrywane jest widmo dźwięku należy przeskalować zgodnie z perceptualną skalą wysokości dźwięku (skalą melową) przed powtórną transformatą Fouriera lub cosinową. Współczynniki cepstralne uzyskane w ten sposób określamy właśnie jako MFCC.

2.2. Algorytm DTW (*Dynamic Time Warping*)

Metoda Dynamic Time Warping (DTW) jest powszechnie wykorzystywana do porównywaniu ze sobą sygnałów różnej długości. Przeważnie znajduje zastosowanie w systemach rozpoznawania mowy, gdzie różnica w prędkości wypowiadania danej głoski nie wpływa na znaczenie całego słowa.

Metoda DTW pozwala na znalezienie najmniejszej odległości między dwoma szeregami czasowymi przy dopuszczeniu przesunięć w czasie dla obu szeregów. Algorytm ten radzi sobie również w przypadku braku części danych lub ich niedokładności. Najważniejsza jest tutaj kolejność występowania poszczególnych faz szeregu czy sygnału. Dzięki takim właściwościom metoda DTW doskonale nadaje się do rozpoznawania izolowanych słów. Porównywane metodą DTW



Rysunek 1: Ścieżka optymalnego dopasowania wyznaczona algorytmem DTW.

szeregi powinny być podzielone na ramki - momenty w jakich dokonywany był pomiar ich cech charakterystycznych. Obliczana jest bowiem macierz odległości między każdą ramką szeregu wejściowego i bazowego. Stosuje się w tym wypadku metrykę Euklidesową. Następnie trzeba wyliczyć macierz zakumulowanej odległości między próbkami (Rysunek 1). Dla komórki $[i, j]$ sumuję się z jej wartość z najmniejszą wartością jednej z trzech komórek sąsiednich $[i-1,j]$, $[i,j-]$ lub $[i-1,j-1]$. Całą operację zaczyna się od komórki [0,0]. Tak otrzymana macierz pozwala na wyznaczenie ścieżki optymalnego dopasowania do siebie tych dwóch szeregów.

Badając zmiany kierunku ścieżki można określi czy próbki te są do siebie podobne czy też różnią się. Istnieje wtedy nawet możliwość określenia momentu w którym różnica ta jest największa. Aby uznać, że dwie próbki reprezentują ten sam sygnał przyjmuje się kilka warunków jakie musi spełniać otrzymana ścieżka:

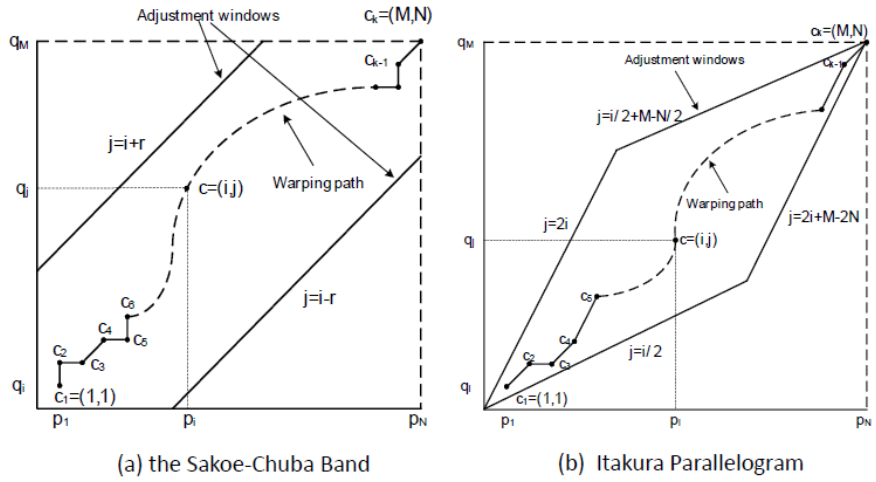
1. Jej przebieg musi być jak najbliższy diagonalnemu.
2. Ścieżka powinna poruszać się jedynie o jedno pole naraz. Eliminuje dziury czy braki dopasowań.
3. Ścieżka nie może zawracać. Jej kolejne współrzędne i, j powinny pozostawać takie same lub rosnąc nigdy maleć.
4. Powinna zaczynać się w punkcie $[0, 0]$ a kończyć $[i, j]$.
5. Ścieżka nie powinna też zbyt szybko lub zbyt wolno rosnąć. Ogranicza to możliwość dopasowania bardzo krótkich sekwencji do bardzo długich.

Dodatkowo stosuje się się jeszcze tak zwane ograniczenia globalne nakładane na przebierz ścieżki optymalnego dopasowania. Najbardziej popularne są :

— Sakoe and Chiba band
— Itakura parallelogram

Wykresy wyznaczonych obszarów dozwolonych przez te ograniczenia przedstawia rysunek 2.

Stosując wszystkie te zasady można ograniczyć ilość możliwych ścieżek jakie da się wyznaczyć. Oznacza to, że wystarczy wtedy mniejsza liczba porównań aby określić czy dane próbki do siebie pasują. Nie spełnienie chociaż jednego z podanych tutaj warunków oznacza bowiem, że w pewnym momencie różnica między szeregami jest za duża. Przyspiesza to same operacje obliczeniowe i skraca czas klasyfikacji próbki wejściowej do tych zapisanych bazie.



Rysunek 2: Obszary dozwolone przez ograniczenia globalne.

3. Opis implementacji

3.1. MFCC (*Mel-frequency cepstral coefficients*)

Implementacja obliczania współczynników MFCC została oparta o algorytm podany w instrukcji do zadania 4b [1]. Wykorzystaliśmy tutaj zewnętrzną bibliotekę do wyliczania współczynników FFT o nazwie jTransforms. Pozwala ona w sposób szybki i pewny wykonać przekształcenie przy pomocy algorytmu FFT. Dla zestawów filtrów w dziedzinie częstotliwości zbudowany w oparciu o skalę melową H_k wartość współczynnika K została dobrana i wynosi 40 natomiast wartość parametru d jest dobierana adaptacyjne na podstawie wzoru 1. Uzyskiwana ilość współczynników MFCC na każdą ramkę wynosi 20.

$$d = \frac{szerokoscOkna_{max} - szerokoscOkna_{min}}{K} \tag{1}$$

,gdzie K to ilość użytych filtrów.

3.2. Algorytm DTW (*Dynamic Time Warping*)

Podobnie jak obliczenia współczynników MFCC algorytm DTW został oparty o implementację opisaną w instrukcji do zadania 4b [1]. Wartym przytoczenia tutaj jest kod reprezentujący obydwa ograniczenia globalne opisane w tym zadaniu:

Itakura parallelogram

```
public boolean check(int i, int j, int I, int J) {
        if (j > 2 * i) {
            return false;
        }
        if (j > i / 2 + I - J / 2) {
            return false;
        }
        if (j < 2 * i + I - 2 * J) {
            return false;
        }
        if (j < i / 2) {
            return false;
        }
        return true;
    }
```

oraz Sakoe and Chiba band

```
1    public boolean check(int i, int j, int I, int J) {
          return Math.abs((i * ((double) I / (double) J)) - j) <= r;
3      }
```

Oba te ograniczenia zostały zaimplementowane w osobnych klasach implementujących wspólnie interfejs:

```
1 public interface IGlobalConstraints {
      public boolean check(int i, int j , int I, int J);
3 }
```

Znaczenie poszczególnych parametrów funkcji *check*:

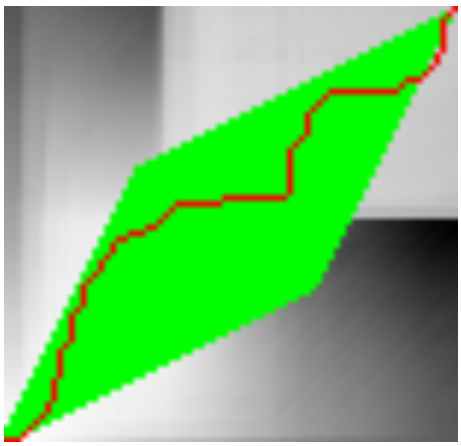
— i,j - współrzędne badanego punktu ścieżki
— I,J - długość sygnału wzorcowego i porównywanego

Dodatkowo w ramach zadania posłużyliśmy się algorytmem do usuwania ciszy przed i po wypowiedzianym słowie. Algorytm ten bazuje na pracy [2] a jego implementacja pochodzi z [3]. Mimo pozytywnych efektów jego niestety czasami pogarsza on rejestrowaną próbkę co utrudnia jej dopasowanie. Powód jego nieprawidłowej pracy mimo prób analizy nie został odkryty. Ze względu na to że pogorszenie występuje stosunkowo rzadko nie ma ono dużego wpływu na pracę całej aplikacji.

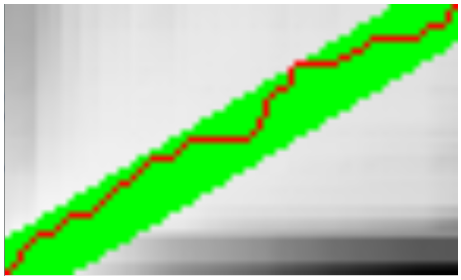
4. Wyniki

Przygotowana przez nas aplikacja ma posłużyć do rozpoznawania 10 izolowanych słów. Zgodnie przykładem w instrukcji wybraliśmy słowa reprezentujące cyfry i liczby poczynając od 1 na 10 kończąc. Próbki tworzące bazę sygnałów wzorcowych pochodzą od pięciu osób. Łącznie w bazie mieliśmy 60 nagrań równo podzielonych po 6 nagrań na każde słowo. Poniżej przedstawiamy kilka różnych prób rozpoznania dla tych słów z różnymi ograniczeniami. Obrazy reprezentują macierz zakumulowanej odległości, której wartości zostały przedstawione w postaci skali szarości. Jaśniejsze obszary reprezentują niższy koszt. Kolorem czerwonym została przedstawiona ścieżka optymalnego dopasowania natomiast kolorem zielonym obszar ograniczony przez przyjęte warunki globalne. Zgodnie z przyjętą w instrukcji do tego zadnia konwencją szerokość obrazu reprezentuje próbkę wzorcową a wysokość nową próbkę wejściową.

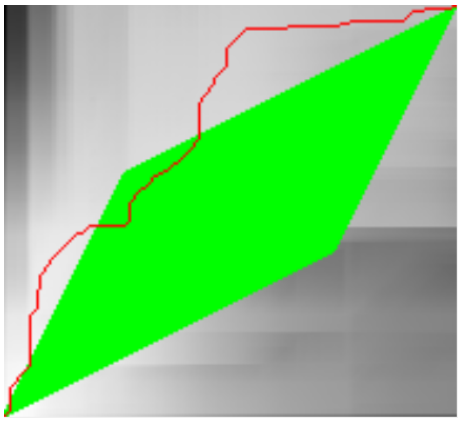
Rysunek 3: Próba dopasowania słowa „jeden”. Ograniczenie jedynie w postaci maksymalnego dystansu dopasowania < 600. Rozpoznano poprawnie.



Rysunek 4: Próba dopasowania słowa „trzy”. Ograniczenie w postaci maksymalnego dystansu dopasowania < 600 oraz Itakura parallelogram. Rozpoznano poprawnie.



Rysunek 5: Próba dopasowania słowa „dwa”. Ograniczenie w postaci maksymalnego dystansu dopasowania < 600 oraz Sakoe and Chiba band. Rozpoznano poprawnie.



Rysunek 6: Próba dopasowania słowa „cztery”. Ograniczenie w postaci maksymalnego dystansu dopasowania < 600 oraz Itakura parallelogram. Nie rozpoznano używając dodatkowo innego wzorca niż dla słowa „cztery”.

5. Dyskusja

W metodzie DTW duże znaczenie ma postać dostarczonych współczynników MFCC. Logarytmowanie podczas obliczania tych współczynników ma na celu uwydatnienie małych różnic, co w efekcie pozwala na lepsze rezultaty w

odróżnianiu poszczególnych dźwięków. Dodatkowo aby odzwierciedlić percepcje ludzką w procesie „dopasowywania" używana jest skala melowa, będąca logarytmiczna skalą odpowiadająca charakterystyce percepcji ludzkiej. Dzięki temu obiektywny wynik metody rozpoznawania słów, teoretycznie powinien lepiej odpowiadać naszemu subiektywnemu odczuciu podobieństwa testowanych słów. Teoretycznie, ponieważ w praktyce zadanie rozpoznawania słów nie jest zadaniem trywialnym. Opracowana przez nas metoda osiąga zadowalające wyniki, jednak jak wspomniano na początku, obliczane współczynniki MFCC mają duży wpływ na rezultat metody. Możliwym wydaje się, zatem lepszy wynik przy lepszym dostosowaniu sposobu obliczania współczynników MFCC.

W przypadku dopasowań przy użyciu algorytmu DTW można zauważyć, że radzi on sobie całkiem nieźle z dopasowaniem dwóch różnych sekwencji audio reprezentujących pojedyncze słowa. Jako, że w naszym zadaniu mieliśmy zaimplementować dodatkowo ograniczenia globalne Itakura parallelogram pokusiliśmy się dodatkowo o zastosowanie i porównanie również Sakoe and Chiba band. W trakcie licznych porównań dało się zaobserwować, że ograniczenia w postaci Sakoe and Chiba band spisują się znacznie lepiej niż Itakura parallelogram. Wynika to głównie z faktu, że optymalne ścieżki często nie mieszczą się w wąskich częściach rąba stanowiącego ograniczenia Itakura parallelogram. Może to wynikać z zaszumienia sygnału nagrywanego z mikrofonu lub też niedokładnego usuwania ciszy przed i po wypowiedzianym słowie.

6. Wnioski

Algorytm DTW wykorzystujący współczynniki MFCC doskonale nadaje się do rozpoznawania dwóch różnych sekwencji różnych w czasie a w szczególności porównywania i dopasowywania izolowanych słów. Ze względu na swoją prostotę w implementacji oraz szybkość działania jest jednym z popularniejszych obok HMM algorytmów wykorzystywanych do tych celów. Jego podstawowa złożoność obliczeniowa wynosząca $O(N^2)$ może być sprowadzona nawet do $O(N)$.

W ramach naszego ćwiczenia udało się nam z powodzeniem wykorzystać do napisania aplikacji służącej do rozpoznawania izolowanych słów. Z wykorzystanych przez nas ograniczeń globalnych lepiej spisywały się te oparte o Sakoe and Chiba band, gdyż są mniej restrykcyjne od Itakura parallelogram jeśli chodzi o kształt ścieżki najlepszego dopasowania na jej początku i końcu. Stosując jednak ograniczenia Itakura parallelogram można poprawić pracę algorytmu przez dostarczenie większej ilości próbek każdego ze słów. Nasza baza była wystarczająca do przeprowadzania tego ćwiczenia ale jej powiększenie o nagrania liczniejszej grupy osób reprezentujących różne sposoby wymowy znacząco może wpłynąć na jej skuteczność.

Literatura

[1] Instrukcja do zadanie 4b `http://ftims.edu.p.lodz.pl/mod/resource/view.php?id=15798`

[2] A New Silence Removal and Endpoint Detection Algorithm for Speech and Speaker Recognition Applications, G. Saha, Sandipan Chakroborty, Suman Senapati

[3] Silence Removal and End Point Detection JAVA Code, Ganesh Tiwari `http://ganeshtiwaridotcomdotnp.blogspot.com/2011/08/silence-removal-and-end-point-detection_29.html`