AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Elektroniki

PRACA DYPLOMOWA
Inżynierska

Temat:

Narzędzie do automatycznego rozpoznawania utworów muzycznych

Software tool for automatic music discovery

Imię i nazwisko: Borys Dubiański
Kierunek studiów: Inżynieria Akustyczna

Opiekun pracy: dr inż. Jakub Gałka

Kraków 2012/2013

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

……..……..

Akademia Górniczo-Hutnicza im. Stanisława Staszica

Wydział Informatyki, Elektroniki i Telekomunikacji

Kierunek: Inżynieria Akustyczna

Borys Dubiański

Praca dyplomowa inżynierska

Narzędzie do automatycznego rozpoznawania utworów muzycznych

Opiekun: dr inż. Jakub Gałka

STRESZCZENIE

Praca dyplomowa prezentuje zrealizowane narzędzie do automatycznego rozpoznawania utworów muzycznych. Celem pracy było stworzenie oprogramowania, które na podstawie krótkiego fragmentu utworu, korzystając z bazy danych, potrafi określi jego tytuł i wykonawcę. Narzędzie zostało zaimplementowane w środowisku MATLAB i posiada interfejs graficzny. System rozpoznawania działa na zasadzie wyszukiwania przez przykład i opiera się na analizie spektrogramu.

SŁOWA KLUCZOWE

rozpoznawanie muzyki, analiza spektrogramu, baza danych

Spis treści

Wstęp

Ludzie nieustannie dążą do wzrostu i wzbogacenia swego zadowolenia z życia. Taka postawa stymuluje niekończące się działania mające na celu poprawę jakości naszego życia. Postęp jest częścią ludzkiej cywilizacji. Z biegiem czasu odmienne dziedziny życia, do których możemy zaliczyć naukę podlegają rozwojowi. Człowiek stara się wykorzystywać ludzkie odkrycia i szuka sposobu na ich użycie w codziennym życiu. Wiedza z różnych dziedzin przenika między sobą w celu znalezienia odpowiedniego zastosowania.

Przykładowo akustyka, która zgodnie z definicją stanowi dział fizyki i techniki, obejmujący zjawiska związane z powstawaniem, propagacją i oddziaływaniem fal akustycznych[1], znajduje zastosowanie w wielu dziedzinach ludzkiego życia. W medycynie korzysta się z ultrasonografu(USG), który jest nieinwazyjną metodą diagnostyczną, wykorzystującą zjawiska rozchodzenia się, rozpraszania oraz odbicia fali ultradźwiękowej. W żegludze funkcjonuje echosonda, urządzenie do pomiaru głębokości wody oraz odległości od unoszących się w niej ciał stałych na podstawie obliczeń czasu powrotu dźwięku odbitego od dna, okrętu podwodnego, czy innej przeszkody.

Wraz postępującym rozwojem technologii cyfrowej, rozwinęło się cyfrowe przetwarzanie dźwięków, którego efekty są widoczne między innym w branży muzycznej. Osiągnięcia w tej dziedzinie umożliwiły wytwarzanie efektów specjalnych (np. chorus, flanger), poprawy jakości sygnału, dostosowywanie parametrów sygnału do określonych zastosowań oraz kompresji sygnałów. Przetwarzanie sygnałów cyfrowych obejmuje również inne zagadnienia i przyczyniło się do powstania systemów rozpoznawania mowy.

Kto doświadcza użycia systemu służącego do automatycznego rozpoznawania utworów muzycznych jest zafascynowany skutecznością tego programu i jednocześnie ciekawy zasad działania tego programu. Dlatego tematem niniejszej pracy inżynierskiej jest stworzenie narzędzia do automatycznego rozpoznawania utworów muzycznych, w celu poznania tajemnicy działania tego „magicznego" urządzenia.

---

[1] Wikipedia.org, *Akustyka* [online] Dostępny: http://pl.wikipedia.org/wiki/Akustyka. (odwiedzona 05.01.2012).

Temat jest interesujący ze względu na wielowymiarowość zadania. Po pierwsze, stanowi on zagadnienie z zakresu przetwarzania sygnałów cyfrowych. Ponadto, programowanie jest twórczym procesem, gdyż jest to język wyrazu ludzkiej kreatywności w postaci różnorodnych aplikacji i programów. Dodatkowo, zachęcająca jest możliwość pracy na utworach muzycznych, a więc obcowanie z muzyką, która sprawia przyjemność. Wreszcie, dane narzędzie znajduje zastosowanie w codziennym życiu, a fakt substytucji umiejętności ludzkiej, jaką jest funkcja rozpoznawania utworów muzycznych jest fascynujący.

Zatem celem niniejszej pracy jest stworzenie narzędzia do automatycznego rozpoznawania utworów. Zrealizowany program powinien umożliwiać rozpoznanie utworu na podstawie odsłuchu krótkiego fragmentu muzyki. Użytkownik powinien uzyskać informacje na temat tytułu danego utworu oraz nazwy artysty go wykonywującego. Sygnałem wejściowym powinien być sygnał zarejestrowany przy pomocy mikrofonu. Obsługa programu powinna być możliwa poprzez interfejs graficzny. Jako środowisko pracy wybrany został MATLAB'a, który jest często wykorzystywanym systemem w procesie przetwarzania sygnałów cyfrowych.

W celu poprawnego wykonania zadania przeanalizowano literaturę z zakresu systemów rozpoznawania utworów muzycznych. Dominująca cześć przeczytanych materiałów to publikacje naukowe pracowników z wielu ośrodków uniwersyteckich, opisujących różne metody rozpoznawania utworów muzycznych. Wybrane zagadnienia zostały przedstawione w poniższej pracy. Do opracowania narzędzia do automatycznego rozpoznawania utworów pomocne okazały się również publikacje różnych funkcji zaimplementowanych w środowisku MATLAB przez osoby zajmujące się przetwarzaniem sygnałów cyfrowych.

Poniższa praca poza wstępem i wnioskami, posiada 4 rozdziały. Pierwszy rozdział stanowi wprowadzenie do tematyki systemów rozpoznawania muzyki. Drugi rozdział przedstawia wybrane zagadnienia związane z systemami rozpoznawania utworów muzycznych, trzeci rozdział przedstawia opis implementacji narzędzia do automatycznego rozpoznawania utworów w środowisku MATLAB, natomiast czwarty rozdział zawiera informacje na temat skuteczności działania narzędzia.

1. Systemy rozpoznawania utworów muzycznych

1.1. Wprowadzenie

Na przestrzeni ostatnich lat znacząco rozwinęła się cyfrowa postać utworów muzycznych, wymieniając jako przykład format MP3, który przyczynił się do zwiększenia udziału muzyki cyfrowej w procesie konsumpcji muzyki. Obecnie, konsument ma możliwość wyszukania utworu w Internecie i jego zakup bez konieczności wychodzenia z domu. Ponadto, konsument jest w stanie przy pomocy odpowiedniego oprogramowania, który może być obsługiwany przez telefon komórkowy, rozpoznać nieznany utwór, który słyszy w telewizji, radiu, centrum handlowym, czy podczas wizyty w lokalu, w którym właśnie pije kawę. Dodatkowo, tego typu aplikacje często umożliwiają wyświetlenie tekstu danej piosenki oraz wyświetlają odnośniki do serwisów, w których można zakupić dany utwór. Do rozpoznania utworu wystarczy krótki fragment utworu, około 10 sekund[2], który może być zarejestrowany przy pomocy mikrofonu zainstalowanego w telefonie. Systemy do rozpoznawania utworów znajdują również zastosowanie w dziedzinie wykrywania naruszeń praw autorskich, serwis YouTube korzysta z tej funkcji tej technologii.

Jak działają takie systemy? W dalszej części niniejszego rozdziału zostaną przedstawione informacje na temat systemów rozpoznawania utworów. Omówiony zostanie sposób ich działania i różne sposoby ich realizacji.

1.2. Systemy rozpoznawania muzyki

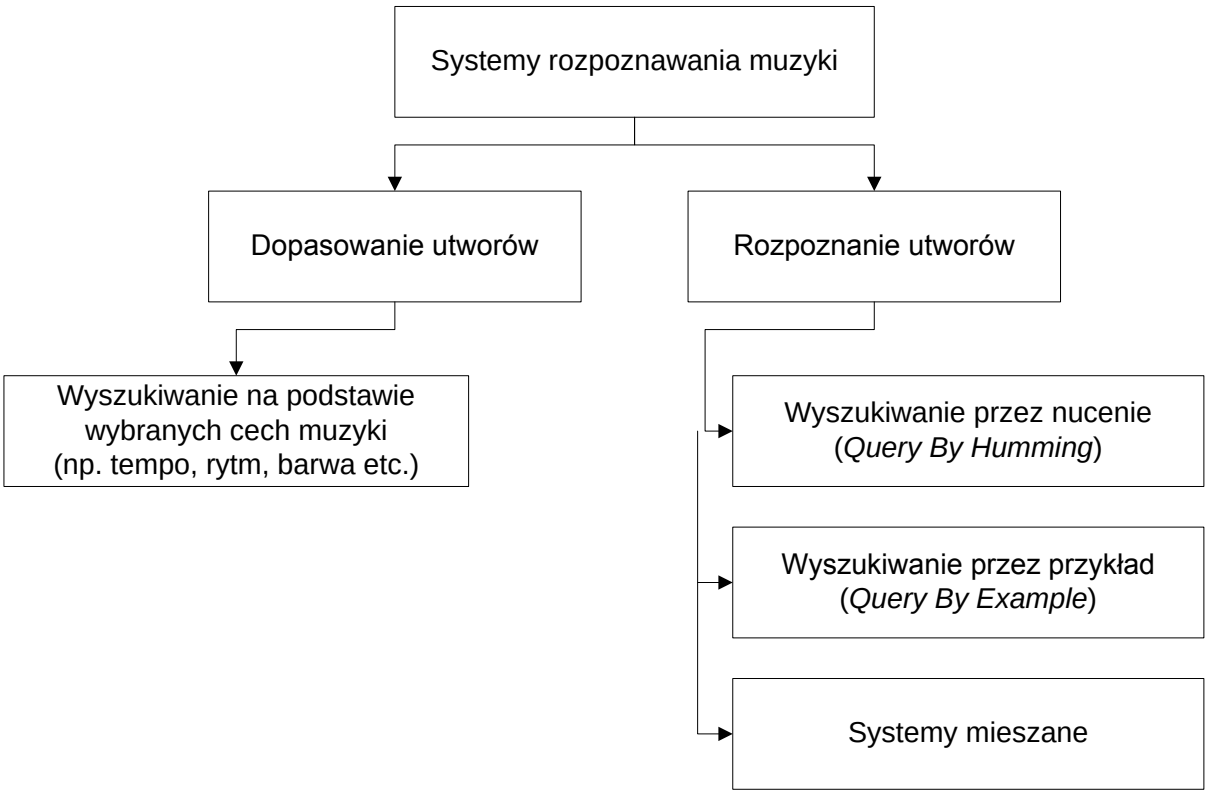
Systemy rozpoznawania muzyki można zasadniczo podzielić na dwie kategorie:

a) dopasowanie utworów (znalezienie utworów podobnych do danej próbki)
b) rozpoznanie utworów (dokładne określenie nazwy analizowanej próbki)

Podstawowym celem dopasowania utworów jest generowanie listy utworów podobnych do analizowanej próbki. Podobieństwo jest określane na podstawie takich parametrów jak melodia, tempo, rytm, barwa etc. W przypadku systemów rozpoznawania utworów wyszukiwanie może odbywać się na podstawie zapytania przez zaśpiewanie fragmentu melodii (*Query by Humming* [QBH]*)* lub zapytania przez

[2] *Czas próbkowania w serwisach Shazam, Soundhound wynosi około 10 sekund*

przykład (*Query by Example* [QBE]*)*. W metodzie QBH na podstawie zanuconego fragmentu utworu, następuje proces śledzenia wysokości dźwięków, tworzony jest kontur melodyczny, na podstawie którego zostaje dopasowany kontur odpowiedniego utworu z bazy. Wyszukiwanie QBE sprowadza się do tworzenia sygnatury dźwięku tzw. „akustycznego odcisku palca" (*z ang. audio fingerprint*) i porównanie go z „odciskami zgromadzonymi w bazie. Istnieją systemy, które wykorzystują elementy obu metod. W niektórych systemach wykorzystuje się rozpoznawanie tekstu piosenki (Wang, 2010).



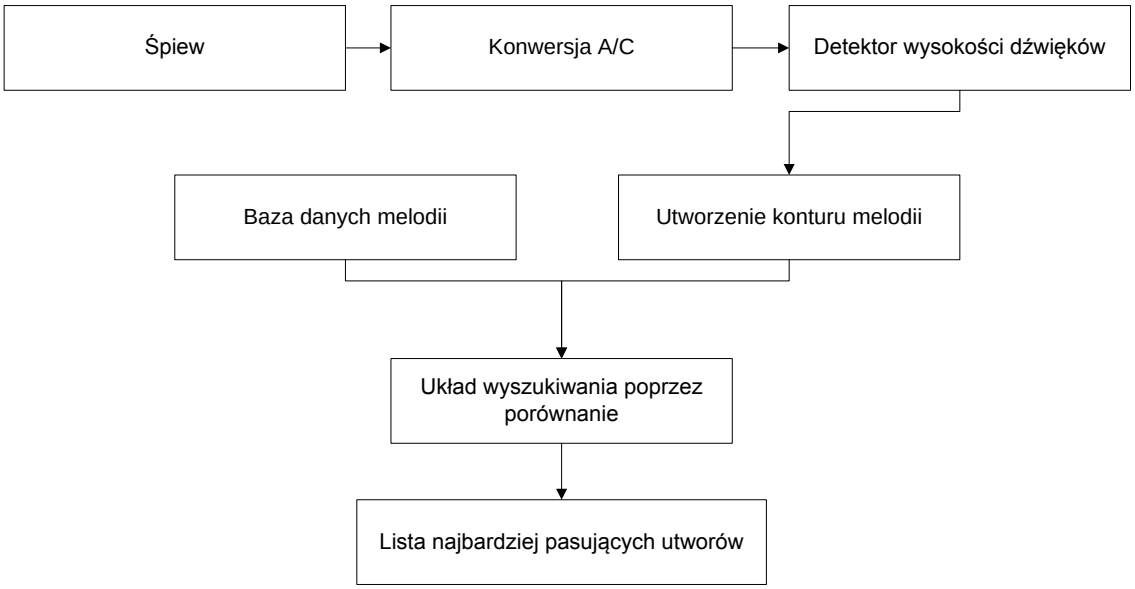
Rysunek 1.Ogólna klasyfikacja systemów rozpoznawania muzyki. Opracowanie własne na podstawie: Ja-Hwung Su, Cheng-Wei Wu, Shao-Yu Fu, Yu-Feng Lin, Wei-Yi Chang, I-Bin Liao, Kuo-Wei Chang Vincent. S. Tseng., *Empirical Analysis of Content-based Music Retrieval for Music Identification* , In Proc. of the15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems *(KES),* Kaiserslautern , Germany , September 12-14, 2011.

1.3. Systemy rozpoznawania utworów

System rozpoznawania utworów stanowi przedmiot tej pracy, dlatego zostaną przedstawione bliższe informacje na ich temat.

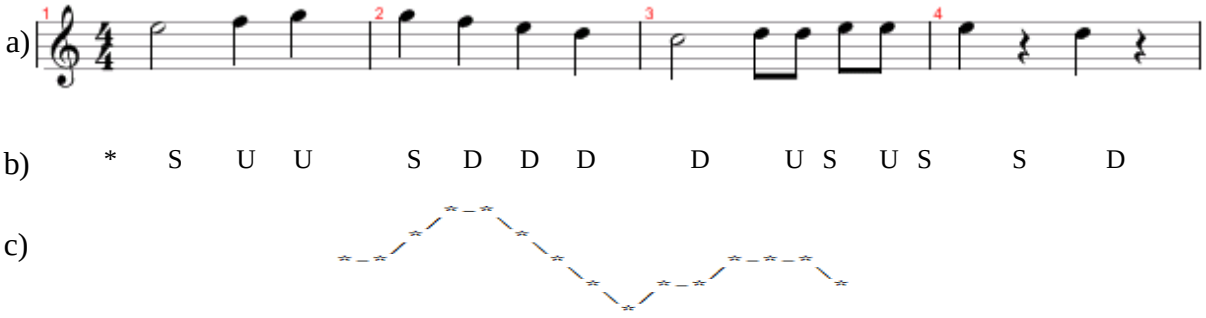
1.3.1. Wyszukiwanie przez nucenie

Proces rozpoznania rozpoczyna się poprzez rejestracje za pomocą mikrofonu śpiewanej melodii lub gwizdanej przez użytkownika. W następnej kolejności algorytm śledzenia wysokości dźwięku zamienia melodie na ciąg znaków informujący o zmianach wysokości dźwięków tzw. kontur melodyczny.



Rysunek 2.Schemat systemu wyszukiwanie przez nucenie (QBH). Opracowanie własne na podstawie: A. Ghias, J. Logan, D. Chamberlin, B. C. Smith. *Query by humming - musical information retrieval in an audio database*, In ACM Multimedia 95, 1995

W końcowej fazie kontur melodyczny jest porównywany z konturami znajdującymi się w bazie utworów w celu znalezienia najbardziej podobnych obiektów. Na rysunku nr 2 został przedstawiony schemat przykładowego sytemu rozpoznawania przez nucenie(Ghias, 1995). Baza danych systemu QBH to sparametryzowany zbiór nagrań, przechowujący informacje o głównej linii melodycznej. W większości przypadków zbiór bazy jest tworzony na podstawie plików MIDI, które zawierają informacje o zapisie nutowym. Kontur melodyczny jest zazwyczaj zapisywany przy pomocy kodu Pearsonsa, który dostarcza informacji o wysokości dźwięku względem nuty poprzedniej. To zapisu wymagany jest 3-elementowy zbiór symboli (U – wyższa, D –niższa, S – taka sama). Kod Parsonsa jest nieczuły na drobne zafałszowania, czy błędy rytmiczne, wynikające z różnego czasu trwania nut.



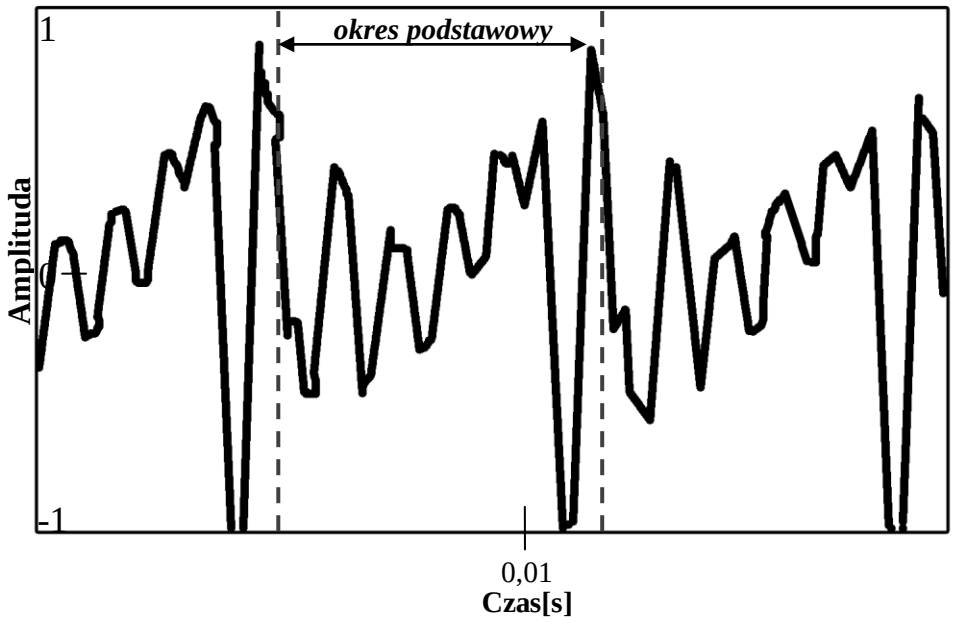
Rysunek 3. Ilustracja kodu Parsonsa. a) zapis nutowy, b) kod Parsonsa c) ilustracja konturu

Wyszukiwanie polega na znajdywaniu wystąpień wzorca w ciągach tekstowych. Algorytmy wyszukiwania mogą działać w oparciu o obliczanie odległości ciągów, czy struktury drzew binarnych. Badania Ghiasa(1995) wskazują, że do rozróżnienia 90% utworów w bazie wystarczy sekwencja 10-12 nut. Czas wyszukiwania jest liniowo zależny od wielkości bazy danych.

Detekcja wysokości dźwięków może odbywać poprzez różne algorytmy, wymienić można:

- autokorelację,
- liczenie przejść przez zero,
- metody statystyczne (HMM), itp.

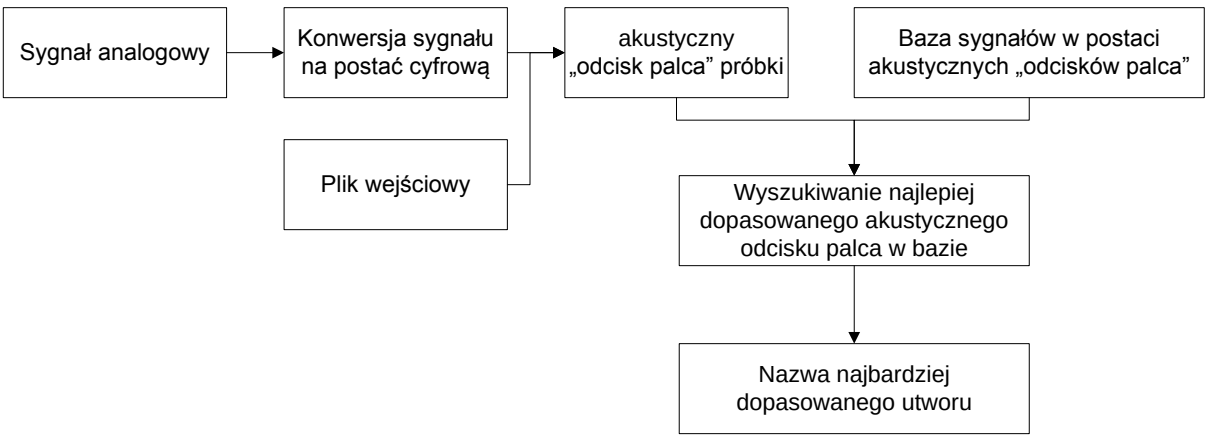
Przykładem detekcji wysokości jest również analiza sygnału w dziedzinie czasu. Na wejściu sygnał jest przetwarzany przez filtr dolnoprzepustowy, ograniczający pasmo do częstotliwości 1 kHz. Sygnał jest analizowany w ramkach o długości 20 ms, co oznacza, że najniższa możliwa częstotliwość do wykrycia to 50 Hz. Detektor wyznacza w przetworzonym sygnale powtarzający się okres podstawowy.



Rysunek 4. Wykrywanie wysokości dźwięku na podstawie okresu podstawowego

1.3.2. Wyszukiwanie przez przykład

Druga grupa systemów rozpoznawania utworów to systemy wyszukiwania przez przykład. Sygnałem wejściowym może być plik dźwiękowy (WAV, MP3 etc.) lub sygnał analogowy (np. ścieżka dźwiękowa telewizji, radio, centrum handlowym, etc.). W przypadku sygnału analogowego konieczna jest rejestracja strumienia audio przy pomocy mikrofonu i jego zamiana na postać cyfrową. Odpowiedni algorytm generuje akustyczny odcisk palca danej próbki, a następnie wyszukuje w bazie najbardziej podobnego obiektu. Następuje porównanie akustycznych odcisków i informacją wyjściową jest nazwa utworu o najbardziej dopasowanym odcisku.



Rysunek 5. Schemat systemu wyszukiwanie przez przykład (QBE).

Technologia akustycznego odcisku palca jest dzisiaj powszechnie używana, wiele komercyjnych serwisów rozpoznawania utworów wykorzystuje algorytm tego rodzaju. Niektóre systemy rozpoznawania utworów poprzez telefony komórkowe, działają w oparciu o tą technologie. Odpowiednia aplikacja przesyła próbkę utworu w postaci pliku audio albo sama dokonuje ekstrakcji odpowiednich cech i przesyła sparametryzowany sygnał w postaci mniejszego pliku, nazywanego sygnaturą pliku audio, do serwera centralnego, odpowiedzialnego za wyszukanie i dopasowanie odpowiedniego utworu z bazy.[3] Następnie serwer wysyła do aplikacji informacje na temat dopasowanej sygnatury, takie jak tytuł piosenki, czy nazwa wykonawcy.

1.4. Wyzwania systemów rozpoznawania utworów

Hałas.

Często analizowana próbka sygnału musi konkurować z występującym hałasem. W centrach handlowych, czy kawiarniach muzyka zazwyczaj leci w tle i jest zagłuszana przez hałas generowany przez rozmawiających ludzi. Często poziom hałasu może być mocniejszy niż próbkowany sygnał. Algorytm rozpoznawania powinien być odporny na działanie znaczącego hałasu.

Zniekształcenia.

System powinien być odporny na zakłócenia wynikające z różnych powodów, przykładowo takie jak niedoskonałości sprzętu grającego, czy warunki otoczenia uwzględniające pogłos i pochłanianie dźwięków. Dodatkowo w przypadku wykorzystania telefonów komórkowych system powinien uwzględniać ograniczenia telefonów, które obsługują pasmo częstotliwościowe w zakresie 300 – 3400 Hz i posiadają również różne algorytmy które mogą zniekształcać sygnał, takiej jak systemy wzmacniania sygnału głosu, nieliniowe tłumienie hałasu, czy kodowanie danych (MP3, GSM, itp.)

Zarządzanie bazą.

System powinien zawierać zbiór sygnatur milionów piosenek. Reprezentacja poszczególnych utworów musi być stosunkowo mała i wymagać jak najmniejszej ilości pamięci w celu jak najszybszej analizy danych. Powinna być możliwa obsługa kilku zapytań jednocześnie. Powiększanie bazy powinno nieznacznie z zmniejszać prawdopodobieństwo poprawnego rozpoznania.

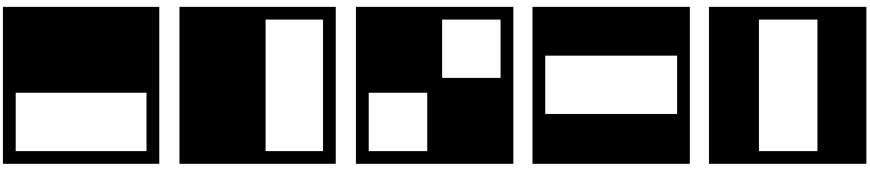
[3] Wang, A.: *The shazam music recognition service*. Commun. ACM 49(8), 44–48 (2006)

1.5. Przykłady systemów rozpoznawania utworów

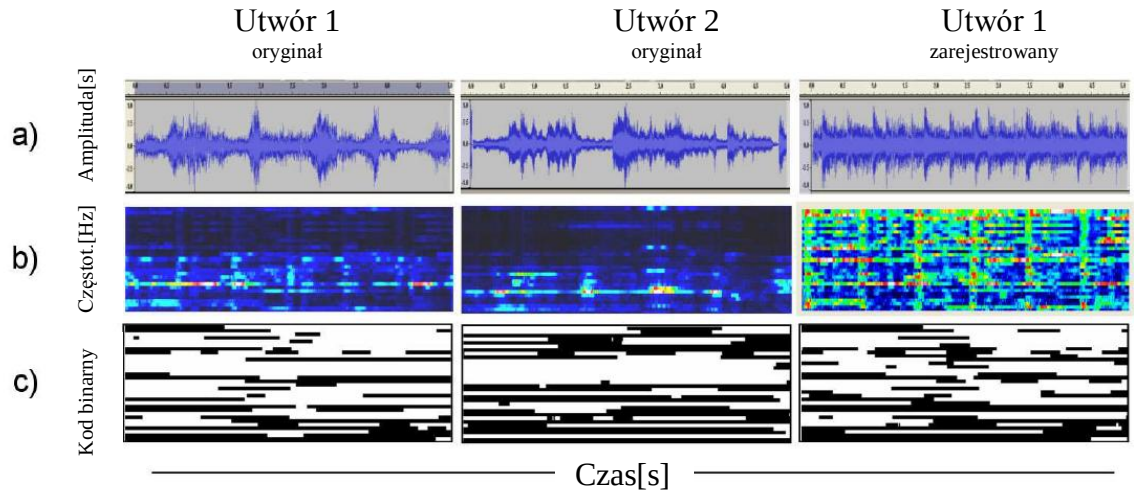
Istnieje różne podejścia do problemu rozpoznawania utworów, niektórzy wykorzystują ukryte modele Markowa, cześć osób wzoruje się na systemach rozpoznawania mowy i wykorzystuje współczynniki cepstralne w dziedzinie częstotliwości w skali melowej (z ang. *Mel-frequency cepstral coefficients* – MFCC), jeszcze inni starają się analizować utwory jak obrazy i wykosztują spektrogramy. Poniżej zostały przedstawione trzy przykłady różnych podejść do problemu rozpoznawania utworów. Wszystkie opierają się na analizie spektrogramu, które jest obecnie najbardziej powszechnym podejście.

Ke, Hoiem, Sukthankar

Punktem wyjścia podejścia Ke, Hoiem, Sukthankar(2005) jest przetworzenie 1-wymiarowego(1-D) sygnału audio do postaci 2-wymiarowej(2-D) w formie spektrogramu, którą obliczany dzięki krótkookresowej transformaty Fouriera (STFT) i reprezentuje poziom energii z 33 logarytmicznie wyskalowanych pasm w zakresie 300 – 2000 Hz. Okno czasowe ma długość 0,372s. Podstawą podejścia autorów jest potraktowanie spektrogramu jak obrazu, wykorzystuję się filtry kwadratowe, które uśredniają obszar spektrogramu, podobnie jak w systemach detekcji twarzy. Uśrednianie zachodzi poprzez wykorzystanie odpowiednich filtrów, które zostały przedstawione na rysunku nr 6. Sygnaturą obrazu jest binarna macierz, przyjmująca wartości ‘0’ lub ‘1’. Na rysunku nr 7 zilustrowano omawiany sposób sygnatury na podstawie 10 sekundowych próbek sygnałów. Można dostrzec, że sygnatura utworu nr 2, zarejestrowana za pomocą mikrofonu, razem z zakłóceniami, jest bardziej podobna to sygnatury oryginalnego utworu nr 1. Dopasowanie na poziomie komputera jest obliczane na podstawie odległości Hamminga.



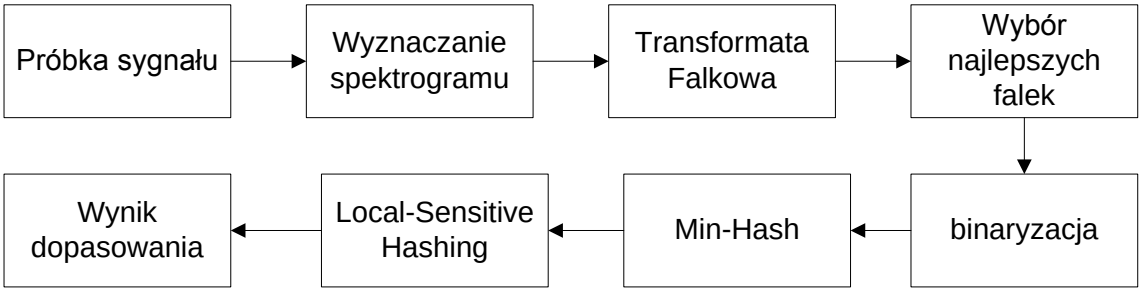
Rysunek 6. Zestaw filtrów kwadratowych uśredniających spektrogram. Źródło: Yan Ke, Derek Hoiem, Rahul Sukthankar: *Computer Vision for Music Identification*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.



Rysunek 7. Reprezentacja audio 3 różnych sygnałów (10 s) metodzie Ke, Hoiem and Sukthankar. a) sygnał audio w dziedzinie czasu b) spektrogram c) sygnatura. Źródło: Yan Ke, Derek Hoiem, Rahul Sukthankar: *Computer Vision for Music Identification*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

Baluja, Covell

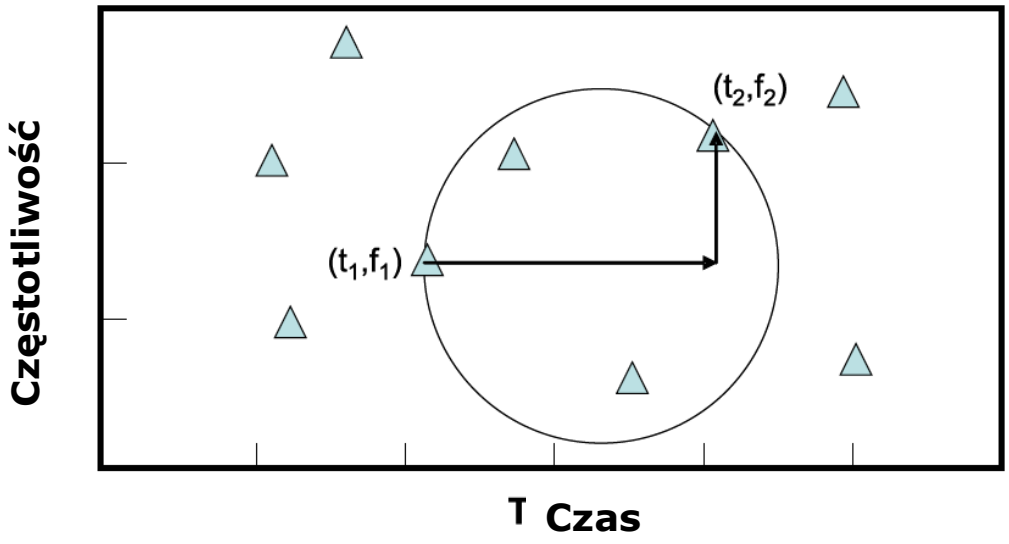
W swojej pracy Baluja i Covell(2006), podobnie jak w podejściu zespołu Ke, Hoiem, Sukthankar, inspirują się systemami przetwarzania obrazów. Najpierw, tak samo ja w podejściu Ke, wyznaczane są spektrogramy dla nachodzących na siebie okien czasowych i traktowane są jak obraz. Następnie podlegają dekompozycji przy użyciu falek Haara. System wybiera zestaw falek, które najlepiej charakteryzują spektrogram. Następnie obraz jest konwertowany do postaci binarnej i dzielony przy pomocy algorytmu Min-Hash na krótsze ciągi. W celu dopasowania najbliższego sąsiada wykorzystano technikę Locality-Sensitive Hashing (LSH).
Na rysunku nr 8 przedstawiono schemat działania systemu zaproponowanego przez Baluja *i* Covell'a.



Rysunek 8. Schemat działania systemu rozpoznawania Baluja,Covell. Źródło: S. Baluja and M. Covell. *Content fingerprinting using wavelets*. 3rd European Conference on Visual Media Production, pp. 198-207, Nov. 2006.

Wang

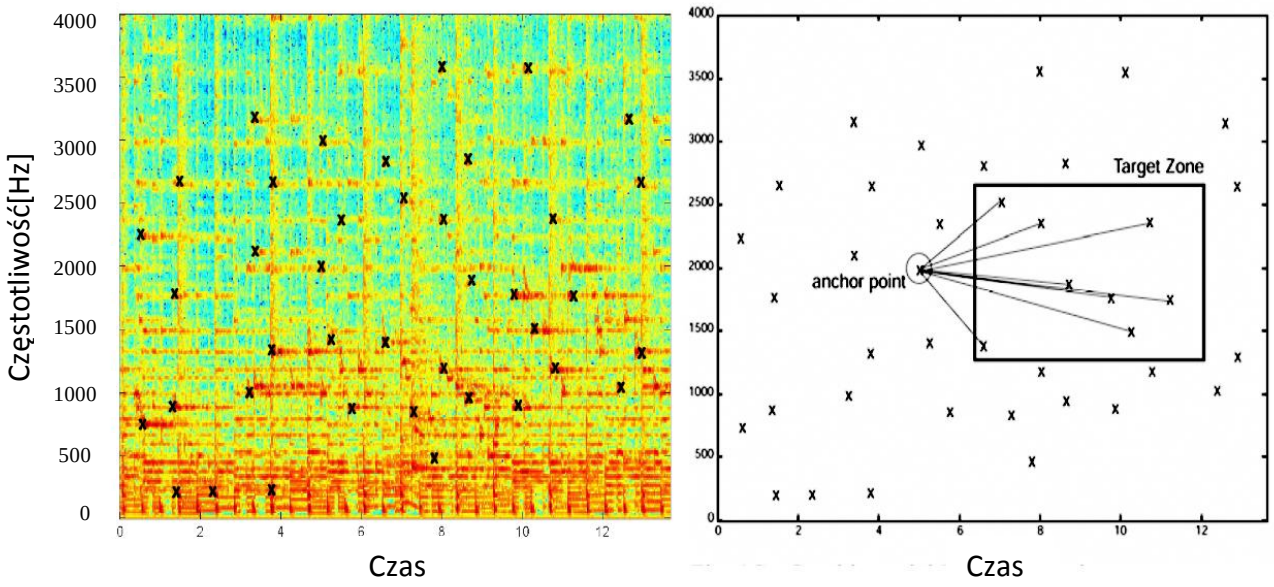
W poprzednich podejściach obliczanie sygnatury wymagają przetwarzania dużych obrazów, w postaci całych spektrogramów. Wang(2003) koncentruje się na analizie szczytów spektrogramów. Argumentuje użycie szczytów dwoma stwierdzeniami. Po pierwsze, bardziej prawdopodobne jest, że szczyty spektrogramu nie zostaną zakłócone przez hałas otoczenia, niż inne punkty. Po drugie, szczyty spektrogramu, spełniają właściwości superpozycji, tzn. szczyty muzyki i hałasu analizowane razem, są tożsame z suma szczytów spektrogramów muzyki i hałasu anglizowanych osobno.



Rysunek 9. Sygnatura sygnału audio według metody Wang'a. Źródło: Chandrasekhar, V.; Sharifi, M. & Ross, D. A., *Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications*. 2005

Na rysunku nr 9 na przykładzie dwóch szczytów, pokazano jak obliczana jest sygnatura. Znaleziony szczyt jest parowany z innym, leżącym w określonym sąsiedztwie. Razem dwa szczyty są opisane zestawem 3 danych: $[(t_2 - t_1), f_1, (f_2 - f_1)]$,

konwertowane do postaci liczbowej i tworzą znacznik, który jest przechowywany w bazie danych. Na rysunku nr 10 znajduje się ilustracja procesu detekcji lokalnych maksimów spektrogramu i wyznaczania ich par, tworzących znacznik. Sygnaturą utworu jest zbiór znaczników, przechowywany w bazie. W procesie wyszukiwania znaczniki zarejestrowanej próbki są porównywane ze znacznikami znajdującymi się bazie danych. Utwór w bazie, który posiada największa liczbę takich samych znaczników jak analizowana próbka stanowi wynik wyszukiwania.



Rysunek 10. Znakowanie szczytów spektrogramu sygnału akustycznego: a) spektrogram z zaznaczonymi szczytami b) zbiór szczytów stanowiących jeden znacznik. Źródło: A.Wang. *An industrial-strength audio search algorithm.* In Proc. Of International Conference on Music Information Retrieval (ISMIR), 2003.

2. Wybrane zagadnienia rozpoznawania utworów muzycznych

2.1. Kontener danych ID3

W związku z pojawieniem się formatu MP3, z czasem pojawił się problem z przechowywaniem informacji na temat danego pliku audio, takich jak tytuł utworu, nazwa wykonawcy, albumu etc. W 1996 roku, Eric Kemp przedstawił ideę dołączenia nagłówka danych do pliku audio, który mógłby rozwiązać problem i przechowywać potrzebne dane. Taka powstała metoda jest znana dzisiaj jako ID3, która stała się standardem przechowywania informacji o plikach MP3. Nazwa ID3 wywodzi się od wyrażenia „**ID**entify MP**3**". Pierwsza implementacja tej metody oznaczana dzisiaj ID3v1, lokowała dodatkowe dane na końcu pliku audio. Dodatkowy pakiet miał stały rozmiar 128 bajtów, aby łatwo można było go zlokalizować. Podział bajtów został przestawiony w tabeli nr 1, można zauważyć, że suma wszystkich elementów daje 125 bajtów, brakujące 3 bajty służą do oznaczenia kontenera danych na początku wyrażeniem 'TAG', które identyfikuje znacznik ID3. W celu znalezienia znacznika należy przeszukać 128 ostatnich bajtów pliku, czy występuje słowo 'TAG'. Gatunek muzyki był określany na podstawie zdefiniowanej listy z gatunkami, Kemp zdefiniował 80 gatunków muzyki które byłe oznaczone numerami od 0 do 79, znacznik przechował indeks odpowiadający danemu gatunkowi. [4]

Tabela 1.Konstrukacja kontenera danych ID3v1

Tytuł utworu	30 znaków
Artysta	30 znaków
Album	30 znaków
Rok wydania	4 znaki
Komentarz	30 znaków
Gatunek	1 bajt

Źródło:ID3.org, *What is ID3(v1)?[*online] Dostępny w Internecie: http://id3.org/ID3v1 (odwiedzona 05.01.2012)

[4] ID3.org, *What is ID3(v1)?* [online] Dostępny w Internecie: http://id3.org/ID3v1 (odwiedzona 05.01.2012)

W miarę postępu czasu ID3 się rozwijał, najpierw rozszerzono kontener do 227 bajtów, aby umożliwić zapis dłuższych łańcuchów (ID3v1.1). W 1998 roku powstał kontener danych kolejnego standardu ID3v2, które oferuje dużo większe możliwości. Znacznik ID3v2 jest zapisywany na początku pliku, dzięki czemu możliwe jest przechowanie przez pliki MP3 obu standardów. Tag ID3v2 jest zapisywany w postaci paczek danych nazywanych ramkami. Ramki mogą przechowywać każdy rodzaj informacji, między innymi: tytuł, nazwę albumu, wykonawcy etc., ale także tekst piosenki, zdjęcia okładek, linki do stron internetowych. Każda z ramek może maksymalnie pomieścić 16 MB, a cały znacznik nie może przekroczyć 256 MB, co w porównaniu do standardu ID3v1 daje zupełnie nowe możliwości.[5]

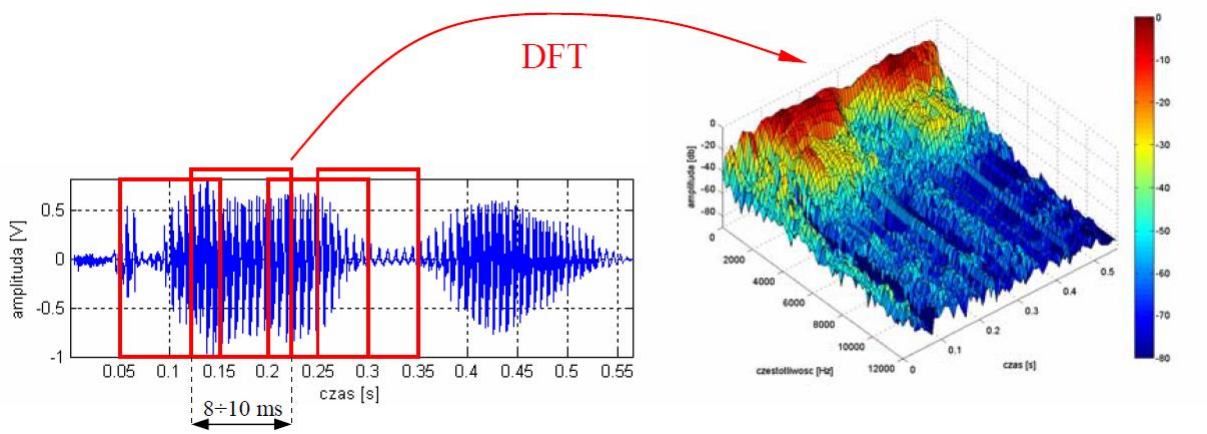
2.2. Spektrogram

W dziedzinie rozpoznawania utworów, ważna rolę pełni spektrogram, czyli wykres widma amplitudowego sygnału w dziedzinie czasu. Można powiedzieć, że jest to wizualizacja analizy czasowo-częstotliwościowej sygnału akustycznego. Wyznacza się go, dzieląc sygnał na krótsze okresy, dla których obliczane są amplitudy składowych harmonicznych. Argumentami funkcji są częstotliwość i czas. W praktyce, najczęściej pozioma oś oznacza czas, pionowa częstotliwość, a wartość amplitudy określa jasność lub kolor danego punktu. Spektrogram jest związany z krótkoczasową transformatą Fouriera(z ang. *Short Time Fourier transform* – STFT), gdyż jest kwadratem jej modułu:

$$spektrogram(t,f) = |STFT(t,f)|^2 \qquad (1)$$

$$t - czas;\ f - częstotliwość;\ STFT - krótkoczasowa\ transformata\ Fourier$$

[5] Wikipedia.org, *ID3* [online] Dostępny w Internecie: http://pl.wikipedia.org/wiki/ID3 (odwiedzona 05.01.2012)



Rysunek 11. Krótkoczasowa transformata Fouriera(STFT). Źródło: Kłaczyński M. *Zjawiska wibroakustyczne w kanale głosowym człowieka.* AGH, Kraków, 2007

STFT pozwala przedstawić zmienność sygnału akustycznego w czasie jako widmo czasowo-częstotliwościowe. Punktem wyjścia w cyfrowej analizie widmowej sygnałów jest dyskretna transformata Fouriera(z ang. *Discrete Fourier Transform* – DFT). STFT w praktyce to operacja DFT dla kolejnych okien czasowych, co ilustruje rysunek nr 11. Szerokość okna zależy od potrzeb, dla analizy sygnału mowy przyjmuje się mniejsze okna, nie większe niż 10 ms, a w przypadku analizy utworów muzycznych, szerokość okna przekracza 50 ms. Poniższy wzór opisuje STFT w postaci dyskretnej, dla przebiegu czasowego x[i]:

$$STFT\{x[i]\}(n,k) = X(n,k) = \sum_{i=n*s+\frac{L}{2}-1}^{n*s+\frac{L}{2}-1} x_n[i]w[i-n*s]e^{-j2\pi ki/N} \qquad (2)$$

$$x_n[i] - n`ty\ segment\ sygnału; w[i] - funkcja\ okna\ śledzącego;$$

$$L - długość\ okna;\ s - długość\ kroku\ przesuwającego\ okno$$

2.3. Tablica mieszająca (tablica z haszowaniem)

Niezbędnym elementem systemów rozpoznawania utworów jest baza danych, która powinna umożliwiać szybki dostęp do informacji. Tablica mieszająca nazywana także tablicą z haszowaniem jest typem struktury danych, która umożliwia szybkie porównywanie danych, dlatego znajduje zastosowanie systemach, które wymagają szybkiego porównywania danych. Zasadne zatem jest ich użycie w systemach rozpoznawania utworów muzycznych.

Pozycja elementu w tablicy z haszowaniem jest wyliczana na podstawie wartości elementu lub tzw. klucza, związanego z elementem. Funkcję wyliczającą

pozycję elementu w zbiorze nazywa się funkcją mieszającą lub haszującą (ang. *hash function*). Szczególną cechą metod haszowania jest, że czas wyszukiwania jest niezależny od liczby elementów w zbiorze, wiąże się tylko z rozmiarem bazy danych

Działanie wyszukiwania w tablicy zostanie opisane na następującym przykładzie: [6]

- Funkcja *numer()* przypisuje wartości liczbowe kolejnym znakom alfabetu od 1 do 26, czyli *numer(a)=1, numer(b)=2, etc.*
- Funkcja *znacznik()*, która zamienia słowa na wartość w zbiorze liczb naturalnych jako sumę wartości przypisanym literom modulo 13, dla dowolnego słowa $w=\{e_1,e_2,..,e_n\}$.
  $znacznik(w) = (numer(e_1) + numer(e_2) + ... + numer(e_n)) \bmod 13.$
- Analizowany jest zbiór słów *X*
  X = {antek, piotr, olek, asia, adam, basia, ola, ina}
- Na podsawie funkcji są obliczane znaczniki(hasze) słów:
  znacznik(antek) = (1+15+20+5+11) mod 13 = 52 mod 13 = 0 ,
  znacznik(piotr) = (17+9+16+20+18) mod 13 = 80 mod 13 = 2,
  znacznik(ola) = (16+12+1) mod 13 = 3, itd.
- Wartości funkcji znacznik dla słowa *w* jest adresem danego słowa w tablicy mieszającej TAB

Tabela 2. Tablica mieszająca dla przykładu

	0	1	2	3	4	5	6	7	8	9	10	11	12
TAB:	antek		piotr	ola	asia	olek	basia	adam		aga			ina

Źródło: Rembelski P. *Algorytmy I Struktury Danych. Wykłady* [online] Dostępny: http://edu.pjwstk.edu.pl/wyklady/asd/scb/asd10/main10_p4.html (odwiedzona 05.01.2012).

- Aby sprawdzić czy dane słowo *w* należy do zbioru reprezentowanego przez tablicę TAB, należy wyliczyć wartość funkcji znacznik i sprawdzić co znajduje się na pozycji *TAB(znacznik(w))*

[6] Rembelski P. *Algorytmy I Struktury Danych. Wykłady* [online] Dostępny: http://edu.pjwstk.edu.pl/wyklady/asd/scb/asd10/main10_p4.html (odwiedzona 05.01.2012).

- Jeśli pozycja jest pusta lub znajduje się tam inne, niż *w*, słowo, to słowo w nie należy do zbioru X. W przeciwnym przypadku w należy do X.

Stosowanie tablicy mieszającej wiąże się z problem kolizji tzn. przypisania przez funkcję mieszającą tej samej wartości dwóm różnym kluczom. Dla omówionego przykładu można to zilustrować próbą dodania do tablicy słowa '*kasia*', które ma ten sam adres co już użyte słowo '*piotr*'

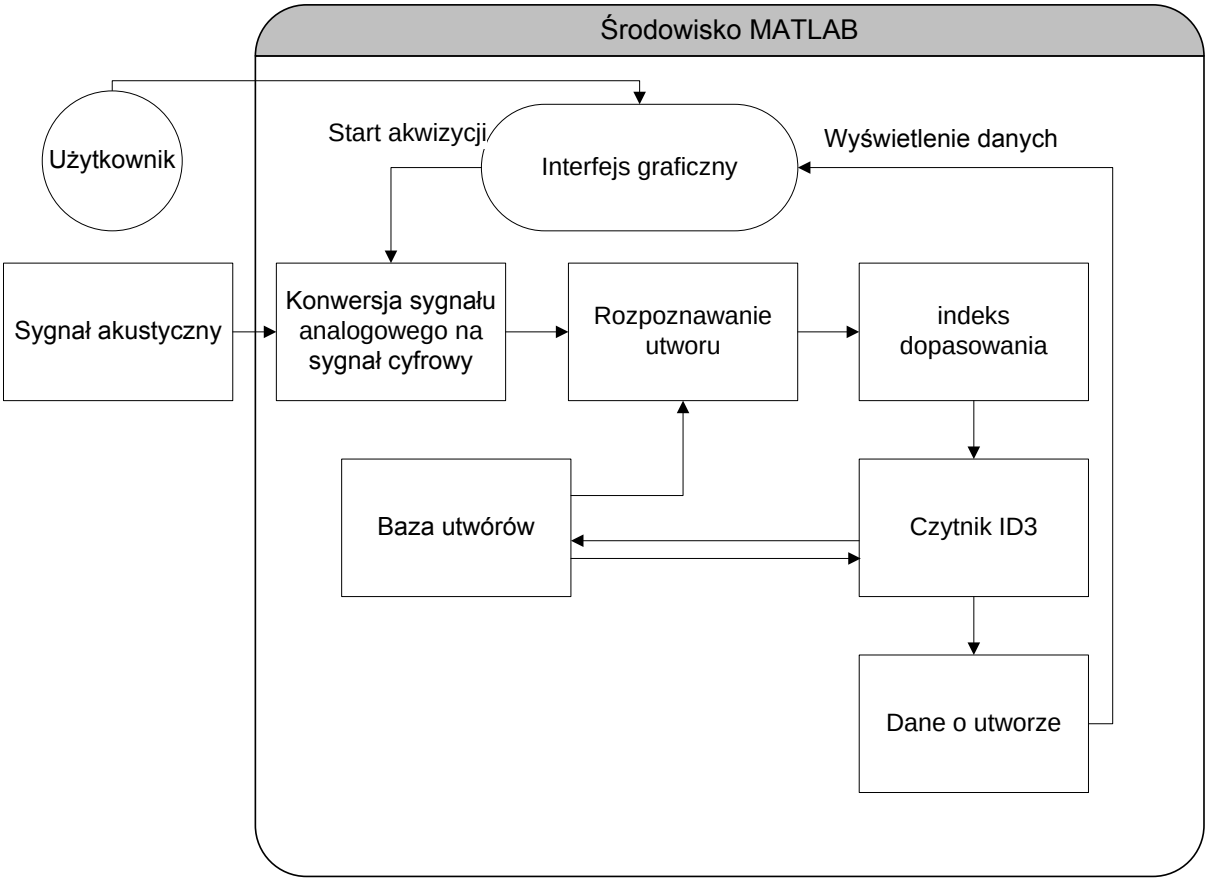
- *znacznik(kasia)= (11+1+19+9+1) mod 13= 2,*
  znacznik(piotr)= (17+9+16+20+18) mod 13= 2,

Możliwość wystąpienia problemu kolizji należy rozpatrywać podczas projektowania tablicy mieszającej. Istnieje różne sposoby rozwiązywania tego problemu. Najprostszym sposobem jest zastąpienie elementu znajdującego się w tablicy przez nowy element lub ewentualnie rezygnacja z wstawiania nowego elementu, w innym przypadku należy użyć innej metody do przechowywania informacji.

3. Implementacja narzędzia do rozpoznawania utworów

3.1. Architektura systemu

Program działa w środowisku MATLAB, który jest podstawowym elementem pracy systemu. Wszystkie funkcje programu są zapisane w plikach tekstowych (*.m), a MATLAB jest środowiskiem, który potrafi je wykonać. Do działania programu, koniecznym jest lokalizacja wszystkich plików we wspólnym folderze docelowym, wraz bazą utworów.



Rysunek 12. Schemat działania narzędzia do automatycznego rozpoznawania utworów

Na rysunku nr 12, został przedstawiony schemat działania programu. Do komunikacji użytkownika z programem służy interfejs graficzny, który również jest elementem środowiska MATLAB. Daną wejściową systemu jest sygnał akustyczny. Użytkownik poprzez przycisk: '*Rozpoznaj utwór'* uruchamia funkcję odpowiedzialną za akwizycję sygnału, czyli zamianę sygnału akustycznego na sygnał cyfrowy. Sygnał akustyczny jest rejestrowany poprzez mikrofon. Czas rejestracji wynosi 7 sekund. Elementem końcowym akwizycji jest postać sygnału akustycznego w postaci cyfrowej,

czyli wektora z wartościami amplitudy sygnału dla każdej kolejnej próbki. Wektor wartości zostaje wysłany do funkcji odpowiedzialnej za rozpoznawanie utworu.

W tym momencie, warto zaznaczyć, że program uruchamiając się wczytuje bazę utworów. Baza utworów do folder z plikami w formacie MP3. Zawartość folderu to zbiór utworów, które program potrafi rozpoznać. Wczytując program, funkcja rozpoznawania przetwarza każdy utwór do postaci zbioru znaczników, który jest sygnaturą każdego utworu, umożliwiającego dopasowanie do sygnału wejściowego.

Sygnał wejściowy jest przetwarzany tym samym algorytmem, co utwory z bazy. Wektor z sygnałem wejściowym poprzez funkcję rozpoznawania jest przetwarzany do wcześniej wspomnianej postaci zbioru znaczników, czyli jest tworzona jego sygnatura. Funkcja rozpoznawania dysponując utworem w postaci, zgodnej z postacią utworów bazy, jest w stanie znaleźć najbardziej dopasowany utwór, czyli taki który ma najwięcej wspólnych znaczników. Elementem wejściowym funkcji rozpoznawania jest indeks utworu najbardziej dopasowanego, czyli numer pliku z bazy.

Następnie znając indeks rozpoznanego utworu, do funkcji odpowiedzialnej za odczyt informacji z pliku MP3, wysyłana jest ścieżka dopasowanego pliku. Funkcja odczytu informacji obsługuje standard ID3v2. Ważne jest zatem, aby ścieżki z bazy były zgodne z tym standardem, gdyż tam zawarte są informacje, między innymi o tytule i nazwy wykonawcy utworu. Daną wyjściową funkcji odczytu informacji ID3 jest tytuł piosenki i nazwa artysty ją wykonującego.

Końcowym etapem sytemu jest wyświetlenie danych w oknie interfejsu graficznego programu, z której użytkownik może odczytać tytuł i wykonawcę fragmentu piosenki poddanej procesowi rozpoznania.

Kolejny proces rozpoznawania utworu jest możliwy poprzez ponowne wciśnięcie przycisku: *‘Rozpoznaj utwór’*.

3.2. Funkcja rozpoznawania utworu

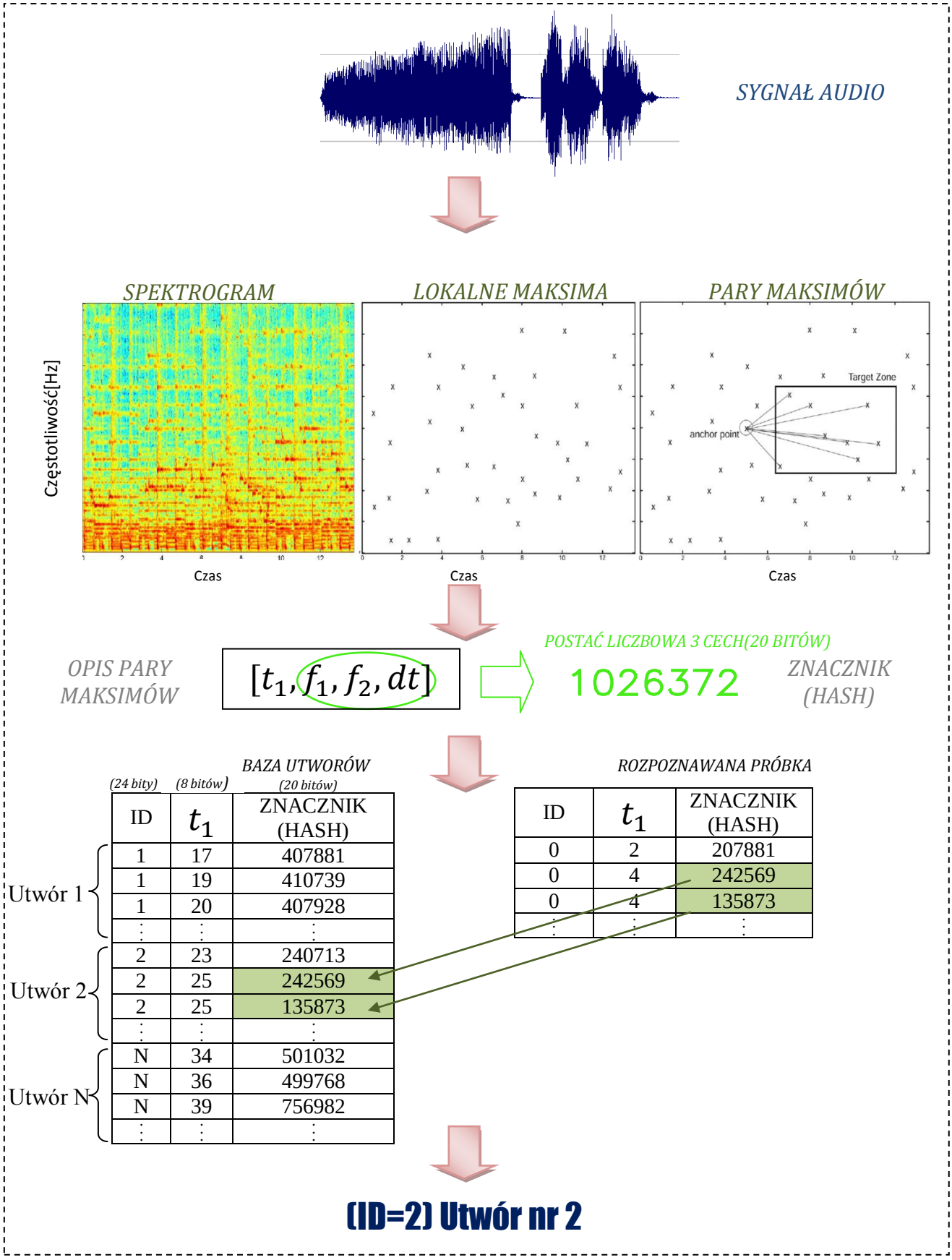
W systemie została wykorzystana funkcja rozpoznawania utworów zrealizowana przez Dana Ellis'a.[7] Jest to implementacja w środowisku MATLAB metody rozpoznawania utworów wzorowanej na algorytmie Avery Wang'a(2003), która jest przykładem metody rozpoznawania przez przykład i znalazła komercyjne zastosowanie w serwisie Shazam. Zdecydowano się wykorzystać to rozwiązanie, ponieważ analiza utworu poprzez spektrogram, uznawana jest obecnie za najbardziej wydajną metodą rozpoznawania utworów, co potwierdza jej komercyjne zastosowanie.

3.2.1. Opis działania

Metoda jest oparta na tworzeniu 'odcisku palca' sygnału akustycznego w oparciu o charakterystyczne punkty spektrogramu. (z ang. *landmark-based audio fingerprinting*). W pierwszej kolejności algorytm, lokalizuje lokalne maksima spektrogramu. Następnie znalezione maksima są łączone w pary i parametryzowane, poprzez ich częstotliwości i różnice w czasie między nimi. Są dekodowane względem częstotliwości pierwszego prążka widma. Wykonując 512-elementową FFT, na sygnale o częstotliwości próbkowania 11025 Hz, otrzymują się pasma o szerokości 21,5 Hz, co wymaga 8 bitów. Informacja na temat odległości pomiędzy maksimami, w dziedzinie częstotliwości, wymaga 6 bitów, gdyż nie zakłada się zapisywania większych różnic. Zapisanie różnicy w czasie wymaga 6 bitów, jednostka skoku to 32 ms. Oznaczenie znalezionej pary wymaga zatem 20 bitów, zestaw tych informacji jest nazywany przez autora hash'em, którą możne rozumieć jako znacznik. 20 bitów na znacznik oznacza, że implementacja umożliwia zapisanie 1 miliona (20^2) różnych znaczników. Parametry zostały tak dostrojone, aby dla 1 sekundy utworu, otrzymać 20-50 znaczników.

Każdy utwór referencyjny jest opisywany przez setki znaczników. Do ich przechowywania służy tablica znaczników, gdzie trafiają wraz z informacją o tym kiedy występują. Jest rodzaj tablicy mieszającej (z ang. *hashtable*), która umożliwia szybkie porównywanie danych. Indeks (24 bity) i czas wystąpienia(8 bitów) są pakowane do 32-bitowej liczby, która jest przechowana pod 20-bitowym adresem określanym przez znacznik.

[7] D. Ellis, *Robust Landmark-Based Audio Fingerprinting*, [online], Dostępny: http://labrosa.ee.columbia.edu/matlab/fingerprint/. 2009. (odwiedzona 05.01.2012).



ID	t_1	ZNACZNIK (HASH)
1	17	407881
1	19	410739
1	20	407928
⋮	⋮	⋮
2	23	240713
2	25	242569
2	25	135873
⋮	⋮	⋮
N	34	501032
N	36	499768
N	39	756982
⋮	⋮	⋮

ID	t_1	ZNACZNIK (HASH)
0	2	207881
0	4	242569
0	4	135873
⋮	⋮	⋮

Rysunek 13.Schemat funkcji rozpoznawania utworów. Opracowanie własne na postawie: A.Wang. *An industrial-strength audio search algorithm*. International Conference on Music Information Retrieval (ISMIR), Baltimore, Maryland, USA, October, 2003.

Wyszukiwanie polega na tym, że zarejestrowana próbka w postaci cyfrowej jest konwertowana do postaci opisanych powyżej znaczników. Następnie baza jest przeszukiwana pod kąt znalezienia wspólnych znaczników, których relatywny czas występowania jest taki sam. Utwór referencyjny, który posiada najwięcej wspólnych znaczników stanowi wynik wyszukiwania. Do rozpoznania wystarcza niewielka liczba znaczników (np.5), gdyż prawdopodobieństwa dopasowania jest niewielkie. Schemat funkcji wyszukiwania utworów został przedstawiony na rysunku numer 13.

3.2.2. Implementacja w MATLABIE

Zbiorem danych wejściowych jest wektor `tks`, który zawiera ścieżki do plików, z których ma zostać zbudowana baza systemu. Proces rozpoznawania jest realizowany przez szereg funkcji autorstwa Dan'a Ellis'a. Poniżej zostały krótko opisane funkcje wykorzystywane przez narzędzie do automatycznego rozpoznawania utworów.

```
clear_hashtable ()
```

funkcja bezargumentowa, która nie zwraca żadnej wartości inicjalizuje tablice globalną, które przechowuje znaczniki i stanowi bazę systemu

```
[N,T]=add_tracks(D,SR,ID)
```

N – liczba dodanych znaczników
T – łączny czas dodanych utworów
D – wektor sygnału wejściowego
SR – częstotliwość próbkowania
ID –numer referencyjny

Funkcja odpowiadająca za wczytanie danych z wektora `tks`, odpowiedzialna za przetworzenie zbioru wszystkich plików. Wykorzystuje szereg funkcji: `find_landmarks, landmark2hash, record_hashes, audioread`

```
[L,S,T,maxes] = find_landmarks(D,SR,N)
```

L – 4-kolumnowa macierz opisująca pary maksimów [d_1 f_1 f_2 dt]
S – macierz powierzchni filtrowanej logarytmicznie
T – macierz rozkładu powierzchni progowej
maxes – macierz lokalnych maskimów, opisywanych przez dwie wartości
D – wektor sygnału wejściowego
SR – częstotliwość próbkowania
N – liczba znaczników na sekundę

Funkcja odpowiedzialna za lokalizowanie lokalnych maksimów w spektrogramie i łączenie je w pary, tworzy znaczniki. Jej argumentem jest wektor z sygnałem akustycznym. Zwraca 4-elementowy zestaw cech sparowanych znaczników`[d1 f1 f2 dt]`: czas pojawienie się pierwszego maksimum, częstotliwość pierwszego maksimum, częstotliwość drugiego maksimum, różnice w czasie.

```
H = landmark2hash(L,S)
```

H – 3-kolumnowa macierz znaczników [indeks, czas wystąpienia, znacznik(hash)]
L – 4-kolumnowa macierz opisująca pary maksimów [d_1 f_1 f_2 dt]
S – macierz powierzchni filtrowanej logarytmicznie

Funkcja odpowiedzialne za konwersje 4-elementowego wyniku funkcji `find_landmarks [d1 f1 f2 dt]` do postaci 3-kolumnowej tablicy. Ostatnia kolumna to 20-bitowy znacznik(*hash*), obliczony na podstawie `[f1 f2 dt]`, druga kolumna to czas wystąpienia w utworze, a pierwsza kolumna do indeks danego utworu.

```
N = record_hashes(H)
```

N – liczba zapisanych znaczników
H – 3-kolumnowa macierz znaczników [indeks, czas wystąpienia, znacznik(hash)]

Funkcja odpowiedzialna za zapis do pamięci globalnej kolejnych tablic znaczników, zachowująca jej podział na 3 kolumny: indeks utworu, indeks czasu startu, znacznik(*hash*)

```
[D, SR] = audioread(FN)
```

D – wektor sygnału wejściowego
SR – częstotliwość próbkowania
FN – wczytywany plik audio

Funkcja, która używa funkcji: `wavread, mp3read, m4aread, flacread`. W zależności od formatu wczytanego pliku audio [*.wav ,*mp3, *.m4a, *.flac] uruchamia odpowiednią funkcję odczytu.

```
match_query [R,L] = match_query(D,SR,IX)
```

R– macierz dopasowania
L– znaczniki dopasowania z najbardziej dopasowanym utworem
D – wektor sygnału wejściowego
SR – częstotliwość próbkowania
IX –numer dopasowanie, którego znaczniki mają być zapisane w L

Funkcja odpowiedzialna za odnajdywanie w bazie znaczników, które są takie same jak znaczniki rozpoznawanego sygnału audio. Zwraca wszystkie dopasowania, uszeregowane według jakości dopasowania. Każde dopasowanie jest określone przez 3 parametry: indeks utworu w bazie, liczba dopasowanych znaczników, czas wystąpienia w danym utworze. Funkcja wykorzystuje: `hash2landmark`, `get_hash_hits`

```
L = hash2landmark(H)
```

L– 4-kolumnowa macierz opisująca pary maksimów [d_1 f_1 f_2 dt]
H– 3-kolumnowa macierz znaczników [indeks, czas wystąpienia, znacznik(hash)]

Funkcja odpowiedzialna za konwersje 20-bitowego znacznika do 4-elementowej postaci `[d1 f1 f2 dt]`

```
R = get_hash_hits(H)
```

R– macierz dopasowania
H– 3-kolumnowa macierz znaczników [indeks, czas wystąpienia, znacznik(hash)]

Funkcja obliczająca liczbę dopasowań do danego utworu referencyjnego

```
[Y,FS] = mp3read(FILE)
```

Y– wektor sygnału wejściowego
H– częstotliwość próbkowania
FILE– plik wejściowy

Funkcja umożliwiająca odczyt plików w formacie MP3 w sposób analogiczny do funkcji wavread, będącej elementem środowiska MATLAB

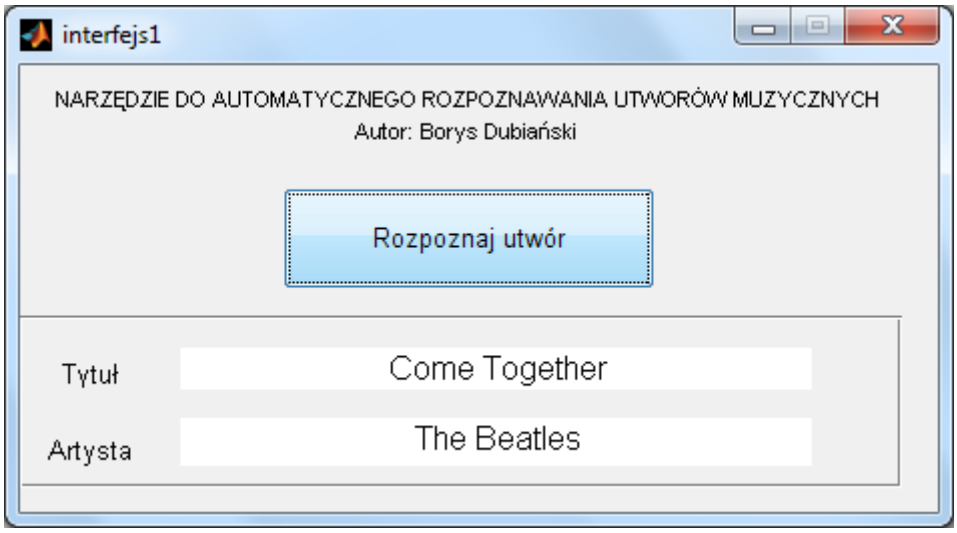
```
id3parse rOut = id3parse(file)
```

rOut– struktura danych z informacjami przechowywanymi w kontenerze ID3v2
file – plik wejściowy

Funkcja autorstwa Brian'a Mearns'a, z modyfikacjami Gordon'a Forbes'a. Umożliwia odczyt danych z plików w formacie MP3, które zawierając informacje w kontenerze danych o standardzie ID3v2.3. W przypadku narzędzia do automatycznego rozpoznawania utworów funkcja jest wykorzystywana do odczytu z pliku MP3 tytułu piosenki oraz nazwy wykonawcy.

3.3. Interfejs graficzny programu

Celem projektowym interfejsu graficznego była łatwość obsługi i przejrzystość, dlatego zaprojektowane okno posiada tylko jeden przycisk '*Rozpoznaj utwór'* i dwa pola tekstowe do wyświetlania danych: 1) tytuł utworu, 2) nazwa artysty wykonującego dany utwór.



Rysunek 14. Interfejs graficzny narzędzia do automatycznego rozpoznawania utworów muzycznych

3.5. Instrukcja obsługi i schemat działania

Przed uruchomieniem programu, koniecznym jest, aby w jednym folderze znajdowały się pliki wszystkich funkcji oraz folder '*baza*', który powinien zawierać utwory w formacie MP3 z kontenerem danych w standardzie ID3v2. Ponadto, program wymaga mikrofonu, który powinien być podłączony do karty dźwiękowej komputera.

Program należy uruchomić w środowisku MATLAB poprzez funkcje 'interfejs1.m', który wyświetli na ekranie interfejs graficzny. Uruchomienie programu wywoła funkcję '*wczytaj'*, która jest opowiedziana za wczytanie bazy utworów, czyli przetworzenie do postaci znaczników wszystkich plików MP3, które znajdują się w folderze '*baza*'. Kiedy na ekranie pojawi się interfejs graficzny, program jest już gotowy do pracy.

Aby rozpoznać utwór, należy wcisnąć przycisk '*Rozpoznaj utwór'*, który uruchamia funkcje '*szukaj*', która prowadzi do rejestracji przez mikrofon próbki nagrania o długości 7 sekund, a następnie przeszukuje bazę, korzystając z funkcji `match_query` w celu dopasowania znaczników utworów bazy do znaczników badanej próbki. Funkcja '*szukaj'* sprawdza indeks najbardziej dopasowanego utworu,

a następnie przy użyciu funkcji id3parse, odczytuje z dopasowanego pliku MP3, tytuł piosenki i nazwę artysty. Informacja o tytule i artyście jest wyświetlona w odpowiednim polu w interfejsie graficznym.

W przypadku, gdy do danego utworu zostały dopasowane mnie niż 2 znaczniki, utwór uznaje się za nieznany i wyświetlany jest komunikat '*brak danych*'.

Kolejne rozpoznanie jest możliwe poprzez ponowne wciśnięcie przycisku '*Rozpoznaj utwór'*.

4. Ocena działania programu

4.1. Wprowadzenie

Działanie narzędzia do automatycznego rozpoznawania utworów sprawdzono na podstawie różnych testów, w których była wyznaczana skuteczność rozpoznawania utworów. W analizie uwzględniono wpływ poziomu hałasu na jakość rozpoznania. Sprawdzono również, jak długość próbkowania wpływa na skuteczność. Większość testów została przeprowadzona na poziomie cyfrowym, tzn. zniekształcenie sygnału odbywało się poprzez addytywne dodanie próbki hałasu do próbki utworu z bazy danych utworów. W celu weryfikacji testów cyfrowych dokonano również jednego testu poprzez rejestrację rozpoznawanego sygnału za pomocą mikrofonu. Test miał również na celu zbadanie wpływu zniekształceń częstotliwościowych mikrofonu na skuteczność rozpoznania.

Cześć ustawień funkcji rozpoznawania utworów można zmieniać, do testów zostały wykorzystane domyślne ustawienia, zaproponowane przez D.Ellisa, które w przybliżeniu obliczają 10 znaczników na sekundę utworu. [8]

4.2. Baza utworów

Baza systemu na potrzeby testowe to pliki w formacie MP3, z zakodowanymi informacjami w standardzie kontenera danych ID3v2. W bazie znalazły się 2-minutowe utwory reprezentujące różne style muzyczne, między innymi: rock, jazz, funk, pop, muzyka elektroniczna, etc. Szczegółowe dane zostały przedstawione w tabeli nr 3

Tabela 3. Charakterystyka bazy danych

Liczba utworów	100
Łączny czas trwania[s]	12014,5 ≈ 3h 20 min
Łączna ilość znaczników	137715
Liczba znaczników na sekundę	11,5

[8] Tamże, str. 24

4.3. Czas działania

Zmierzono czas działania programu. Na funkcjonalność systemu mają wpływ: a) czas wczytania bazy utworów b) czas wyszukiwania utworu.

Czas potrzebny na wczytanie bazy o łącznej długości utworów 3 godzin i 20 minut wynosi blisko 6 minut, na podstawie czego można obliczyć, że czas potrzebny na wczytanie jednej 1 minuty utworów to około 2 sekundy. Zakładając, że jeden album trwa średnio 50 minut, oznacza to że czas wczytania 100 albumów wyniósłby ok. 2 godzin i 50 minut.

Czas wyszukiwania utworu jest bardzo krótki i wynosi niecałą sekundę (0,6 s), czyli od momentu rozpoczęcia rejestracji próbki do wyświetlenia wyniku w interfejsie graficznym mija niecałe 8 sekund. W zależności od wielkości bazy czas nieznacznie się zmienia, dla bazy o wielkości 10 30-sekundowych utworów czas wyszukiwania wynosił również podobny czas (0,62s).

Analizując uzyskane czas, należy uwzględnić moc obliczeniową komputera dlatego w tabeli nr 4 podano specyfikacje komputera testowego.

Tabela 4. Parametry komputera testowego

System operacyjny	Windows 7 Professional 64-bit
Procesor	Intel® Core™ 2 CPU T6600 2.20 GHz
Pamięć RAM	4GB
Karta dźwiękowa	Sound Blaster Live 5.1
Mikrofon	Dynamiczny ION Audio
Środowisko MATLAB	MATLAB wersja 7.6 (R2008a)

4.4. Wpływ hałasu na jakość rozpoznania

Analiza opiera się na wynikach testów o liczebności 1000 próbek dla każdej konfiguracji. Każda analizowana próbka jest niezależna względem reszty tzn. nie posiada wspólnych fragmentów utworu. Podczas testów czas próbkowanego sygnału wynosił 7 sekund.

Punktem wyjścia analizy było sprawdzenie skuteczności programu na podstawie niezniekształconego sygnału. Rozpoznaniu zostały poddane próbki, wczytane bezpośrednio z bazy danych. System na 1000 prób wskazał tylko 6 błędów oznacza to, że skuteczność dla zniekształconego sygnału wynosi 99,4%. Analizując błędy zauważono, że nierozpoznane fragmenty to ciche i małą dynamiczne fragmenty utworu, gdzie występuje pauza albo spada amplituda sygnału.

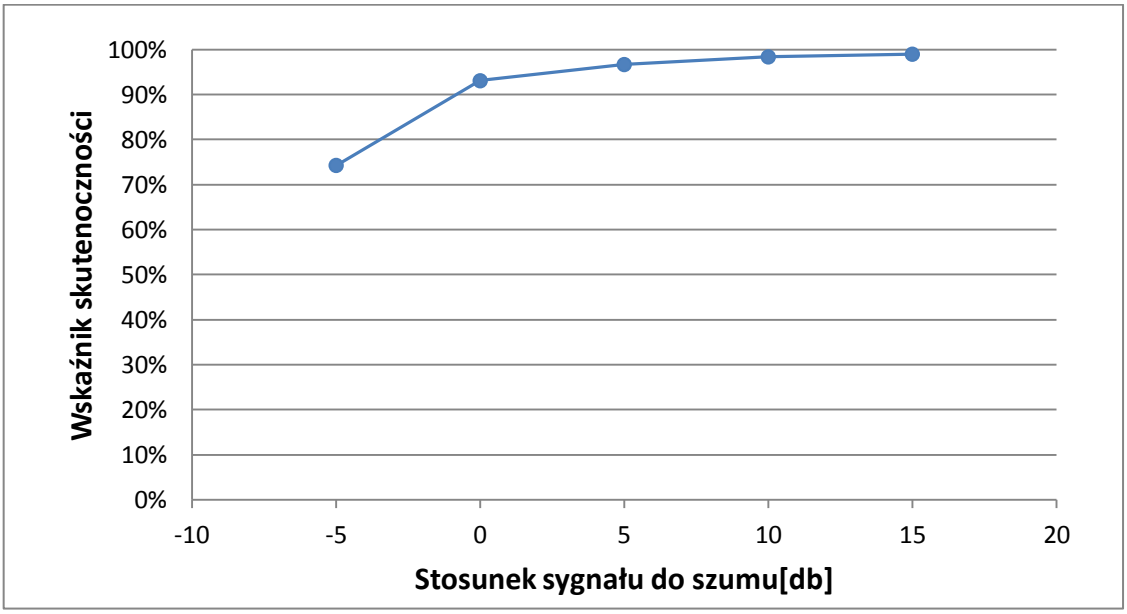
Do testów użyto próbki hałasu zarejestrowanej w centrum handlowych, w celu odzwierciedlenia rzeczywistego rodzaju zakłóceń. Hałas był dodawany addytywnie to sygnału niezniekształconego. Próbka z poziomem hałasu była odpowiednio skalowana względem każdej analizowanej próbki osobno w celu osiągnięcia pożądanego stosunku sygnału do hałasu w skali decybelowej. Stosunek odstępu sygnału do szumu (*ang. signal-to-noise-ratio – SNR*) był obliczony na podstawie stosunku unormowanych sygnałów.

$$SNR = 20\log_{10}\left(\frac{\|niezniekształcony\ sygnał\|}{\|hałas\|}\right)[db] \tag{3}$$

Wyznaczono skuteczności działania programu względem następujących stosunków SNR: 15, 10, 5, 0,-5. Skuteczność systemu jest proporcjonalna do poziomu hałasu co można zaobserwować na rysunku nr 15. Im większy odstęp, tym większa skuteczność. Dokładne wartości zostały przedstawione w tabeli nr 4. Na postawie wyników z tabeli, można stwierdzić, że skuteczność rozpoznania nawet dla stosunku sygnału do szumu na poziomie 0 db wynosi ponad 90%. Dla wartości SNR równej 15 dB, wskaźnik rozpoznania wynosi 99%.

Tabela 5. Skuteczność rozpoznania w zależności od poziomu zakłóceń

stosunek sygnału do hałasu[dB]	liczba błędów	poprawne wskazania	liczba próbek	wskaźnik skuteczności rozpoznania
brak hałasu	6	994	1000	99,4%
15	10	990	1000	99,0%
10	16	984	1000	98,4%
5	33	967	1000	96,7%
0	69	931	1000	93,1%
-5	257	743	1000	74,3%



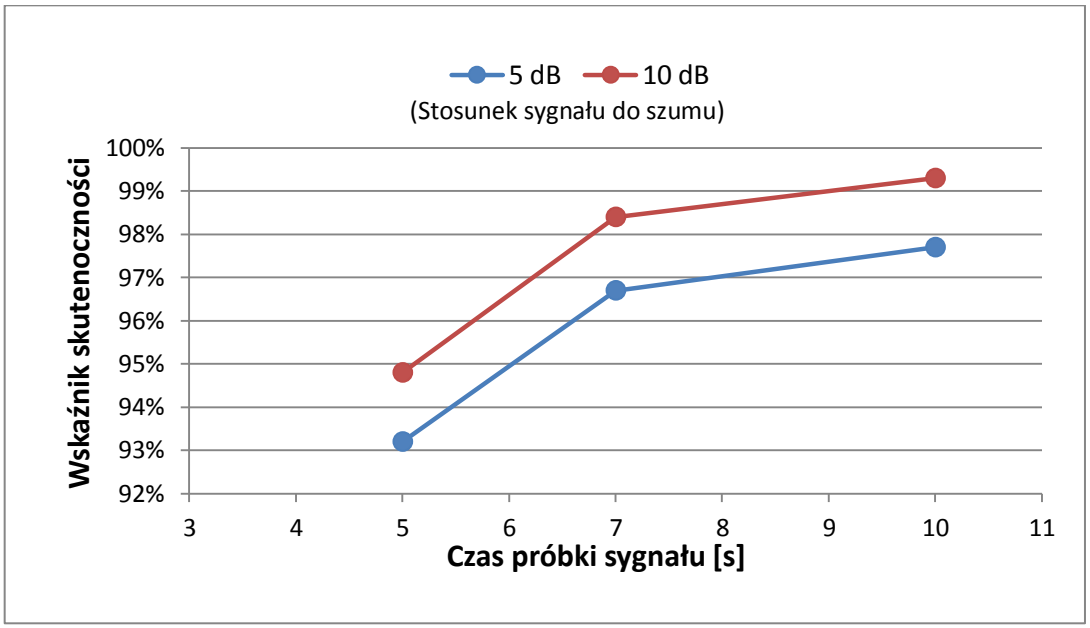
Rysunek 15. Skuteczność rozpoznania w zależności od poziomu zakłóceń

4.4. Wpływ czasu próbkowania na jakość rozpoznania

Zbadano również wpływ długości czasu próbkowania sygnału na skuteczność rozpoznawania utworu. Dla stosunku sygnału do hałasu o poziomie 5 oraz 10 decybeli, przeprowadzono testy z różnymi czasami analizowanych próbek. Czas analizowanego sygnału wynosił odpowiednio 5,7 lub 10 sekund.

Tabela 6. Skuteczność rozpoznania w zależności od długości próbki sygnału

		Stosunek sygnału do szumu [dB]					
		5			10		
Długość próbki [s]		Błędne odp.	Poprawne odp.	**Skuteczność rozpoznania**	Błędne odp.	Poprawne odp.	**Skuteczność rozpoznania**
	5	68	932	**93,2%**	52	948	**94,8%**
	7	33	967	**96,7%**	16	984	**98,4%**
	10	23	977	**97,7%**	7	993	**99,3%**

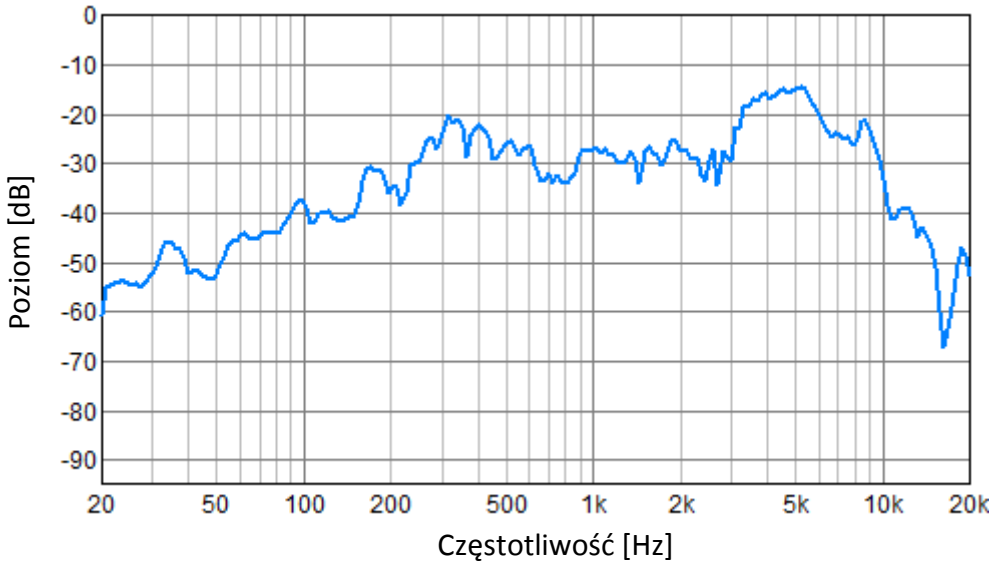


Rysunek 16. Skuteczność rozpoznania w zależności od czasu próbki

Długość próbkowania sygnału wpływa korzystnie na skuteczność rozpoznawania. Jak widać na rysunku nr 16 jest zależność skuteczności od czasu próbki jest proporcjonalna. Im dłuższy czas próbki sygnału, tym większa skuteczność rozpoznania. Ta prawidłowości są zgodne zarówno dla poziomu SNR równego 5 oraz 10 decybeli. W tabeli nr 5 zostały przedstawione dokładne wartości liczby błędów dla danej próby.

4.5. Wpływ zniekształceń mikrofonu na jakość rozpoznania

Aby sprawdzić działanie narzędzia do automatycznego rozpoznawania utworów w warunkach rzeczywistych, wykonano próbę polegająca na rejestracji sygnału poprzez mikrofon. Ta próba daje obraz jak zniekształcenia sygnału przez mikrofon wpływają na rozpoznanie. Czas rejestracji wynosił 7 sekund. W programie diagnostycznym Realtime Analyzer zbadano charakterystykę częstotliwościową



Rysunek 17. Charakterystyka częstotliwościowa mikrofonu testowego

mikrofonu, która została przedstawiona na rysunku nr 17. Źródłem sygnałów testowych były utwory muzyczne odtwarzane na zestawie 3 głośników o mocy skutecznej: 2 x 8 W (głośniki wysoko-średniotonowe), 18 W (głośnik niskotonowy). Założenie badania stanowi, że wszystkie odgrywane utwory znajdują się w bazie systemu, także teoretycznie każdy utwór jest znany przez system. Celem było rejestrowanie sygnałów dobrze słyszalnych. Membrana mikrofonu rejestrującego była położona 30 cm od membrany prawego głośnika.

Dla każdej rejestrowanej próbki obliczano stosunek sygnału do szumu, w celu dokładnej analizy wyników i możliwości bieżącego dostrajania modelu. Zarejestrowano 1000 niezależnych próbek. Ze względu na to że utwory różnią się poziomem głośności między sobą, stosunek sygnału do szumu wahał się w przedziale 5-12 decybeli. Średnia wartość stosunku dla wszystkich prób wyniosła 8,2 dB. Dla człowieka utwory byłe relatywnie dobrze słyszalne. Liczba błędnych odpowiedzi

wyniosła 21, zatem wskaźnik skuteczności wyniósł 97,9%. Dla zbioru błędnych odpowiedzi stosunek sygnału do szumu nie przekraczał 9 decybeli. Uzyskany wynik potwierdza wiarygodność testów przez cyfrowe zaszumienie sygnału, gdyż dla SNR=10 dB skuteczność była równa 98,4%, a dla SNR=5 to 96,7%. Uzyskana średnia wartość SNR=8,2 dB wskazuje wartość skuteczności leżącą pomiędzy tymi dwoma wartościami. Ponadto można stwierdzić, że zniekształcenia mikrofonu nie wpływają znacząco na wskaźnik rozpoznania utworów.

4.6. Podsumowanie

Czas rozpoznania jest krótki, wyszukiwanie utworu po rejestracji wymaga mniej niż 1 sekundę, co oznacza, że łącznie z próbkowaniem sygnału czas rozpoznania wynosi 8 s. Analiza sygnału bez zakłóceń o długości 7 sekund daje skuteczność 99,4%. Na podstawie uzyskanych wyników, możne powiedzieć że program cechuje się wysoką skutecznością działania, nawet przy występujących zakłóceniach sygnału. Dla stosunku sygnału do hałasu o wartości 0 dB, skuteczność programu nie spada poniżej 93,1%, dla odstępu na poziomie 15 dB wynosi 99%. Kiedy poziom zakłóceń jest o 5 decybeli większy niż sygnał poddany do analizy, skuteczność spada do 74,3%. Zarówno czas próbkowania, jak i stosunek sygnału do hałasu proporcjonalnie wpływają na wskaźnik skuteczności działania programu. Dla określonego poziomu zakłóceń, najwyższą skuteczność cechuję test dla najdłuższej długości próbkowania. Ponadto, wykazano, że zniekształcenia częstotliwościowe mikrofonu nie mają większego wpływu na wyniki rozpoznania. W skrócie wynik badań można sprowadzić do krótkich wniosków:

- czas wyszukiwania to mniej niż 1s
- dla niezniekształconego sygnału system ma skuteczność rozpoznania 99,4%,
- odstęp sygnału od szumu ma proporcjonalny wpływ na skuteczność rozpoznania, dla SNR =15 (99%), dla SNR=10 (98,4), a dla SNR=5(96,7%), czas próbkowania 7 sekund
- czas próbkowania sygnału ma proporcjonalny wpływ na skuteczność rozpoznania, dla t=5s (94,8%), dla t=7s(96,7%), dla t=10(99,3%), dla SNR=10 dB
- zniekształcenie mikrofonu mają mały wpływ na skuteczność rozpoznania. Dla średniej wartości SNR=8db i czas próbkowania t=7s, skuteczność wyniosła 97,9%.

Wnioski

W ramach pracy dyplomowej zrealizowano narzędzie do automatycznego rozpoznawania utworów. W wyniku analizy literatury przedmiotu wybrano metodę rozpoznawania utworów przez przykład, opartą na analizie spektrogramu. Działanie programu opiera się na podejściu Avery'ego Wang'a, metody opisanej w rozdziale 1.5. oraz 3.2.1., która polega na lokalizacji lokalnych maksimów spektrogramu i odpowiedniego przekształcenia informacji do postaci 20-bitowej liczby, przechowywanej w tablicy. Do implementacji tej metody w środowisku MATLAB wykorzystano funkcje rozpoznawania zrealizowaną przez Dan'a Ellis'a opisanej rozdziale 3.2.2. W celu wygodnej obsługi przez użytkownika zaprojektowano interfejs obsługi graficznej, opisany w rozdziale 3.3.

Skuteczność działania systemu zostało poddana testom o liczebności 1000 prób, opisanym dokładnie w rozdziale 4. Kluczowe informacje to czas wyszukiwania utworu w bazie równy 1 sekundzie. Ponadto, skuteczność rozpoznania niezniekształconego sygnału o długości 7 sekund wynosi 99,4%. Dla poziomu stosunku sygnału użytkowego do sygnału zakłóceń równego 15 dB, wskaźnik rozpoznania utworu wynosi 99%. Zarówno czas próbkowania, jak i stosunek sygnału do sygnału zakłóceń, proporcjonalnie wpływa na wskaźnik skuteczności działania programu. Dodatkowo, zniekształcenia częstotliwościowe mikrofonu nie mają większego wpływu na wyniki rozpoznania. Dla rejestrowanego sygnału o stosunku do sygnału zakłóceń równym 8 dB, skuteczność rozpoznania systemu wynosi 97,9%.

Dalsze prace nad zrealizowanym narzędziem do automatycznego rozpoznawania utworów mogą stanowić pracę nad poprawą skuteczności programu dla niskiego stosunku sygnału do sygnału zakłóceń. Ponadto, warto zastanowić się nad próbą przeniesienia narzędzia na urządzenia mobilne, takie jak telefony komórkowe. Dodatkowo, narzędzie może być rozwinięte o funkcje rozpoznawania utworów poprzez śpiew.

Spis Rysunków

Spis Tabel

Literatura

1. Baluja S., Covell M.. *Content fingerprinting using wavelets*, 3rd European Conference on Visual Media Production, pp 198 - 207, Nov. 2006.
2. Chandrasekhar, V.; Sharifi, M. & Ross, D. A., *Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications*., in. Anssi Klapuri & Colby Leider, ed., 'ISMIR', University of Miami, p.801-806, 2011
3. Ellis D., *Robust Landmark-Based Audio Fingerprinting*, [online], Dostępny: http://labrosa.ee.columbia.edu/matlab/fingerprint/. 2009. (odwiedzona 05.01.2012).
4. Gałka J. *Zastosowanie transformacji falkowych do analizy sygnału mowy.* AGH, Kraków, 2003.
5. Ghias A., J. Logan, D. Chamberlin, B. C. Smith. *Query by humming - musical information retrieval in an audio database*, In ACM Multimedia 95, 1995
6. ID3.org, *What is ID3(v1)?*[online] Dostępny: http://id3.org/ID3v1 (odwiedzona 05.01.2012).
7. Izumitani T., Mukai R., Kashino K., *A Background Music Detection Method Based on Robust Feature Extraction*, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), pp. 13--16, Apr. 2008.
8. Ke Yan, Hoiem Derek, Sukthankar Rahul: *Computer Vision for Music Identification*. 2005
9. Kłaczyński M. *Zjawiska wibroakustyczne w kanale głosowym człowieka.* AGH, Kraków, 2007.
10. Rembelski P. *Algorytmy I Struktury Danych. Wykłady* [online] Dostępny: http://edu.pjwstk.edu.pl/wyklady/asd/scb/asd10/main10_p4.html (odwiedzona 05.01.2012).
11. Su Ja-Hwung, Wu Cheng-Wei, Fu Shao-Yu, Lin Yu-Feng, Chang Wei-Yi, I-Bin Liao, Chang Kuo-Wei, Vincent. S. Tseng. , *Empirical Analysis of Content-based Music Retrieval for Music Identification*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
12. Wang A., *The shazam music recognition service*. Commun. ACM 49(8), 44-48(2006)
13. Wang A.. *An industrial-strength audio search algorithm*. International Conference on Music Information Retrieval (ISMIR), Baltimore, Maryland, USA, October, 2003.
14. Wang C.-C., J.-S. R. Jang W. Wang. *An Improved Query by Singing/Humming System Using Melody and Lyrics Information*. Proc.of 11th International Society for Music Information Retrieval Conference, August 2010
15. Wikipedia.org, *Akustyka* [online] Dostępny: http://pl.wikipedia.org/wiki/Akustyka. (odwiedzona 05.01.2012).
16. Wikipedia.org, *ID3* [online] Dostępny: http://pl.wikipedia.org/wiki/ID3 (odwiedzona 05.01.2012).