

*Akustyka muzyczna*

**MIR**

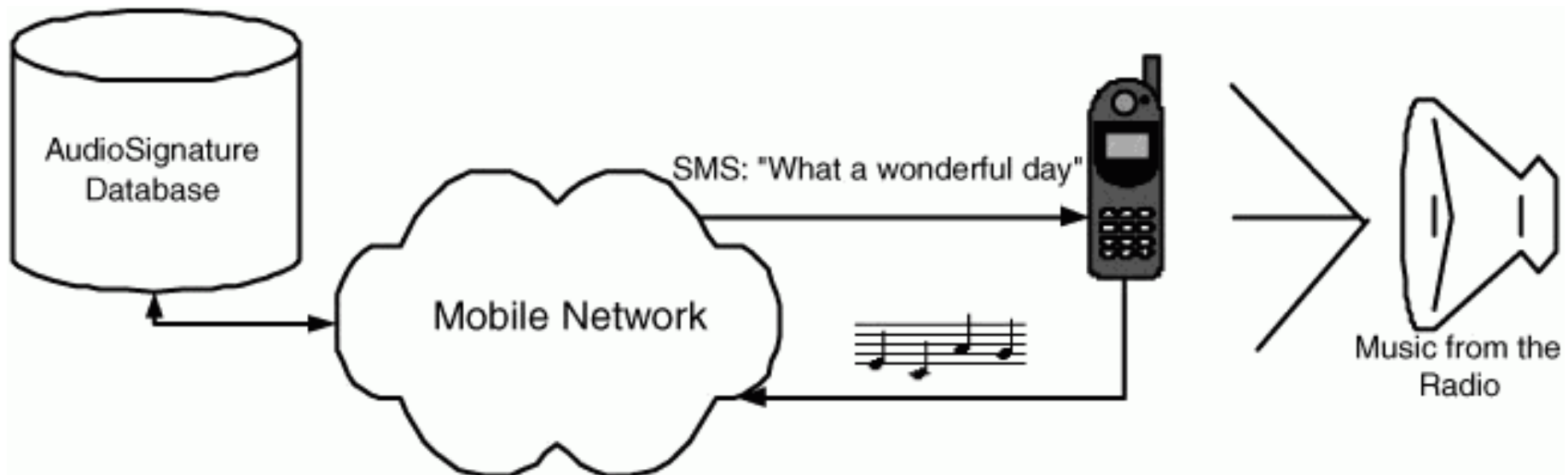
**Systemy rozpoznawania muzyki**

# Multimedialne bazy danych

---

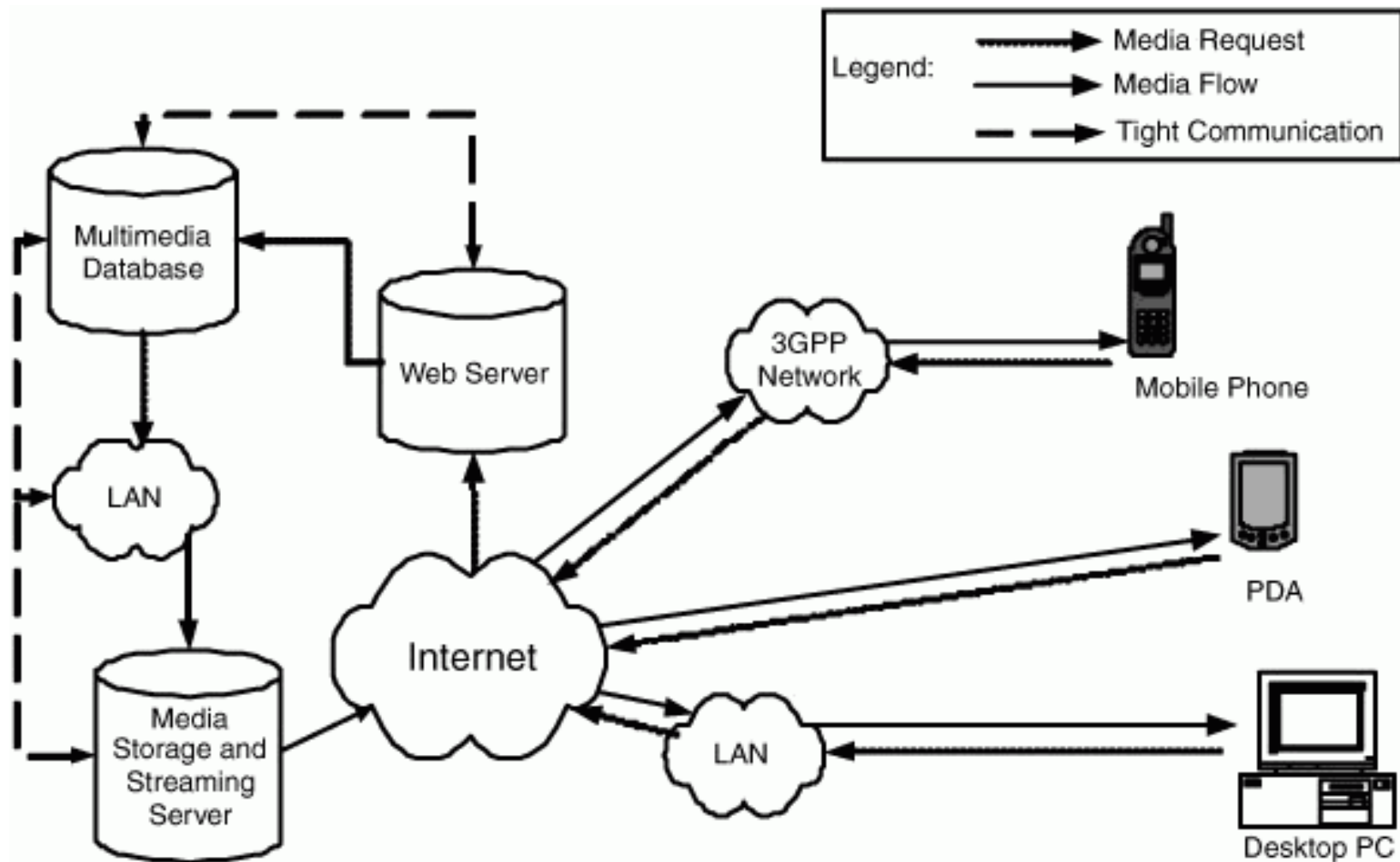
MBD (ang. *Multimedia Database*):

- przechowują dane o zawartości multimedialnej (np. o nagraniach muzycznych)
- umożliwiają wyszukiwanie wg. kryteriów **nietekstowych** (np. fragment nagrania)



# Systemy rozproszonych MBD

Klient łączy się z serwerem systemu za pośrednictwem "chmury"



# Systemy wyszukiwania muzyki

---

**MIR** – ang. *Music Information Retrieval*

Systemy umożliwiające wyszukiwanie muzyki wg kryteriów multimedialnych.

Metody przesyłania zapytań:

- znanucenie lub zagwizdanie melodii
- zapytanie przez przykład – parametryzacja fragmentu nagrania
- bezpośrednio podanie np. zapisu nutowego (rzadko stosowane)

# Zastosowanie systemów MIR

---

Możliwe zastosowania systemów rozpoznawania muzyki:

- wyszukiwanie danych o nagraniu (użytkownik przesyła nagranie lub nuci melodię, chce poznać wykonawcę i tytuł)
- ochrona praw autorskich – porównywanie nagrań, wyszukiwanie plagiatów
- monitorowanie programu radiowego – automatyczne tworzenie listy emitowanych nagrań
- systemy rekomendacji muzyki

# Przesłanie zapytania

---

- Podejście intuicyjne do wyszukiwania:
  - przesyłamy do serwera nagranie (plik)
  - serwer parametryzuje nagranie i dokonuje wyszukiwania
- Wada: duże obciążenie łącza.
- Lepsze rozwiązanie:
  - oprogramowanie po stronie klienta (np. aplikacja mobilna) dokonuje **parametryzacji**
  - do serwera przesyłane są **tylko parametry**
  - serwer dokonuje tylko wyszukiwania i zwrócenia wyników

# Systemy QBH

---

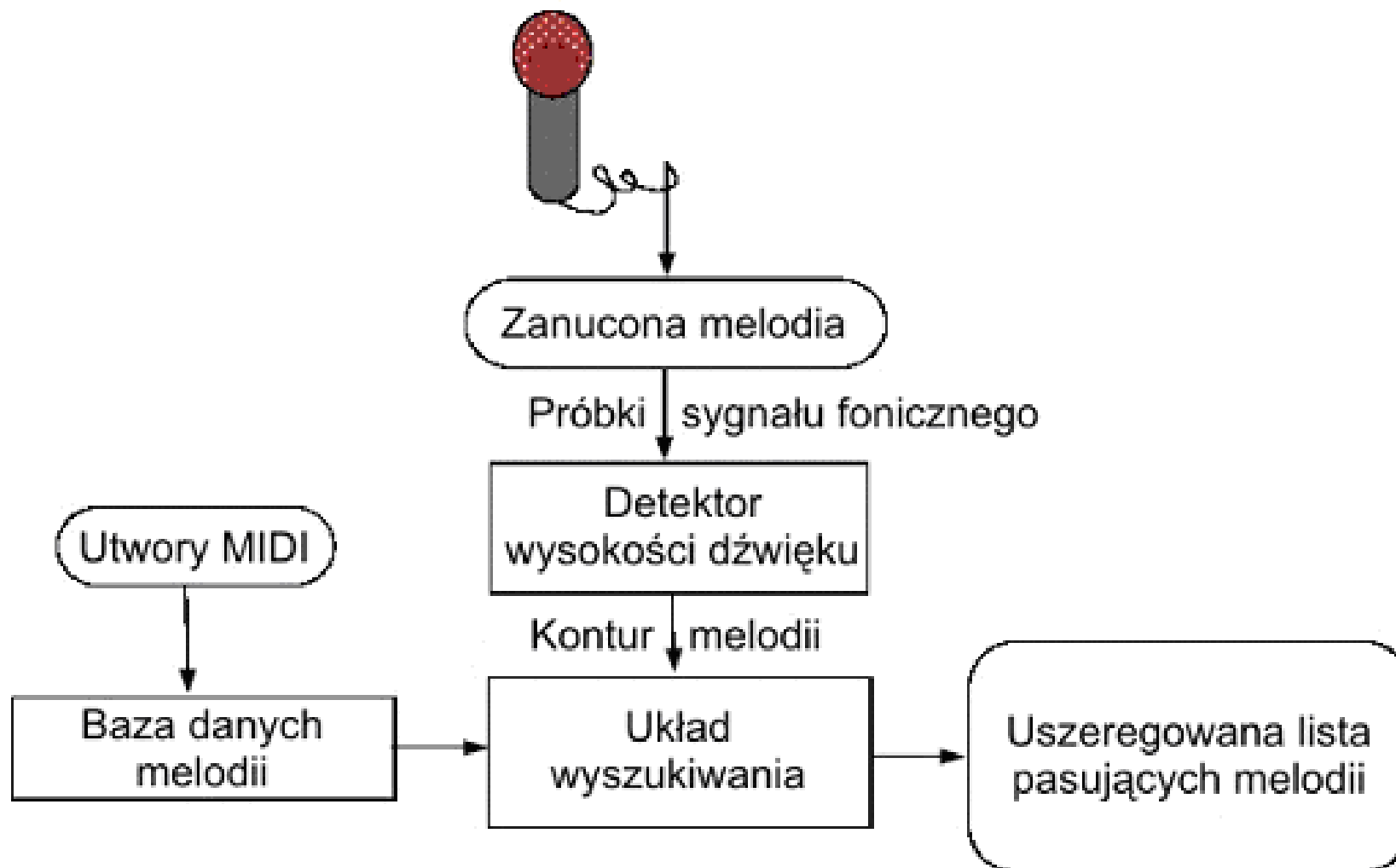
QBH – *Query-by-Humming* („zapytanie przez zanucenie melodii”)

- użytkownik nuci lub gwizdże do mikrofonu melodię,
- algorytm **śledzenia wysokości dźwięku** (*pitch tracking*) zamienia melodię np. na **kontur melodyczny**,
- moduł wyszukiwający porównuje kontur melodyczny uzyskany z zapytania z konturami zapisanymi w bazie, znajduje najbardziej podobne obiekty.

# Schemat systemu QBH

---

Ghias *et al.*, 1995





# Kontur melodyczny

---

Najprostszy opis: **kontur melodyczny** jest zapisywany przy pomocy **kodu Parsons**.

Zapisywana jest tylko informacja o wysokości każdej nuty względem poprzedniej:

**U** – wyższa, **D** – niższa, **R** (lub **S**) – taka sama.

Przykładowy kod: \*UURRDUDDDDDRUDUD



The image shows a musical staff in 3/4 time with a treble clef and a key signature of one flat. The melody is written with notes and rests, with a star symbol at the beginning. Below the staff, the corresponding Parsons code is written: \* D U U D D D U R D R U U U U. The code consists of 14 characters: a star, followed by D, U, U, D, D, D, U, R, D, R, U, U, U, U.

# Kontur melodyczny

---

Zakłada się, że kod Parsonsa dla danej melodii jest unikalny. Kod jest nieczuły na:

- drobne zafałszowania przy nuceniu melodii,
- błędy rytmiczne (czasy trwania nut).

Mogą jednak wystąpić błędy, które należy brać pod uwagę podczas wyszukiwania:

Tekst	casablanca	casablanca	casablanca
Struktura	sbbla	s bla	saabla
	Błąd transpozycji	Błąd zaniku	Błąd powielenia

# Rozszerzenia systemu QBH

---

Bardziej zaawansowane systemy QBH używają do wyszukiwania informacji o:

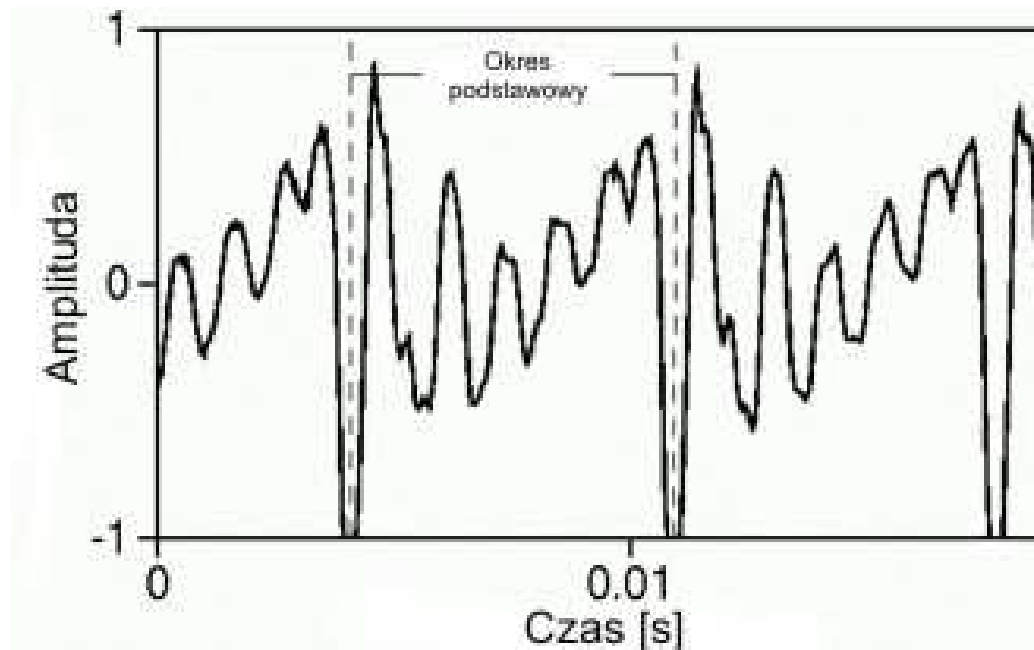
- bezwzględnych wysokościach nut
- czasie trwania poszczególnych nut

Detekcja wysokości nut może wykorzystywać różne algorytmy (autokorelacja, liczenie przejść przez zero, FFT, itp.).

# Detekcja wysokości nut (przykład)

---

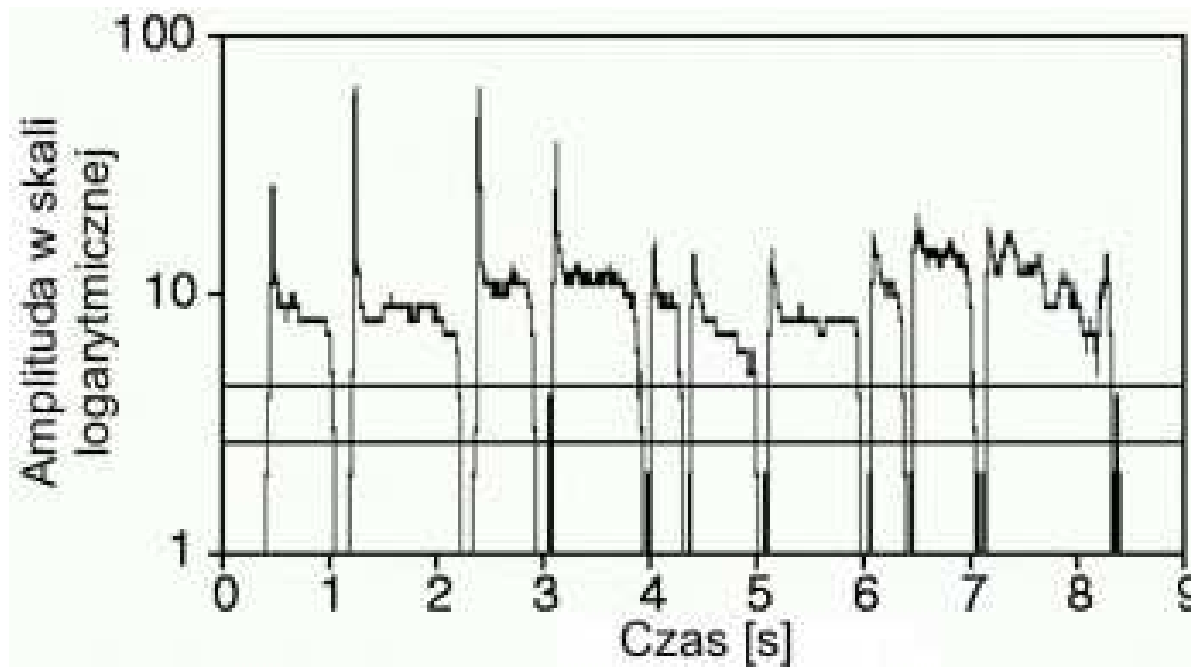
- Sygnał jest przetwarzany przez filtr dolnoprzepustowy – ogr. pasma do 1 kHz.
- Detektor wyznacza w przetworzonym sygnale powtarzający się okres podstawowy
- Analiza w ramkach o dł. ok. 20 ms.



# Detekcja czasu trwania nut (przykład)

---

- Użytkownik nucąc melodię wyraźnie rozdziela każdą nutę („na na na na...”).
- Gwałtowny wzrost, a następnie spadek amplitudy sygnału (trwający ok. 60 ms).
- Wartości progowe amplitudy pozwalają wyznaczyć początek i koniec każdej nuty.



# QBH a QBW

---

Czasami rozróżnia się dwa typy QBH:

- właściwe QBH – zapytanie przez „zanucenie”
- *Query by whistling* – zapytanie przez zagwizdanie melodii

Oba typy wykorzystują te same algorytmy.

QBW w porównaniu do QBH:

- znacznie prostsza analiza (gwizdanie produkuje wielotony łatwe do analizy)
- trudniej jest podać melodię bez zafałszowań

# Query by rhythm

---

- QBR (*Query by Rhythm*) to metoda, w której podaje się kontur rytmiczny, np. przez wystukanie rytmu na klawiaturze komputerowej.
- Jest mało dokładna, rytm rzadko identyfikuje jednoznacznie utwór, trudno dokładnie podać rytm utworu.
- Metoda raczej pomocnicza, stosowana wraz z innymi metodami.

# Wyszukiwanie danych

---

Zadanie dla algorytmu wyszukiującego:

- wyszukać wystąpienia wzorca  
 $P = p_1 p_2 p_3 \dots p_m$
- w ciągach tekstowych  $T = t_1 t_2 t_3 \dots t_n$
- przy założeniu maksimum  $k$  różnic

Baza zwraca listę znalezionych utworów uszeregowanych wg podobieństwa do zapytania.

Algorytmy wyszukiwania:

- obliczanie odległości ciągów
- drzewa binarne i inne algorytmy



# Optymalizacje wyszukiwania

---

Przyspieszenie wyszukiwania:

- wybieramy losowo z bazy  $k$  obiektów
  - punktów węzłowych, liczymy ich odległość od każdego obiektu w bazie
- każdy obiekt jest przypisany do najbliższego punktu węzłowego
- liczymy odległość szukanego ciągu od  $k$  punktów węzłowych
- wybieramy najbliższy punkt węzłowy i liczymy odległość tylko od jego punktów potomnych

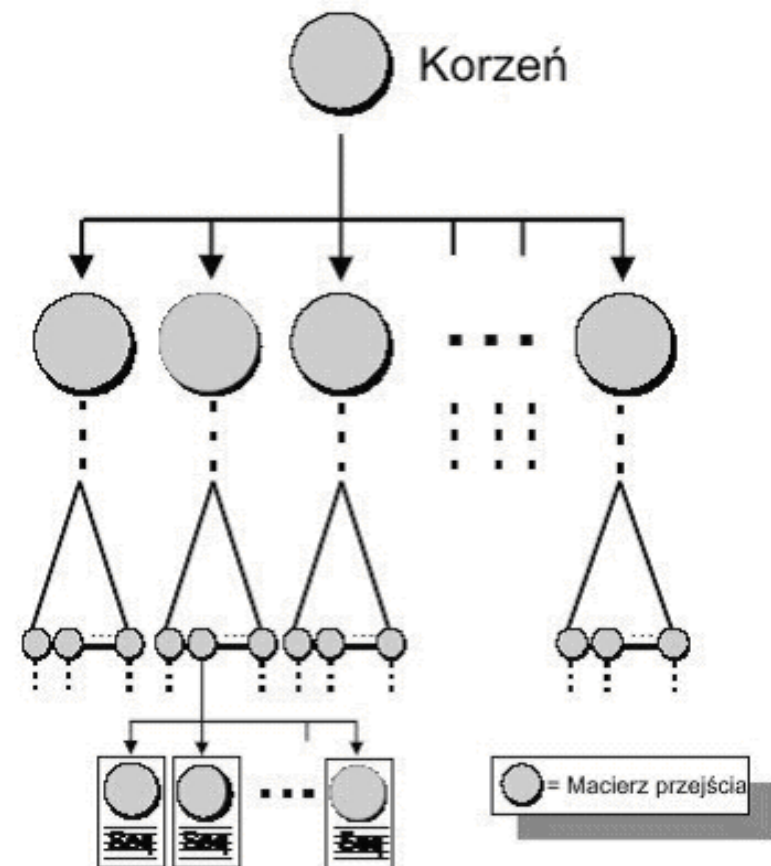
# Optymalizacje wyszukiwania

---

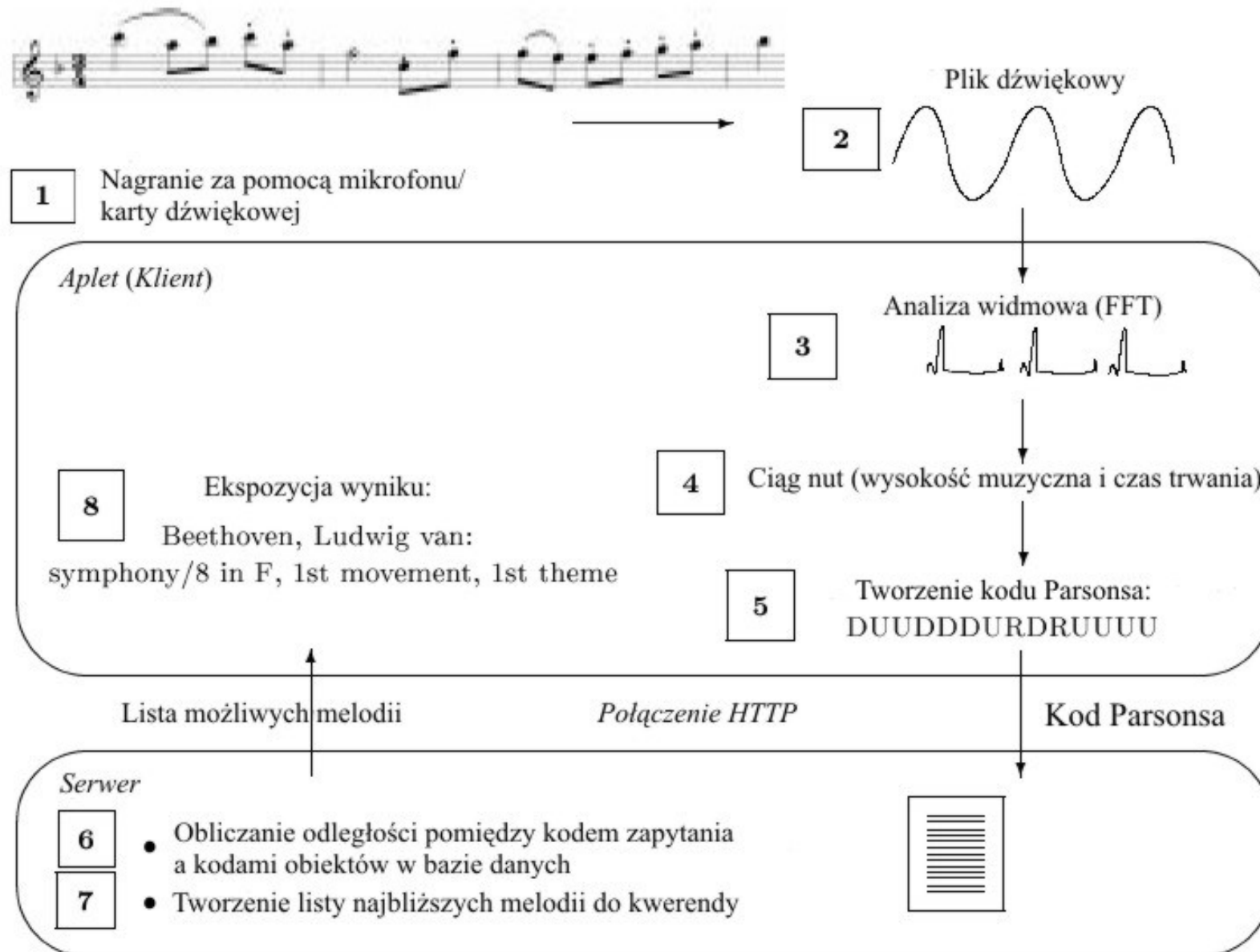
Przy dużych zbiorach danych można powtórzyć tę operację dla mniejszych grup ciągów.

Tworzy się w ten sposób struktura drzewiasta.

Na każdym poziomie drzewa – wybór potomka z najmniejszą odległością.



# Przykład: system Musipedia



# Przykład: system Musipedia

---

## Panel do wprowadzenia zapytania

The image shows a software interface for entering a musical query. At the top, there are two tabs: "Symbolic Input" (selected) and "Raw Audio Input". Below the tabs is a piano keyboard graphic with 12 keys. Underneath the keyboard is a button labeled "Add a rest".

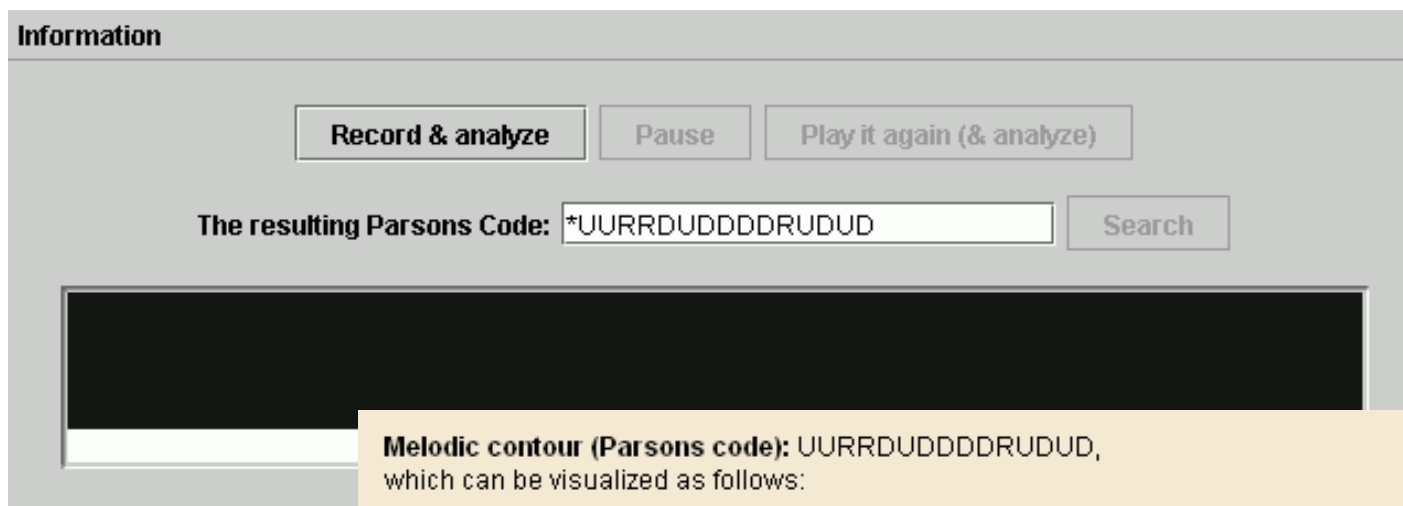
The "Duration" section features five icons representing different note values: eighth note, quarter note, half note, dotted half note, and whole note. The half note icon is currently selected. Below these icons is a horizontal slider with a scale from 0 to 8. The slider is positioned at 1.0, and a text box to the right of the slider displays the value "1.0".

At the bottom of the interface, there is a status bar that reads "Information: Ready to record". Below this are several control buttons: a red record button, a play button, a stop button, a delete button, a folder icon, and a floppy disk icon. A large "Begin Search" button is located at the bottom right of the panel.

# Przykład: system Musipedia

---

## Kod Parsons'a i zwrócony wynik wyszukiwania



**Melodic contour (Parsons code):** UURRDUDDDDRUDUD,  
which can be visualized as follows:



Not happy with the search result? [Refine your search here.](#) Or ask other users for help by using the [Forum.](#)

**1 exact match found:**

[A] national anthem of Poland

[view details, edit, or delete this entry](#)

# Musipedia - parametryzacja

---

## Parametryzacja:

- podział sygnału na ramki (46 ms, zakładkowanie 50%)
- analiza widmowa każdej ramki (FFT) – decyzja:
  - sygnał – częstotliwość i amplituda maksimum
  - cisza
- ramki zawierające sygnał są łączone w nuty, rozdzielone ciszą lub gwałtowną zmianą częstotliwości
- częstotliwości nut zamieniane są na kod Parsonsa.

Parametry analizy mogą być ustawiane przez użytkownika.

# Musipedia - wyszukiwanie

---

Wyszukiwanie danych w systemie Musipedia:

- obliczanie odległości między kodem Parsonsa szukanego nagrania a wszystkimi kodami zapisanymi w bazie danych
- miara odległości – ważona suma minimalnej liczby przekształceń kodu (wstawień, zamiany i usunięć znaków) potrzebnej do dokładnego dopasowania
- zwracana jest lista „najbliższych” elementów
- podawane są również informacje dodatkowe o utworze, jeżeli zostały wprowadzone do bazy (np. zapis nutowy, odnośnik do sklepu, itp.)

# Musipedia - skuteczność

---

Skuteczność systemu Musipedia oceniana za pomocą zbioru testowego, przy gwizdaniu melodii:

- przy braku zakłóceń w sygnale wejściowym uzyskuje się średnią liczbę poprawnych odpowiedzi 4 na 5
- szum pochodzący od oddechu ma największy wpływ na skuteczność (szum ten jest filtrowany, parametry filtracji mogą być regulowane przez użytkownika)
- liczba nut mniejsza niż 8 znacząco pogarsza skuteczność
- najbardziej podatne na błędy w kodzie Parsons'a są elementy R (taka sama wysokość)
- najczęstsze zniekształcenia w kodzie Parsons'a to kody wstawienia (nieistniejące nuty)
- skuteczność zależy też od muzyki (uzyskano większą skuteczność dla muzyki klasycznej)



# Midomi / SoundHound

---

Midomi – obecnie część systemu *SoundHound* ([www.soundhound.com](http://www.soundhound.com))

- jedyny komercyjny system wykorzystujący technologię QBH (nucenie, śpiewanie)
- oprócz tego umożliwia wyszukiwanie według przykładu oraz przez rozpoznawanie głosu (wypowiedzenie tytułu lub wykonawcy)
- baza QBH w 100% opracowana przez użytkowników
- technologia wyszukiwania nosi nazwę *Sound2Sound*
- aplikacje klienta dla urządzeń mobilnych

# Midomi / SoundHound

---

Technologia rozpoznawania muzyki wykorzystuje m.in. informacje o:

- zmianach wysokości dźwięku,
- rytmie,
- położeniu pauz,
- zawartości fonetycznej,
- treści mowy.

Dane są wykorzystywane w zależności od typu zapytania. Np. treść mowy jest wykorzystywana przy śpiewaniu, a nie jest wykorzystywana przy nuceniu.

Wyszukiwanie jest niezależne od tonacji, tempa, języka i (do pewnego stopnia) jakości śpiewu.

# Systemy MIR audio

---

Drugą grupę systemów MIR stanowią systemy, w których parametryzuje się:

- pliki dźwiękowe (np. mp3)
- strumień audio (np. z radia „na żywo”)

Systemy tego typu nazywa się czasami **QBE** (ang. *Query by Example*

– zapytanie przez przykład).

Parametryzacja jest trudniejsza niż w QBH.

# Philips Audio Fingerprinting

---

- *Philips Audio Fingerprinting Technology*
  - algorytm opracowany przez firmę Philips, służący do identyfikacji nagrań muzycznych :
    - przesyłanych w postaci strumienia (*on-air*),
    - przesłanych w postaci pliku
- Technologia komercyjna, dostarczana jako zestaw procedur (API) do zaimplementowania w oprogramowaniu klienta.
- System „klient-serwer”.
- Nie jest znana dokładna struktura algorytmów parametryzujących i wyszukujących dane.

# Philips Audio Fingerprinting

---

- Oprogramowanie po stronie klienta oblicza sygnaturę (*fingerprint*, „odcisk palca”):
  - *sub-fingerprints* – obliczone na podstawie krótkich ramek czasowych (kilka ms)
  - *fingerprint blocks* – sygnatury złożone z 256 *sub-fingerprints* (ok. 3 sek.)
- *Fingerprint-blocks* są przesyłane do serwera, który dokonuje ich identyfikacji.
- Serwer przesyła identyfikator utworu (*Song ID*) oraz pozycję wewnątrz pliku, odpowiadającą sygnaturze.

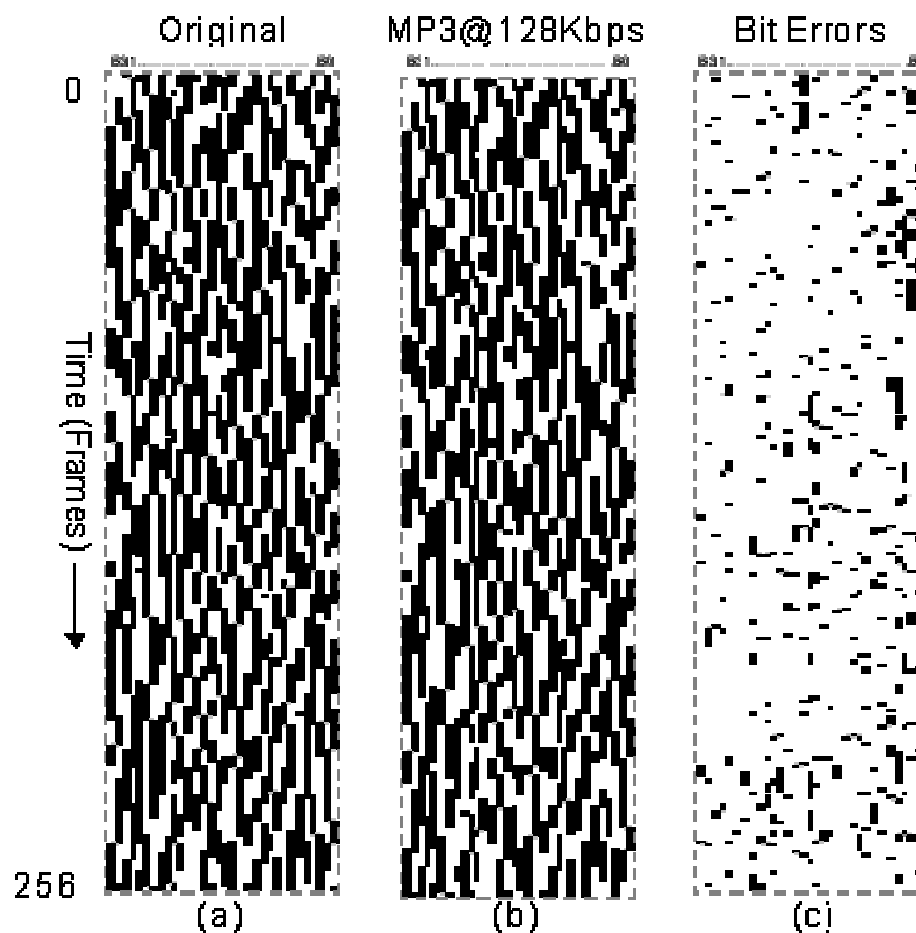
# Philips Audio Fingerprinting

---

Według autorów, system jest niewrażliwy na:

- zmniejszanie przepływności do 64 kbit/s,
- filtrację,
- dodawanie echa,
- przepróbkowanie,
- transpozycję,
- zaszumienie.

Wystarczy fragment o długości 3 s.



# AcoustID / MusicBrainz

---

- AcoustID – system rozpoznawania muzyki, opracowany na licencji Open Source.
- Adres: [acoustid.org](http://acoustid.org)
- Wykorzystuje algorytm parametryzacji o nazwie *Chromaprint*.
- Jest wykorzystywany m.in. w systemie *MusicBrainz* ([www.musicbrainz.org](http://www.musicbrainz.org)) do opisywania (tagowania) plików muzycznych na podstawie zawartości.

# Chromaprint

---

Krótki opis algorytmu:

- analizowane są pierwsze 2 minuty utworu,
- obliczenie widma (FFT),
- chromagram - analiza prowadzona dla 12 zakresów wysokości (*pitch classes*)
- zapis parametrów 8 razy na sekundę dla każdego zakresu
- postprocessing – usunięcie nadmiarowych danych przy zachowaniu wzorca



# Chromaprint

---

## Bardziej szczegółowy opis

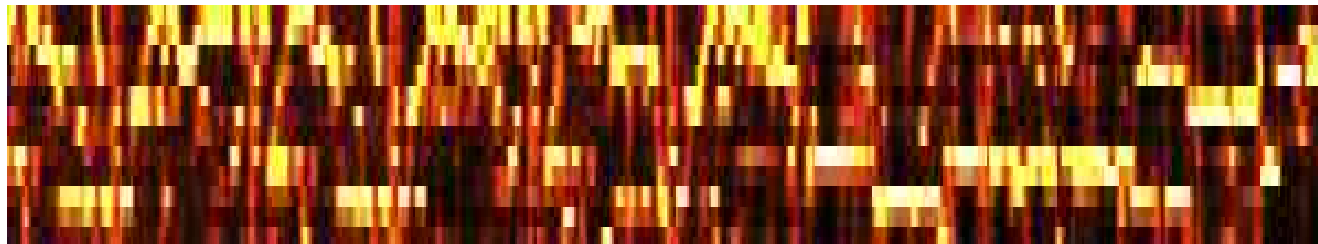
(na podstawie: <http://oxygene.sk/2011/01/how-does-chromaprint-work/>)



Postać czasowa



Spektrogram



Chromagram

# Chromaprint

---

Wyniki analizy – wykresy chromagramów uzyskane dla poszczególnych okien analizy, są parametryzowane za pomocą filtrów graficznych:



- 16 filtrów
- każdy daje wynik w postaci liczby od 0 do 3
- wynik zapisywany na dwóch bitach
- sumaryczny wynik: liczba 32-bitowa

Zbiór tych liczb dla kolejnych okien analizy stanowi wzorzec (*fingerprint*)

# Chromaprint - przykład

---



Heaven FLAC



Heaven 32kbps MP3



Differences between Heaven FLAC and Heaven 32kbps MP3

---

Pierwszy utwór



Under The Ice FLAC



Under The Ice 32kbps MP3



Differences between Under The Ice FLAC and Under The Ice 32kbps MP3

---

Drugi utwór



Differences between Heaven FLAC and Under The Ice FLAC

---

Różnica obu utworów

# Shazam

---

Shazam ([www.shazam.com](http://www.shazam.com)) – przykład popularnego, komercyjnego systemu typu *Query by Mobile Phone*

- aplikacje klienckie dla większości używanych systemów mobilnych
- strumień audio rejestrowany przez mikrofon
- według autorów, wystarcza nagranie o długości 1 sekundy (w praktyce do 15 s.)
- obliczony wzorzec przesyłany jest do serwera
- wyniki: dane o utworze, odnośniki do sklepów, informacje o wykonawcy, itp.

# Shazam - zasada działania

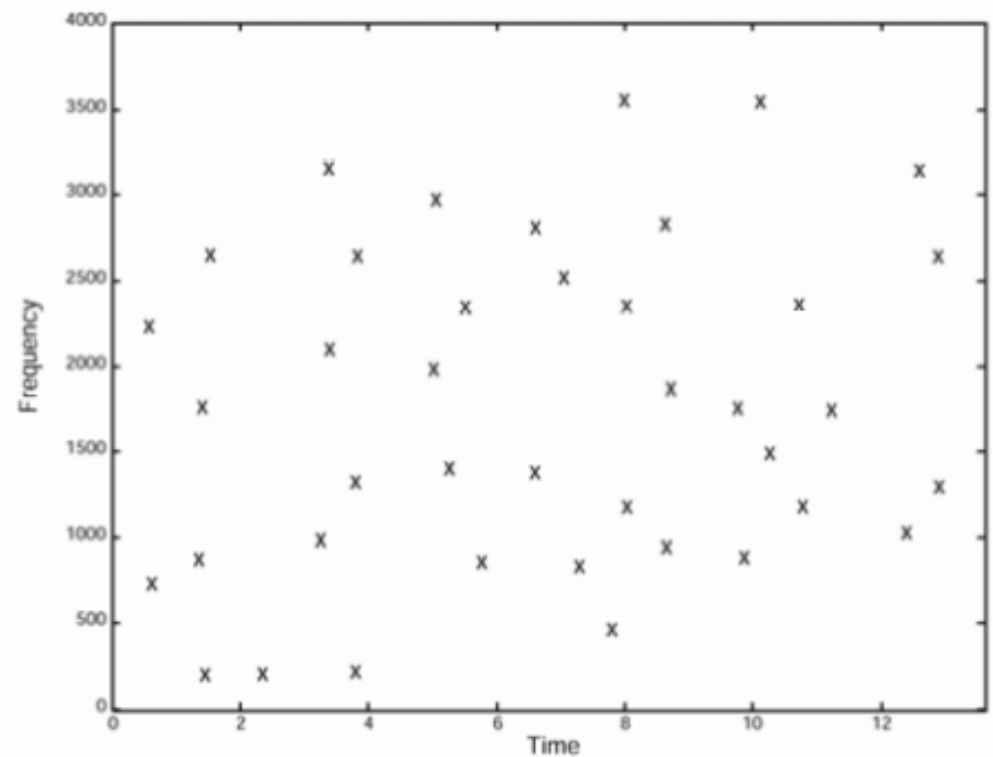
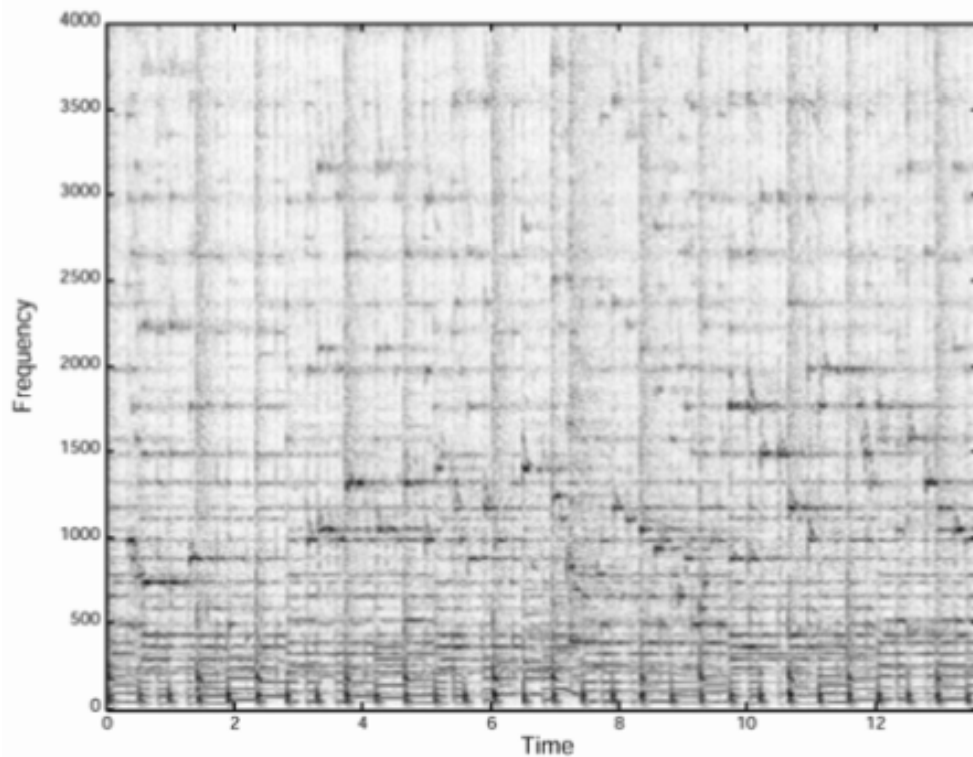
---

## Opis działania systemu Shazam w roku 2011

(na podstawie: <http://www.soyoucode.com/2011/how-does-shazam-recognize-song>)

### Sposób wyznaczania wzorca:

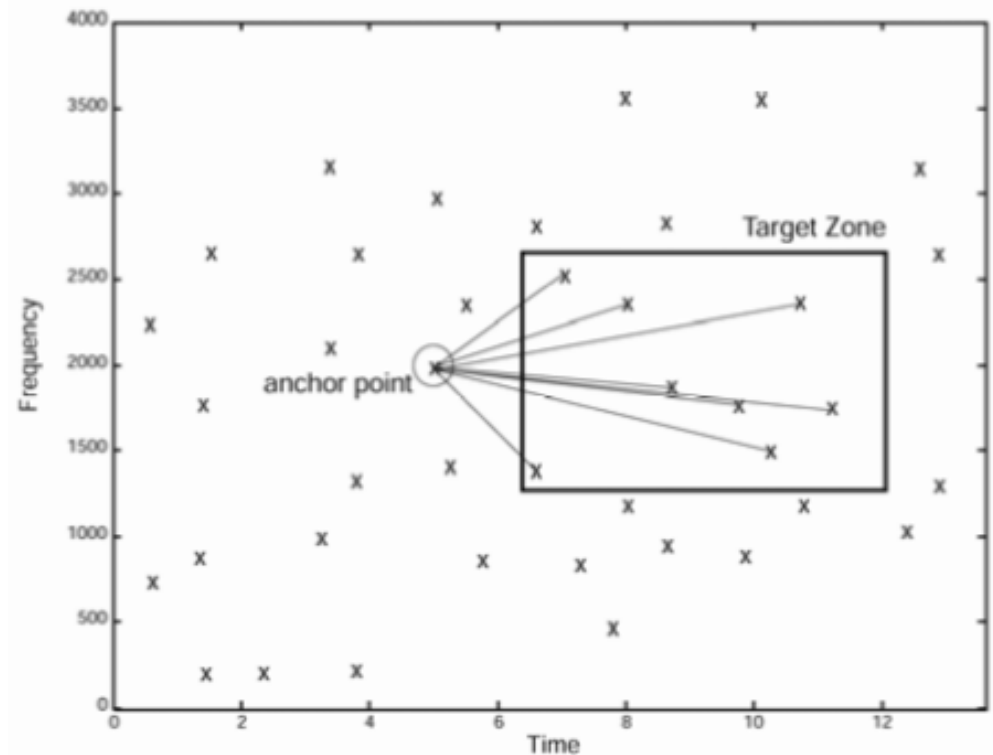
- obliczenie spektrogramu
- wyznaczenie dominujących składowych



# Shazam - zasada działania (cd.)

---

- Wybierane są punkty (*anchor points*) i strefy w ich pobliżu (*target zones*)
- Obliczane są odległości między punktem *anchor* i każdym z punktów w strefie
- Odległość zapisywana jako *hash*, np. punkty  $(t_1, f_1)$  i  $(t_2, f_2)$   
hash =  $(f_1 + f_2 + (t_2 - t_1)) + t_1$
- Wszystkie hashe zapisywane we wzorcu

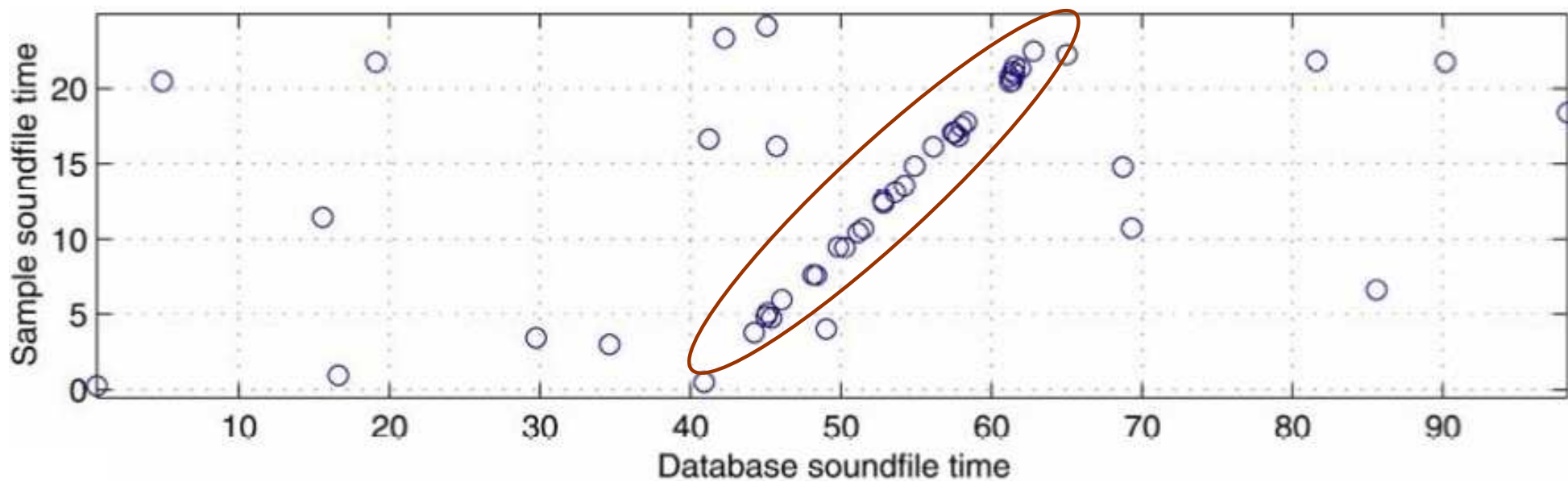


## Shazam - zasada działania (cd.)

---

### Wyszukiwanie:

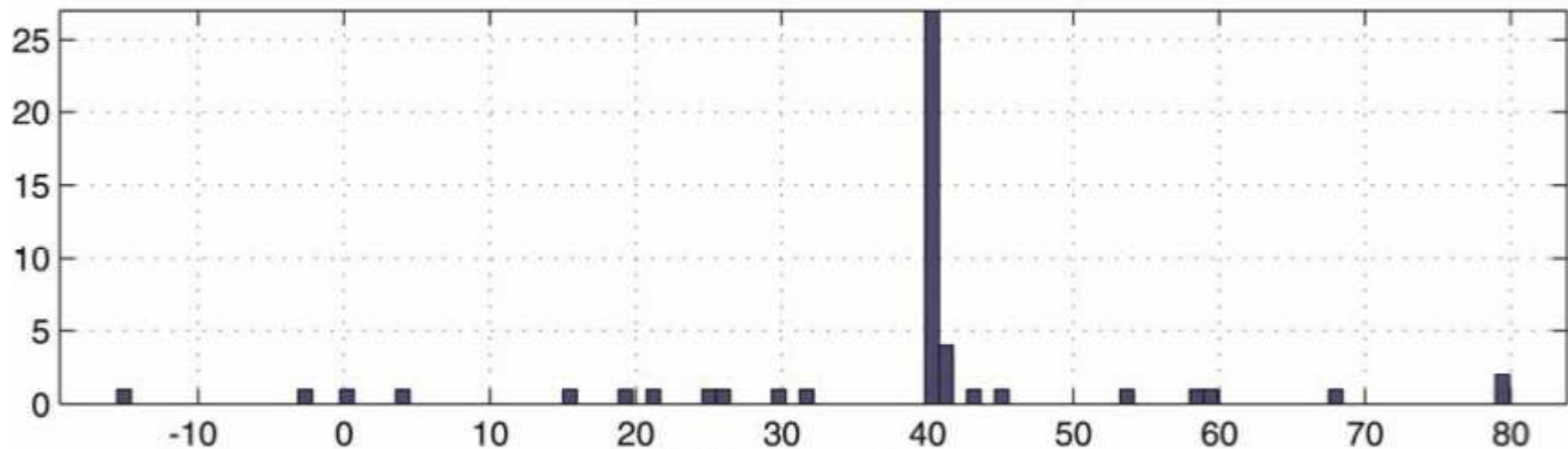
- obliczenie wzorca dla wyszukiwanego utworu
- znalezienie pasujących hashów z obu wzorców
- zaznaczenie na wykresie (*scatter graph*) czasu wystąpienia dopasowania
- ciąg dopasowań tworzących linię prostą oznacza znalezienie dopasowania



## Shazam - zasada działania (cd.)

---

- Różnice czasu wystąpienia dopasowania są zaznaczano na histogramie.
- Wysoki słupek histogramu = stała różnica, zatem mamy dopasowanie utworu.



Opis na podstawie: <http://www.soyoucode.com/2011/how-does-shazam-recognize-song>