Informatyka, studia dzienne, mgr II st. semestr I

Przetwarzanie obrazu i dźwięku **2011/2012**
Prowadzący: dr inż. Bartłomiej Stasiak czwartek, 8:30

Data oddania: _____ Ocena: _____

Grzegorz Graczyk 178717
Andrzej Stasiak 178736

Zadanie 3: Analiza częstotliwości podstawowej dźwięku*

Cel

Celem zadania było napisanie aplikacji wykrywającej częstotliwość w nagraniu dźwiękowym w postaci pliku WAVE. Aplikacja miała umożliwiać rozpoznanie częstoliwości zarówno w plikach zawierających pojedynczy dźwięk, jak i sekwencję dźwięków. Do rozpoznania częstotliwości użyto następujących metod.

- Metoda w dziedzinie czasu: Autokorelacja (T3)
- Metoda w dziedzinie częstotliwości: Analiza cepstralna (F2)

Wstęp teoretyczny

Autokorelacja

Metoda ta polega na znalezieniu pierwszego maksimum funkcji:

$$c(m) = \sum_{n=0}^{N-1} x(n)x(n+m)$$

Należy jednak zauważyć, że taka funkcja zawsze posiada maksimum w punkcie $m = 0$, który jednak odrzucamy. Należy odrzucić również punkty

* SVN: `https://serce.ics.p.lodz.pl/svn/labs/poid/bs_czw0830/idgrupy/zadanie3`

bliskie 0, których definicja nie jest jednak jednoznaczna. Odnaleziona wartość m jest okresem (wyrażonym w próbkach) badanego dźwięku.

Algorytm szukania pierwszego maksimum zaprojektowano następująco:

1. Zapamiętanie wartości w punkcie $m = 0$ jako y_{max}.
2. Zwiększanie m zapamiętując najmniejszą napotkaną wartość jako y_{min}
3. Powtarzanie powyższej czynności tak długo jak:

$$y_m - y_{min} < 0.1 \cdot (y_{max} - y_{min})$$

4. Iteracyjne odnalezienie pierwszego maksimum dla wartości nie mniejszych niż m znalezione powyższą metodą.

Analiza cepstralna

Analiza cepstralna jest jedną z metod analizy w dziedzinie częstotliwości. Cepstrum jest to widmo widma amplitudowego sygnału. Zostaje uzyskane poprzez transformację (np. Fouriera) przeprowadzoną na widmie amplitudowym sygnału. Ponieważ widmo sygnału jest zasadniczo okresowe (częstotliwość podstawowa i wyższe alikwoty), to maksimum cepstrum odpowiada częstotliwości podstawowej sygnału.

W opisanym zadaniu analiza cepstralna została wykonana przy użyciu transformacji Fouriera widma dźwięku. W celu ułatwienia ekstrakcji składowych widma, przed przekształceniem z dziedziny czasu do dziedziny częstotliwości dany fragment sygnału zostaje poddany operacji okienkowania, przy użyciu okna Hanninga. Po przejściu do dziedziny częstotliwości na widmie zostaje wykonanych szereg operacji mających na celu uproszczenie jego analizy. Widmo zostaje obcięte w połowie, ze względu na jego nadmiarową symetrię. Odrzucona zostaje informacja o fazie, więc dalsza analiza zostaje przeprowadzona wyłącznie dla widma amplitudowego. Trzecią operacją jest progowanie, usuwające z widma słabe składowe nie będące wielokrotnością częstotliwości podstawowej. W celu poprawy wyników tej operacji i dodatkowego oczyszczenia widma odrzucone zostają elementy niebędące lokalnymi maksimami.

Analiza cepstrum widma ma na celu określenie częstotliwości podstawowej widma. Położenie maksimum tego widma odpowiada częstotliwości podstawowej - jest jego odwrotnością (przy uwzględnieniu liczby próbek). Dlatego znając położenie tego maksimum możemy określić częstotliwość podstawową analizowanego dźwięku przy użyciu wzoru:

$$f = \frac{l_s z}{wm}$$

,gdzie l_s - liczba elementów widma (na którym wykonujemy drugą transformację), z - częstotliwość próbkowania dźwięku, w - szerokość okna analizy (liczba próbek), m - położenie maksimum cepstrum.

Wyszukiwanie dźwięków w sekwencji

Wyszukiwanie dźwięków w sekwencji zrealizowano jedynie poprzez analizę amplitudy, a nie częstotliwości — pozwala to wykryć następujące po sobie identyczne dźwięki oraz ogranicza użycie kosztownych obliczeniowo metod analizy częstotliwości. Zaimplementowana metoda została zaprojektowana do analizy dźwięku fortepianu. W wypadku skrzypiec sekwencje dźwięków wykonywane są w sposób ciągły co uniemożliwa wykrycie niektórych dźwięków oraz rozbicie jednego dźwięku na kilka.

Algorytm wykonuje operacje na kolejnych próbkach trwających $\frac{1}{20}$ sekundy (pobierane na zakładkę co $\frac{1}{40}$ sekundy). Dla każdej próbki wyznaczana jest amplituda:

1. Amplituda pierwszej próbki (a_0) jest zapamiętywana jako a_{top} oraz a_{bottom}.
2. Pobieramy kolejne próbki:
   a) Jeśli $a_i > a_{top}$ to $a_{top} = a$ oraz $i_{begin} = i$
   b) Jeśli $a_{top} - a_i > 0.2 \cdot (a_{top} - a_{bottom})$ to $a_{bottom} = a_i$, $i_{end} = i$ oraz przechodzimy do kroku 3.
3. Pobieramy kolejne próbki:
   a) Jeśli $a_i < a_{bottom}$ to $a_{bottom} = a$ oraz $i_{end} = i$
   b) Jeśli $a_i - a_{bottom} > 0.2 \cdot (a_{top} - a_{bottom})$ - idź do kroku 3.
4. Para (i_{begin}, i_{end}) opisuje część sygnału, która prawdopodobnie zawiera dźwięk. Dźwięk zapamiętujemy i wracamy do punktu 2.

Dla wyżej opisanego algorytmu warto wprowadzić dodatkowe kryteria — ograniczając zapamiętane dźwięki jedynie do takich o słyszalnej częstotliwości, amplitudzie oraz czasie trwania.

Implementacja

Jako rozwiązanie przygotowano aplikację w języku Python. Aplikacja posiada dwa tryby pracy: tryb dla wielu plików, w którym dla kolejnych plików odnajdywana jest częstotliwość dźwięku w nich zawartych oraz tryb dla jednego pliku, w którym następuje próba odnalezienia sekwencji dźwięków, a także generowane są dodatkowe informacje o zawartości pliku.

W implementacji wykorzystano między innymi biblioteki **wave** do wczytywania plików WAVE oraz **numpy** do optymalnego wyznaczenia transformaty Fouriera.

Funkcja dokonująca obliczenia częstotliwości bazowej na podstawie przekazanej tablicy próbek wykonuje kolejno opisane działania: okienkowanie Hanninga, wyliczenie widma amplitudowego za pomocą FFT, odrzucenie górnej połowy widma, progowanie widma, odrzucenie składowych niebędących maksimami lokalnymi, wyliczenie cepstrum i znalezienie maksimum tego cepstrum.

W celu poprawy dokładności zastosowano metodę polegającą na znalezieniu n-tego maksimum i podzieleniu jego pozycji przez n - liczbę maksimów. Dzięki temu położenie pierwszego maksimum można określić z mniejszym błędem.

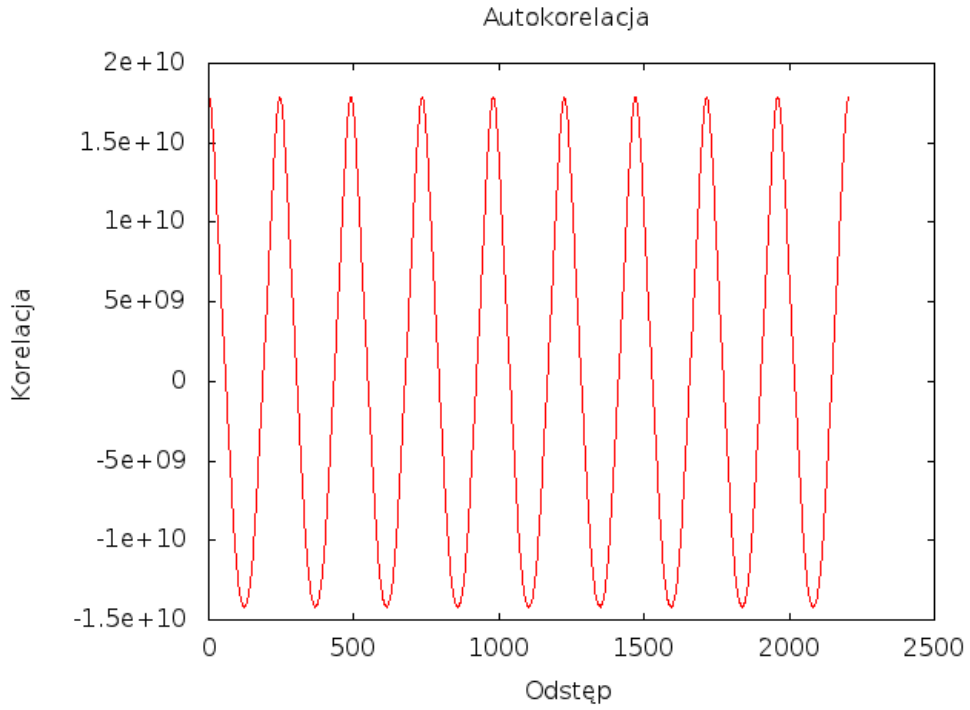
Wyniki

Dźwięki

Badany plik	Autokorelacja	Cepstrum
artificial/diff/120Hz.wav	120 Hz	120 Hz
artificial/diff/1366Hz.wav	1378 Hz	1356 Hz
artificial/diff/180Hz.wav	90 Hz	180 Hz
artificial/diff/270Hz.wav	271 Hz	270 Hz
artificial/diff/405Hz.wav	405 Hz	411 Hz
artificial/diff/607Hz.wav	604 Hz	600 Hz
artificial/diff/80Hz.wav	80 Hz	80 Hz
artificial/diff/911Hz.wav	919 Hz	909 Hz
artificial/easy/100Hz.wav	100.0 Hz	100 Hz
artificial/easy/1139Hz.wav	1131 Hz	1138 Hz
artificial/easy/150Hz.wav	150 Hz	160 Hz
artificial/easy/1708Hz.wav	1696 Hz	1695 Hz
artificial/easy/225Hz.wav	223 Hz	220 Hz
artificial/easy/337Hz.wav	339 Hz	340 Hz
artificial/easy/506Hz.wav	501 Hz	499 Hz
artificial/easy/759Hz.wav	760 Hz	760 Hz
artificial/med/1025Hz.wav	1025 Hz	1019 Hz
artificial/med/135Hz.wav	135 Hz	140 Hz
artificial/med/1537Hz.wav	1521 Hz	1547 Hz
artificial/med/202Hz.wav	200 Hz	200 Hz
artificial/med/303Hz.wav	302 Hz	300 Hz
artificial/med/455Hz.wav	455 Hz	460 Hz
artificial/med/683Hz.wav	678 Hz	681 Hz
artificial/med/90Hz.wav	90 Hz	100 Hz
natural/flute/1265Hz.wav	1260 Hz	1259 Hz
natural/flute/1779Hz.wav	1764 Hz	1781 Hz
natural/flute/276Hz.wav	276 Hz	276 Hz
natural/flute/443Hz.wav	441 Hz	440 Hz
natural/flute/591Hz.wav	588 Hz	592 Hz
natural/flute/887Hz.wav	882 Hz	882 Hz
natural/viola/130Hz.wav	131 Hz	131 Hz
natural/viola/196Hz.wav	196 Hz	196 Hz
natural/viola/247Hz.wav	246 Hz	247 Hz
natural/viola/294Hz.wav	294 Hz	294 Hz
natural/viola/369Hz.wav	368 Hz	368 Hz
natural/viola/440Hz.wav	441 Hz	1268 Hz
natural/viola/698Hz.wav	689 Hz	691 Hz

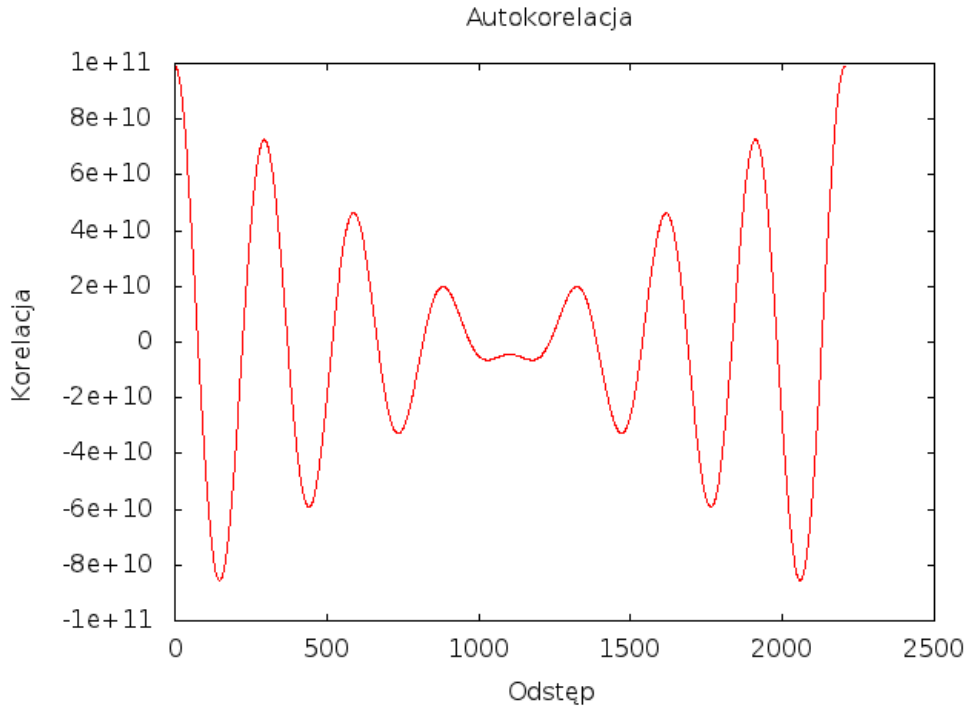
Jak widać rozpoznanie większości dźwięków się powiodło. Dla metody w dziedzinie czasu widoczne jest jedno nieprawidłowe rozpoznanie w którym znaleziono częstotliwość dwa razy mniejszą, a dla metody w dziedzinie częstotliwości jedną prawie trzy razy większą. W pozostałych przypadkach dokładność jest na poziomie kilku herców.

Dla błędnego pliku wykres korelacji przedstawia się następująco:



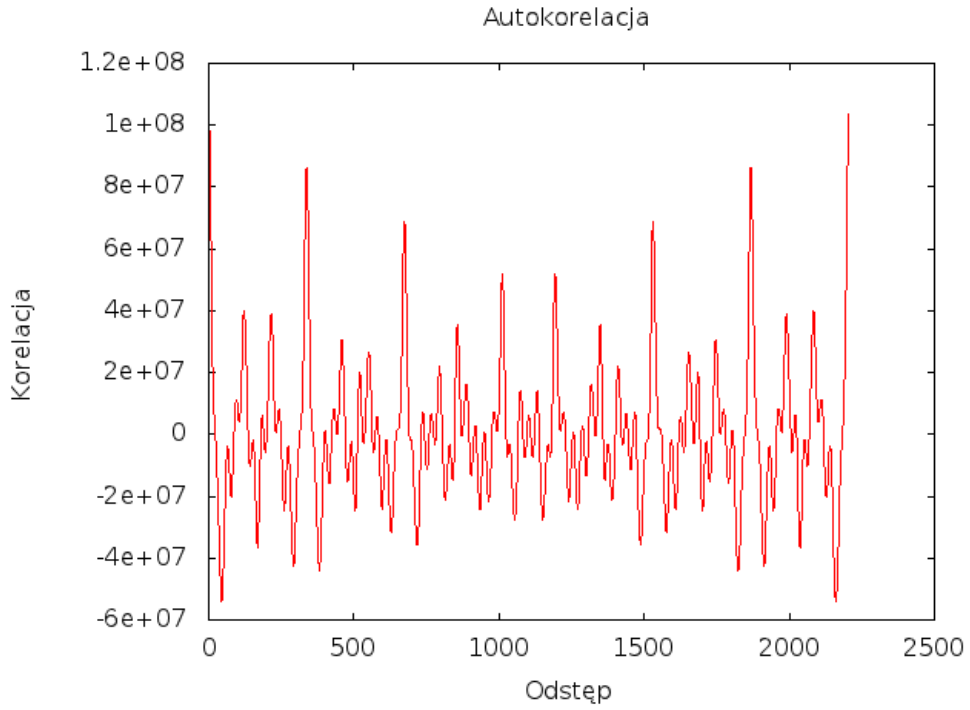
Wartości korelacji są bardzo podobne. Zwiększenie rozmiaru próbki pozwoliło poprawnie rozpoznać częstotliwość.

Dla `artificial/easy/150Hz.wav`:



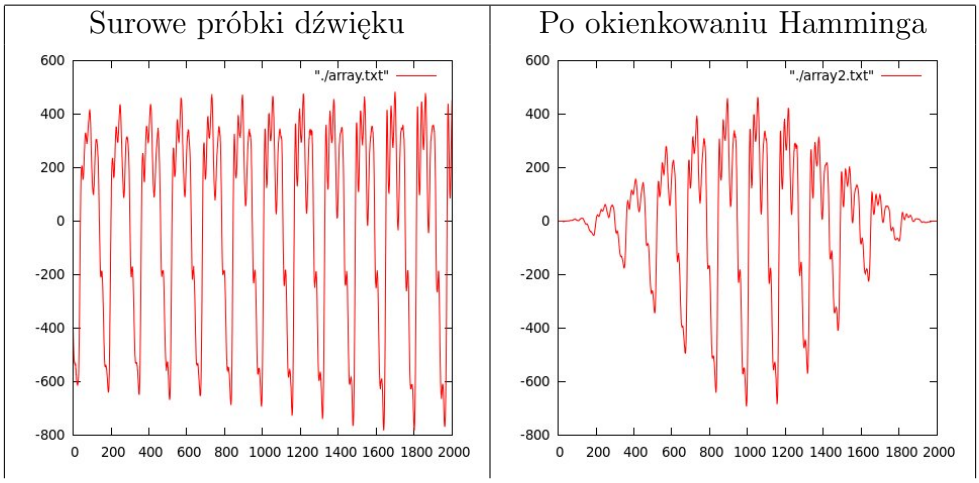
Dla człowieka odpowiedź jest oczywista, choć w wypadku programu określenie tego maksimum było problematyczne.

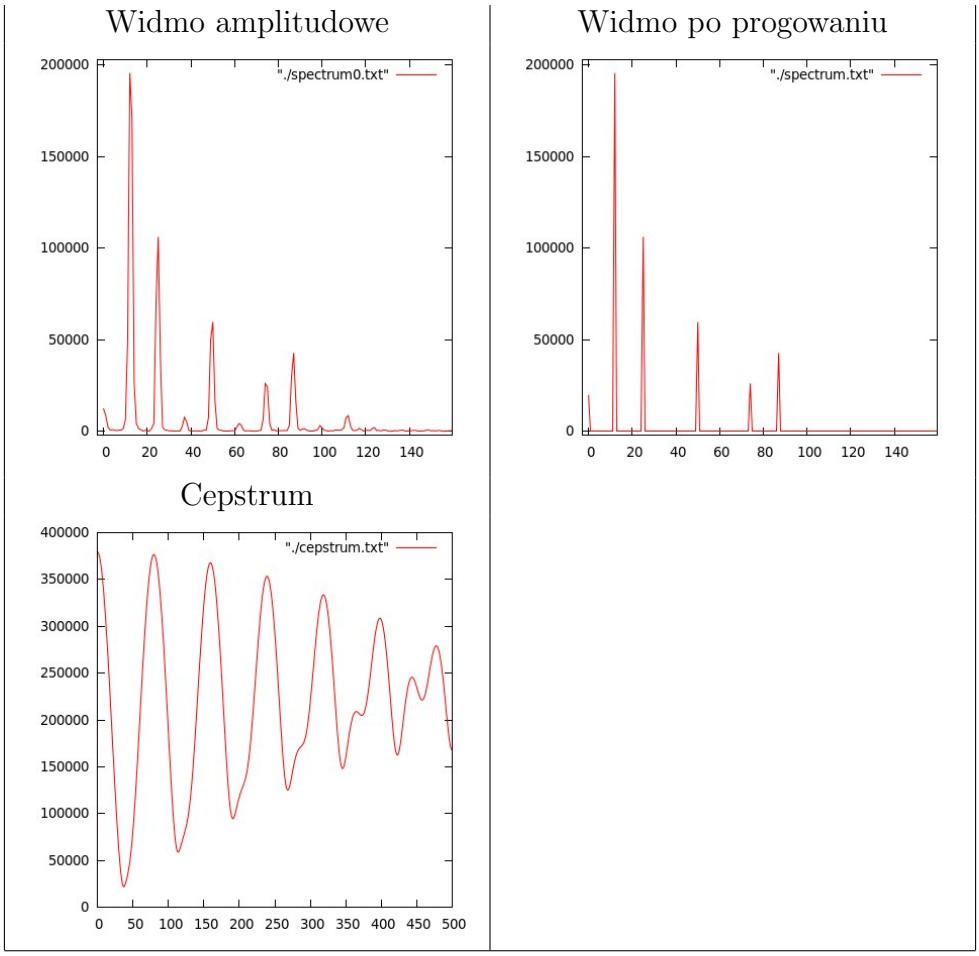
Dla dźwięku altówki o częstotliwości 130Hz:



Na tym wykresie widać pomniejsze minima, których zignrowanie jednak było bardzo łatwe.

Analiza cepstralna przebiega następująco:





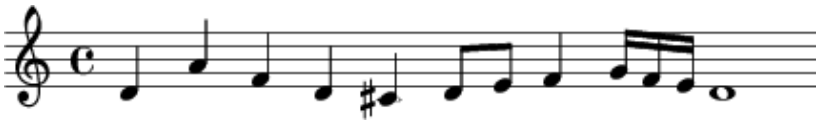
Sekwencje

Pianino

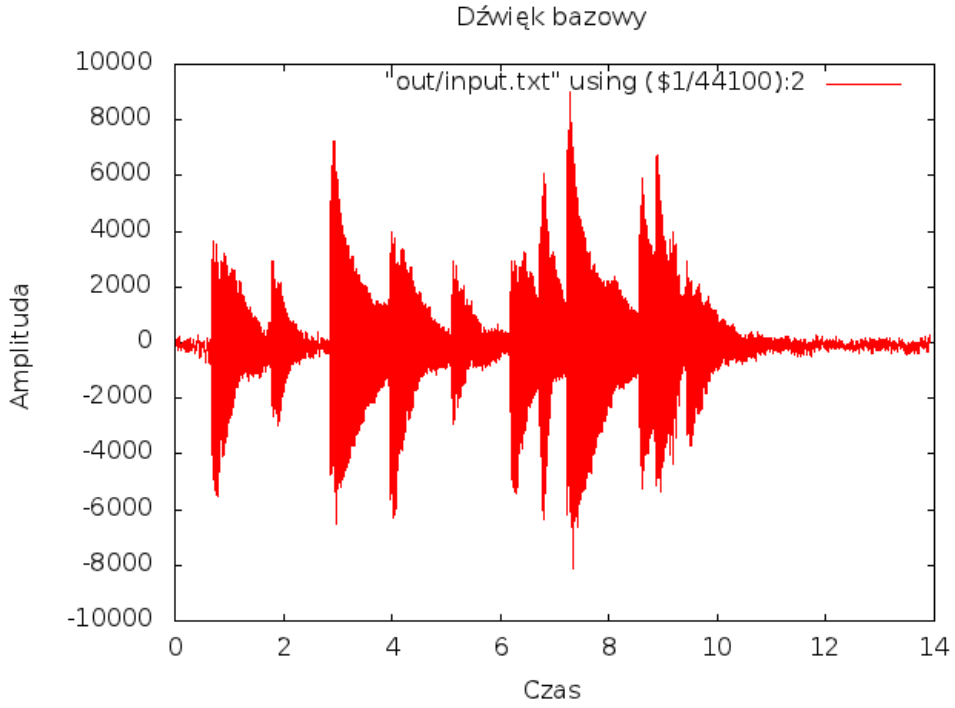
W wypadku pianina efekty programu są bardzo dobre. Przygotowany algorytm generowania sekwencji dopasowuje amplitudę tak, by przypominała dźwięk wejściowy. Efekty są następujące:

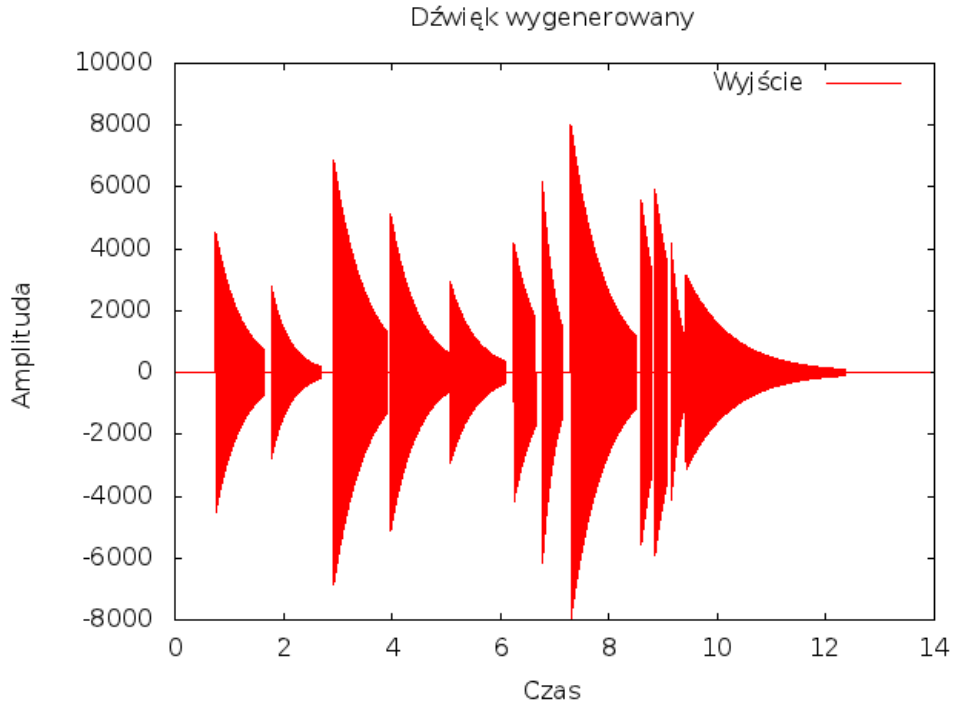
Początek	Czas trwania	Częstotliwość
0.75s	0.90s	294 Hz
1.77s	0.92s	439 Hz
2.92s	0.97s	349 Hz
3.97s	1.07s	294 Hz
5.07s	1.02s	278 Hz
6.26s	0.40s	295 Hz
6.77s	0.37s	331 Hz
7.30s	1.20s	349 Hz
8.60s	0.20s	390 Hz
8.85s	0.22s	347 Hz
9.15s	0.22s	329 Hz
9.42s	2.95s	294 Hz

W zapisie muzycznym:



Przygotowano również porównanie amplitud plików wejściowego i wygenerowanego:





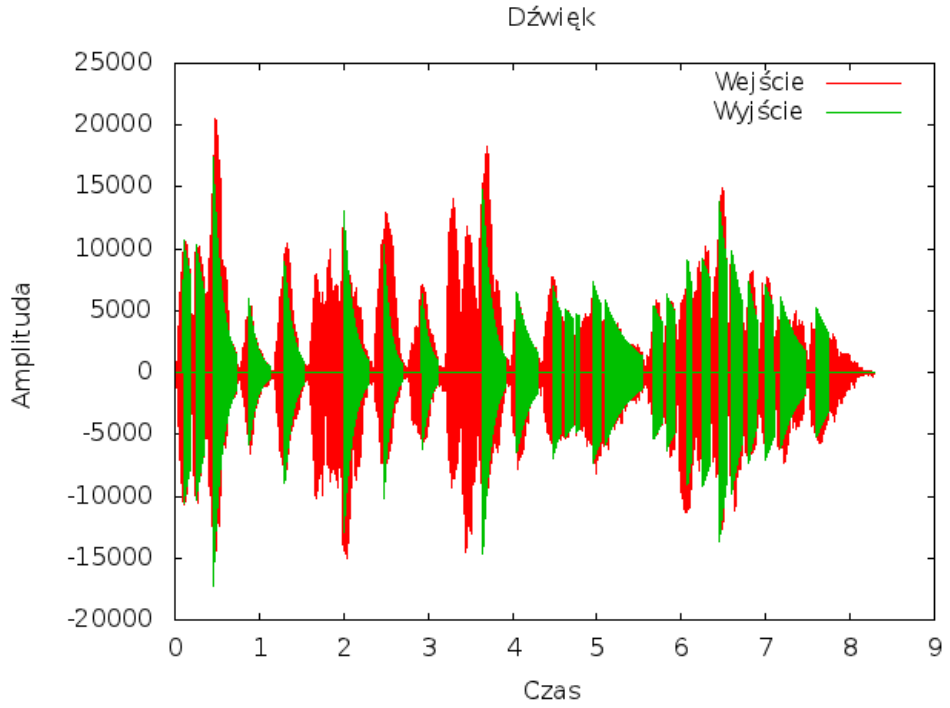
Jak łatwo zaobserować na wykresach, a także odsłuchując wygenerowany plik dźwiękowy rozpoznanie dźwięków zakończyło się sukcesem.

Skrzypce

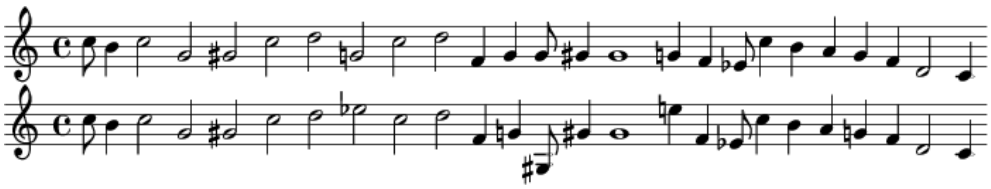
W wypadku skrzypiec wyniki są nieco gorsze — nie obejmują wszystkich dźwięków:

```
0.0999546485261 second: 518.823529412 Hz for 0.0749659863946 seconds
0.249886621315 second: 495.505617978 Hz for 0.0999546485261 seconds
0.449795918367 second: 518.823529412 Hz for 0.274875283447 seconds
0.874603174603 second: 393.75 Hz for 0.249886621315 seconds
1.29941043084 second: 416.037735849 Hz for 0.224897959184 seconds
1.99909297052 second: 525.0 Hz for 0.299863945578 seconds
2.47387755102 second: 588.0 Hz for 0.224897959184 seconds
2.92367346939 second: 390.265486726 Hz for 0.199909297052 seconds
3.6483446712 second: 518.823529412 Hz for 0.274875283447 seconds
4.04816326531 second: 588.0 Hz for 0.249886621315 seconds
4.47297052154 second: 350.0 Hz for 0.0999546485261 seconds
4.62290249433 second: 393.75 Hz for 0.0999546485261 seconds
4.74784580499 second: 383.47826087 Hz for 0.049977324263 seconds
4.94775510204 second: 416.037735849 Hz for 0.0999546485261 seconds
5.09768707483 second: 416.037735849 Hz for 0.449795918367 seconds
5.67242630385 second: 393.75 Hz for 0.0999546485261 seconds
5.82235827664 second: 350.0 Hz for 0.0999546485261 seconds
6.07224489796 second: 310.563380282 Hz for 0.049977324263 seconds
6.24716553288 second: 525.0 Hz for 0.0999546485261 seconds
6.44707482993 second: 495.505617978 Hz for 0.0999546485261 seconds
6.59700680272 second: 436.633663366 Hz for 0.124943310658 seconds
6.79691609977 second: 393.75 Hz for 0.0999546485261 seconds
6.99682539683 second: 350.0 Hz for 0.0999546485261 seconds
7.17174603175 second: 294.0 Hz for 0.299863945578 seconds
7.59655328798 second: 260.946745562 Hz for 0.149931972789 seconds
```

Na wykresie możemy zauważyć oczywiste braki dźwięków:



Ponadto odsłuchując zauważamy licznie zniekształcenia — zwłaszcza pod koniec sekwencji, gdzie nuty są znacznie częstsze. Zniekształcenia są większe przy użyciu metody w dziedzinie częstotliwości. Efektem tego są różne zapisy nutowe dla obu metod:



Wnioski

- Zaimplementowane metody wykazały się wysoką, choć nie 100 procentową skutecznością. Brak skuteczności wynika z płynnej definicji maksimum szukanego w obu metodach — czasami wskazanie właściwego maksimum jest niemożliwe nawet dla człowieka.
- Obie metody wykazały się również pewną niedokładnością. Głównym czynnikiem wpływającym na brak idealnych rezultatów było ograniczenie próbki. Tworząc próbkę, której długość nie jest wielokrotnością częstotliwości bazowej uniemożliwiliśmy w pełni dokładną analizę dźwięków.
- W sekwencyjnych utworach zauważono zniekształcenia dla krótkich dźwięków, co potwierdza, że rozważane metody działają lepiej otrzymując dłuższe próbki.
- Zaproponowana metoda rozdzielania dźwięków w sekwencji okazała się wysoce skuteczna dla pianina i znacznie mniej dla skrzypiec. Stało się tak dlatego, iż metoda została przygotowana z myślą o charakterystyce

dźwięku pianina, która znacząco różni się od charakterystyki dźwięku skrzypiec.