Intelligent Modular Audio Recognition Engine

Podręcznik użytkownika

http://www.imare.pl/

Spis treści

Wstęp

IMARE to środowisko umożliwiające sympatykom muzyki przetworzenie na zapis nutowy ulubionych utworów bądź własnych kompozycji.

Rozpoznawanie muzyki polega na matematycznej analizie sygnału audio (zwykle w formacie WAVE), wyodrębnianiu z niego słyszalnych struktur dźwiękowych i przedstawieniu ich w postaci notacji muzycznej. Jest to bardzo złożony problem nieposiadający jednoznacznego rozwiązania.

Dla porównania, zagadnienie rozpoznawania zeskanowanego tekstu – OCR – jest rozwiązywane z dokładnością 95% (wartość ta oznacza przeciętną dokładność programów tej klasy). Programy służące do rozpoznawania mowy działają już z dokładnością 70-80%, zaś systemy rozpoznawania muzyki mają 60-70% dokładności, jednakże osiągają taki wynik tylko dla monofonicznych melodii (jedna nuta na raz) – dla muzyki polifonicznej ta skuteczność jest jeszcze mniejsza.

Dlaczego używać?

IMARE jest efektywny, funkcjonalny i prosty w obsłudze. Skutecznie przetwarza na nuty pliki dźwiękowe nawet o wysokim poziomie zaszumienia.

Cel

Celem projektu jest wspieranie pasjonatów muzyki w żmudnej pracy tworzenia ręcznego zapisu nutowego ich kompozycji. Jednocześnie celem jest zagwarantowanie użytkownikowi możliwości obserwacji oraz analizy notacji utworów muzycznych.

Zalety

- Wieloplatformowość – staranny dobór narzędzi programistycznych zapewnia stabilną pracę na różnych platformach sprzętowych i pod kontrolą różnych systemów operacyjnych.
- Prostota obsługi – dzięki licznym testom domyślny dobór parametrów rozpoznawania daje bardzo dobre rezultaty dla znaczącej większości utworów muzycznych.
- Wydajność – dzięki zastosowaniu algorytmów Szybkiej Transformaty Fouriera i wydajnej transformaty *Constant Q* oraz wykorzystaniu zrównoleglonych obliczeń IMARE jest wyjątkowo efektywnym narzędziem do rozpoznawania dźwięku.
- IMARE vs profesjonalne narzędzia – wyniki testów pokazały, że dla większości utworów IMARE lepiej rozpoznał ich zapis nutowy w porównaniu z komercyjnymi narzędziami dostępnymi na rynku, m.in. WIDI lub TS-AudioToMidi.

Wymagania systemowe

Do optymalnego działania aplikacji wymagany jest procesor Pentium III i 256 MB pamięci operacyjnej. W systemie operacyjnym zainstalowane musi być środowisko JRE w wersji 1.5 lub wyższej wraz z Java™ SoundAPI (http://java.sun.com/products/java-media/sound/soundbanks.html).

Testy

IMARE został przetestowany na następujących konfiguracjach sprzętowych:

- Intel Core2Duo @1,66 GHz, 1024 MB RAM, Windows XP Professional SP3 32-bit
- Intel Core2Duo @2,00 GHz, 3072 MB RAM, Windows Vista Home Premium 64-bit
- Intel Core2Duo @1,73GHz, 2048 MB RAM, Sabayon 5
- Intel Pentium IV 3,00 GHz, 1024 MB RAM, Windows 7 Professional 32-bit

Działanie programu

Metoda rozpoznawania

Proces rozpoznawania linii melodycznych składa się z kilku etapów. Po pierwsze, materiał dźwiękowy musi zostać wczytany do pamięci lub udostępniony przez urządzenie multimedialne. Następnie, przebieg dźwiękowy badany jest w określonych odstępach czasu przy pomocy odpowiednich transformacji, umożliwiających uzyskanie informacji o spektralnej strukturze sygnału. Analiza uzyskanego w ten sposób widma dźwięku pozwala na wyodrębnienie w nim wyróżniających się z szumu struktur, które są następnie analizowane pod kątem ich czasowej zmienności. Znalezione w ten sposób ciągi dźwiękowe są ze sobą porównywane w celu określenia zależności harmonicznych pomiędzy tonami składowymi. Ostatnim etapem rozpoznawania jest dopasowanie otrzymanego materiału dźwiękowego do formy zapisu nutowego oraz jego odpowiednia wizualizacja.

Wczytywanie danych

W przypadku standardowego trybu rozpoznawania cały materiał dźwiękowy wczytywany zostaje do pamięci operacyjnej, skąd wskazane fragmenty mogą być przekazywane na żądanie do odpowiednich jednostek. Tak wybrany sposób wczytywania danych jest efektywny pod względem technicznym, gdyż minimalizuje liczbę dostępów do dysku, a ponadto umożliwia równoczesne przeprowadzanie analizy dźwięku i jego wizualizację. Do obsługiwanych formatów plików zaliczają się:

- PCM Wave, próbkowanie 8, 16 i 24 bity, zarówno jedno- jak i dwukanałowe,
- MPEG-1 Audio Layer-3 (MP3)

W trybie rozpoznawania melodii w czasie rzeczywistym dane przesyłane są bezpośrednio do jednostki rozpoznającej. Stworzony zostaje nieskompresowany, jednokanałowy strumień wejściowy, kwantowany 16-bitowo, o domyślnej częstotliwości próbkowania równej 22050 Hz.

Transformacja Constant Q

Jak już wspomniano, do wydobycia informacji o widmowej strukturze sygnału służą transformacje zespolone, zazwyczaj związane z obliczaniem Dyskretnej Transformaty Fouriera (DFT). Najczęściej stosowaną i dość efektywną metodą uzyskiwania widma dźwięku jest Szybka Transformata Fouriera, nie jest ona jednak odpowiednim rozwiązaniem w przypadku rozpoznawania melodii. Problem z transformacją FFT w tym kontekście polega na tym, że zwraca ona widmo dźwięku w liniowej skali częstotliwości, co nie odpowiada rozkładowi dźwięków w powszechnie przyjętej skali muzycznej. Na skutek tego, górne partie widma zawierają niewielką liczbę sąsiadujących dźwięków, tymczasem w jego dolnej części niskie dźwięki zaczynają na siebie nachodzić, znacznie utrudniając efektywną analizę melodii.

Jednym z rozwiązań dla tego problemu jest zastosowanie transformacji Constant Q, zaproponowanej[1] jako analogiczna do DFT transformacja, zwracająca widmo dźwięku nie w liniowej, lecz logarytmicznej skali częstotliwości. Cechą charakterystyczną tej transformacji jest zmienna szerokość przedziału czasowego dla różnych częstotliwości – jest to powodowane tym, że, zgodnie z nazwą, transformacja ta stara się zachować stałą, niezależną od częstotliwości jakość odwzorowania.

---

1) Judith C. Brown, Calculation of a constant Q spectral transform, *J. Acoust. Soc. Am.*, 89(1):425–434, 1991.

W bardzo podobny sposób zachowuje się ludzki zmysł słuchu. Również i tutaj naturalna zdaje się być logarytmiczna skala częstotliwości, a co więcej, czasowa zdolność rozdzielcza okazuje się być mniejsza dla dźwięków niższych.

W przypadku transformacji Constant Q problem stanowi jej niewielka efektywność obliczeniowa. Pierwotna, oparta bezpośrednio o DFT postać transformacji przedstawia się bowiem następująco:

$$X[k] = \frac{1}{N[k]} \sum_{n=1}^{N[k]} W_k[n] x[n] e^{\frac{2\pi i Q n}{N[k]}}$$

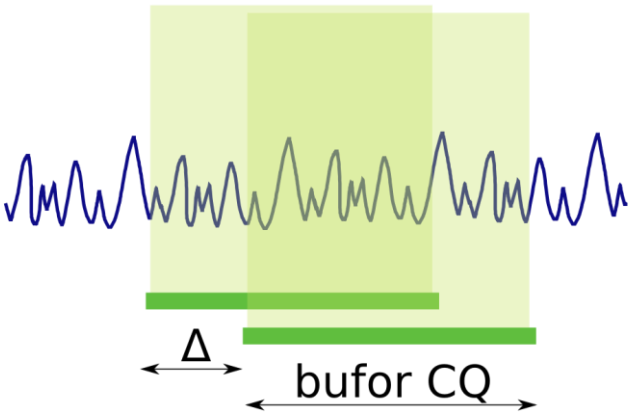
x[1..N] – próbka sygnału
W[1..N] – wartości okna czasowego
X[k] – transformata Constant Q sygnału
Q – parametr jakości transformacji

Szczęśliwie, okazuje się[2], że transformata Constant Q może być obliczona, korzystając z transformacji FFT z wykorzystaniem splotu z odpowiednio dobranymi funkcjami. Wynikająca stąd metoda wyznaczania transformaty jest także o tyle atrakcyjna, że umożliwia uprzednie obliczenie wymaganych wartości pośrednich, co znacznie przyspiesza pracę w przypadku wielokrotnego obliczania transformaty. Szczegóły obliczania transformaty można znaleźć w pracy cytowanej w bieżącym akapicie. Wersja transformacji Constant Q wykorzystana w projekcie Imare jest oparta właśnie o tą metodę.

Analiza widmowa

Opisana powyżej transformacja Constant Q pełni w procesie rozpoznawania rolę kluczową. Pozwala ona na uzyskanie widma dźwięku w dowolnie momencie – a właściwie przedziale – czasu. W przypadku transformacji Constant Q, szerokość tego przedziału, a co za tym idzie, rozmiar wejściowego bufora transformacji, jest wyznaczony jednoznacznie przez częstotliwość próbkowania sygnału i parametr Q określający jakość przekształcenia[3]. Poszczególne kanały wynikowego widma rozmieszczone są co ½ półtonu (ćwierćton), co daje 24 kanały na oktawę. Zakres widma wyznaczony jest przez przedział częstotliwości ustalany przez użytkownika, przebiegający domyślnie pomiędzy 110 Hz i 10000 Hz. Jeśli górna granica częstotliwości przekracza częstotliwość Nyquista dla danej częstotliwości próbkowania, zostaje odpowiednio obniżona.

W celu odpowiednio precyzyjnego odtworzenia czasowej zmienności dźwięku, nierzadko wymagane staje się, aby odstęp czasowy między dwoma kolejnymi wyznaczeniami widma (parametr ustalany przez użytkownika) był mniejsze od szerokości bufora. Warto zauważyć, że nawet w takiej sytuacji, szczególnie w przypadku wysokich



2) Judith C. Brown and Miller S. Puckette, An ef Q transform, *J. Acoust. Soc. Am.*, 92(5):2698–2701, 1

3) Dla systemu dwunastopółtonowego powinno zachodzić Q ≥ 17 i taka wartość graniczna jest przyjęta w programie.

częstotliwości, dwa sąsiednie wyznaczenia widma wciąż mogą być w pełni niezależne, gdyż dla większych częstotliwości efektywny rozmiar okna wejściowego stanowi jedynie część rozmiaru bufora.

Pierwszą czynnością po załadowaniu aktualnego obrazu widma jest oszacowanie poziomu szumu. W obecnej implementacji oblicza się w tym celu medianę poziomu sygnału, zaś próg detekcji uzyskuje się, mnożąc otrzymany poziom szumu przez jeden z parametrów ustalanych przez użytkownika (próg rejestrowanego sygnału). Po obliczeniu progu detekcji, zostają wykryte wszystkie lokalne maksima w widmie, które jednocześnie mają wartość wyższą od progowej. Oczywiście, skoro widmo składa się z wartości zespolonych, wszelkie porównania wykonywane są na amplitudach poszczególnych wartości. Następnie, porównywane są otrzymane listy maksimów dla aktualnego i poprzedniego widma, zaś do dalszego przetwarzania przeznaczane są jedynie maksima znajdujące się jednocześnie w obu listach. Z tego też powodu, dla każdego kroku czasowego przechowywane są dwa przebiegi widma: aktualny i poprzedni.

Wstępną wartość częstotliwości danego maksimum otrzymuje się, odczytując częstotliwość przypisaną do danej pozycji widma. Precyzja wyznaczenia częstotliwości może być znacznie zwiększona, wykorzystując metodę określaną jako „phase vocoder". W tym celu porównywana jest aktualna faza sygnału z fazą sygnału otrzymaną w poprzednim widmie. Jednocześnie, obliczana jest różnica faz wynikająca wprost ze znanej częstotliwości dźwięku i znanego przesunięcia czasowego pomiędzy widmami. Na podstawie różnicy pomiędzy tymi przesunięciami wyznaczana jest addytywna poprawka do częstotliwości. Jeśli poprawiona częstotliwość nie odbiega znacząco od częstotliwości wyjściowej[4], wykryte maksimum lokalne (od tej pory określane mianem *szczytu*) wraz z poprawioną częstotliwością jest przekazywane do dalszego rozpoznawania.

Wyznaczanie czasowego przebiegu dźwięków

W celu wyznaczenia dźwięków linii melodycznych, dotychczasowy przebieg każdego z dźwięków porównywany jest ze szczytami znalezionymi w obecnym widmie, aby stwierdzić, czy i które z nich stanowią kontynuację wcześniej rozpoczętych dźwięków. W tym celu, częstotliwość szczytu porównywana jest z uśrednioną częstotliwością dźwięku (średnią ze wszystkich szczytów wchodzących w jego skład). Jeśli różnica mieści się w przedziale tolerancji ustalonym przez użytkownika (domyślnie 1 ćwierćton), szczyt dodawany jest do przebiegu dźwięku, aktualizując jednocześnie jego średnią częstotliwość. W każdym kroku czasowym, dźwięki od pewnego czasu nieaktualizowane (parametr czasowy ustalany przez użytkownika) są usuwane z listy oczekujących. Szczyty, które nie zostały dodane do żadnego dźwięku, same tworzą nowy dźwięk (składający się z jednego szczytu), który zostaje dodany do listy oczekujących.

Istnieje jednakże jeden wyjątek od tej reguły. Dla każdego dźwięku, poza przebiegiem częstotliwości, analizowany jest także przebieg głośności poszczególnych szczytów. Jeśli dodawany szczyt znacząco przekracza średnią głośność w krótkim odstępie czasu[5], nie jest on uznawany za kontynuację dźwięku, lecz rozpoczyna nowy dźwięk na tej samej wysokości, a dźwięk poprzedni jest usuwany z listy oczekujących.

---

4) Przedział tolerancji dla metody phase vocoder wynosi ±0.55 ćwierćtonu.

5) Domyślnie, średnia obliczana jest na podstawie szczytów z przedziału czasowego 0.1 s.

Instrukcja obsługi

Menu programu

Projekt

- Otwórz plik audio – otwiera plik w formacie WAV lub MP3.
- Ostatnio otwierane – lista ostatnio używanych projektów.
- Rozpoznawaj ze źródła – uruchamia transkrypcję dźwięku w czasie rzeczywistym.
- Zamknij – zamyka program IMARE.

Okna

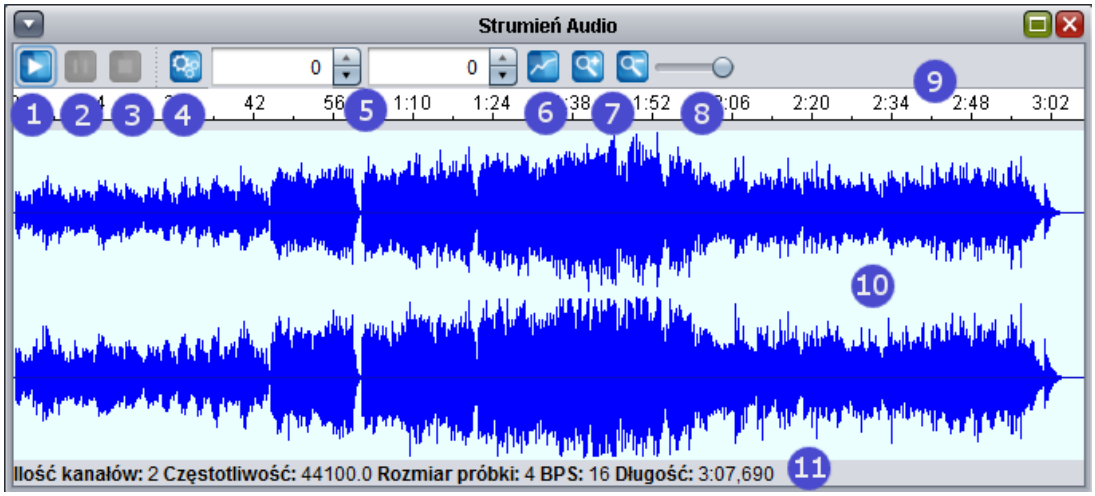
- Ustawienia transkrypcji – wyświetla okno ustawień procesu rozpoznawania.
- Ułóż pionowo – układa wszystkie okna w oknie głównym jedno pod drugim.
- Ułóż poziomo – układa wszystkie okna w oknie głównym jedno obok drugiego.
- Ułóż kaskadowo – układa wszystkie okna w oknie głównym jedno na drugim.

Pomoc

- O programie... – otwiera okno z informacjami o programie.
- Licencja – wyświetla licencję programu.

Opis okien aplikacji

Strumień audio

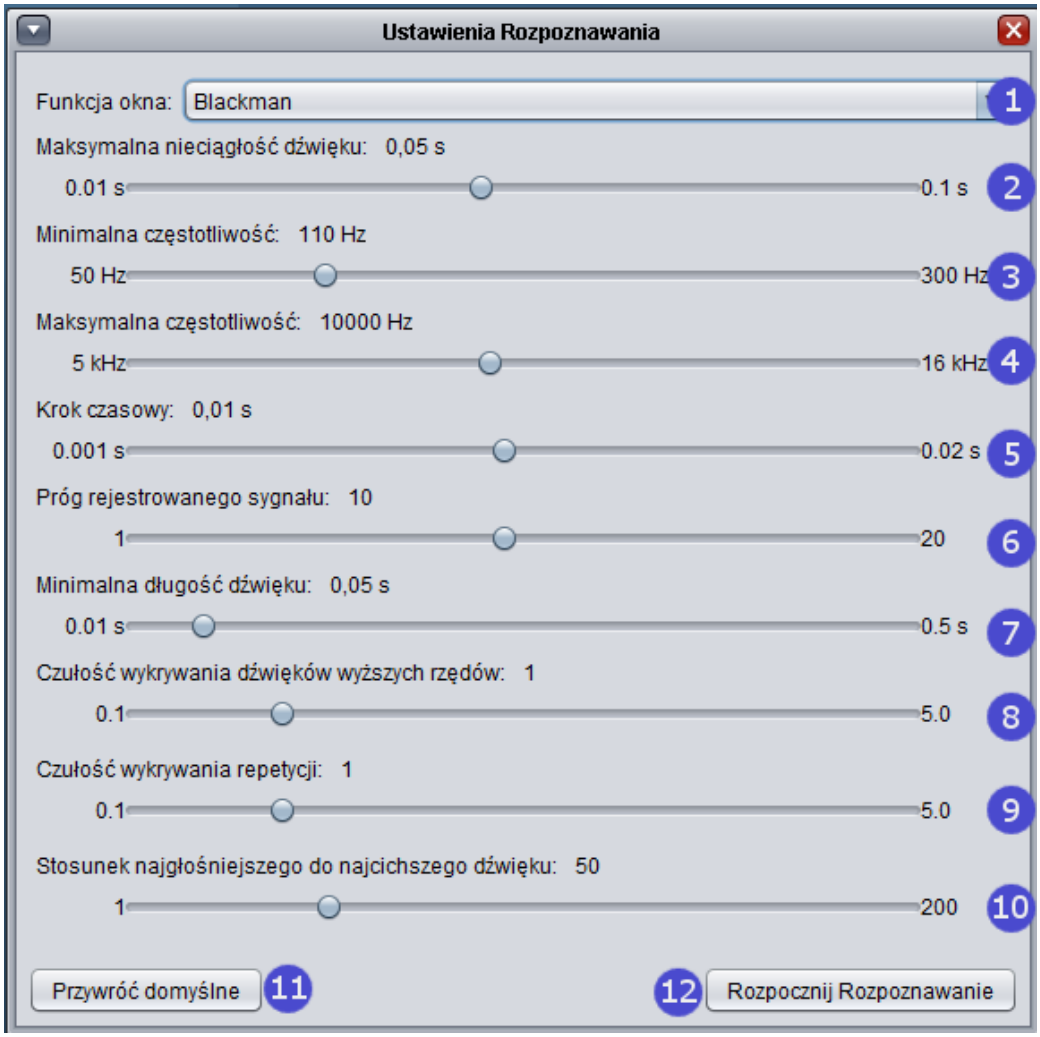


Rys. 2.3: Okno Strumień Audio.

1: Rozpoczyna odtwarzanie pliku dźwiękowego.
2: Wstrzymuje odtwarzanie.
3: Zatrzymuje odtwarzanie.
4: Rozpoczyna rozpoznawanie dźwięku.
5: Pola wskazujące początkową i końcową pozycję zaznaczenia w sekundach.
6: Wyświetla okno rysujące widmo zaznaczenia (patrz rozdział Okno widma).
7: Przybliżenie/oddalenie. W przypadku, gdy zaznaczony jest fragment ścieżki dźwiękowej, przybliżenie obejmuje całość zaznaczenia.
8: Regulacja głośności odtwarzania.
9: Skala czasowa pliku.
10: Rysunek fali dźwiękowej wyświetlający jeden kanał dla mono lub dwa dla stereo. Możliwe jest zaznaczenie fragmentu ścieżki z użyciem myszy.
11: Informacje o pliku zawierające następujące dane:

- **Ilość kanałów –** 1 lub 2;
- **Częstotliwość** pliku dźwiękowego w hercach;
- **Rozmiar ramki** w bajtach;
- **BPS –** Bits Per Sample**;**
- **Długość** dźwięku z dokładnością do jednej tysięcznej sekundy.

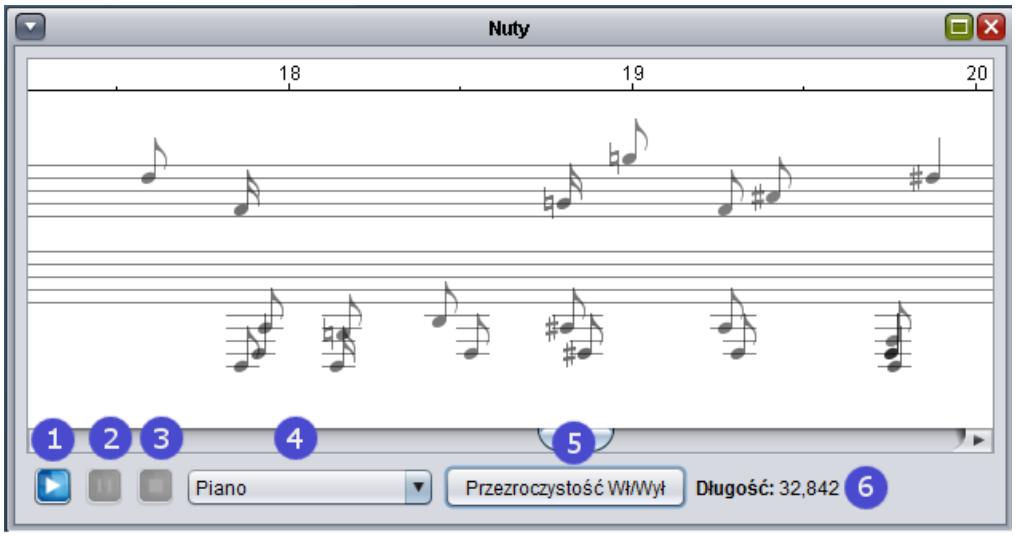
Ustawienia transkrypcji



Rys. 2.2: Okno opcji mechanizmu rozpoznającego.

1: Wybór funkcji okna spośród czterech do wyboru: Dirichlet, Blackman, Blackman Nutall lub Hamming.
2: Maksymalna przerwa w danym półtonie, przy której jest on traktowany jako ciągła nuta.
3: Minimalna częstotliwość brana pod uwagę przy rozpoznawaniu.
4: Maksymalna częstotliwość brana pod uwagę przy rozpoznawaniu.
5: Częstotliwość pobierania próbek z pliku dźwiękowego podczas rozpoznawania. Im niższa wartość, tym dokładniejsze rozpoznawanie.
6: Wielokrotność mediany siły sygnału, przy której jest on brany pod uwagę przy rozpoznawaniu.
7: Minimalna długość nuty, przy której jest ona wyrzucana na pięciolinię.
8: Zakres różnicy głośności nut. Im wyższa wartość, tym wyraźniej zaznaczona dynamika.
9: Czułość wykrywania wyższych dźwięków.
10: Czułość wykrywania repetycji.
11: Przywraca domyślne ustawienia.
12: Rozpoczyna proces rozpoznawania utworu.

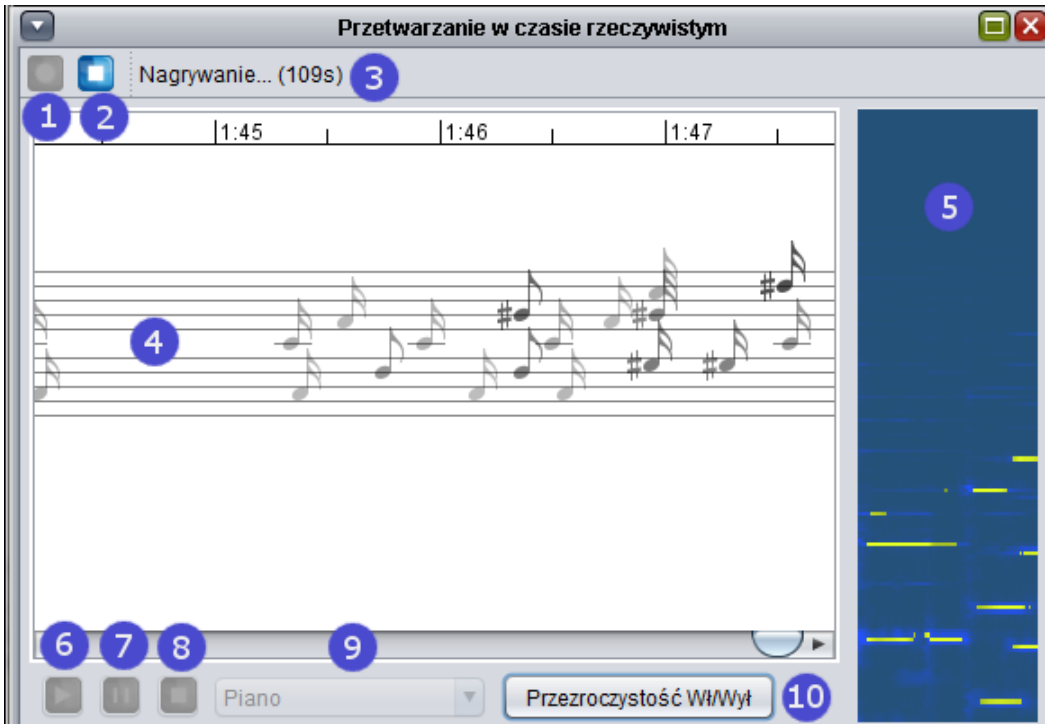
Zapis nutowy



Rys. 2.4: Okno z nutami.

1: Rozpocznij odtwarzanie nut.
2: Wstrzymaj odtwarzanie.
3: Zatrzymaj odtwarzanie.
4: Wybór instrumentu MIDI, za pomocą którego rysowane są nuty.
5: Włącza lub wyłącza przejrzystość nut w zależności od ich głośności.
6: Wyświetla długość ścieżki dźwiękowej w sekundach.

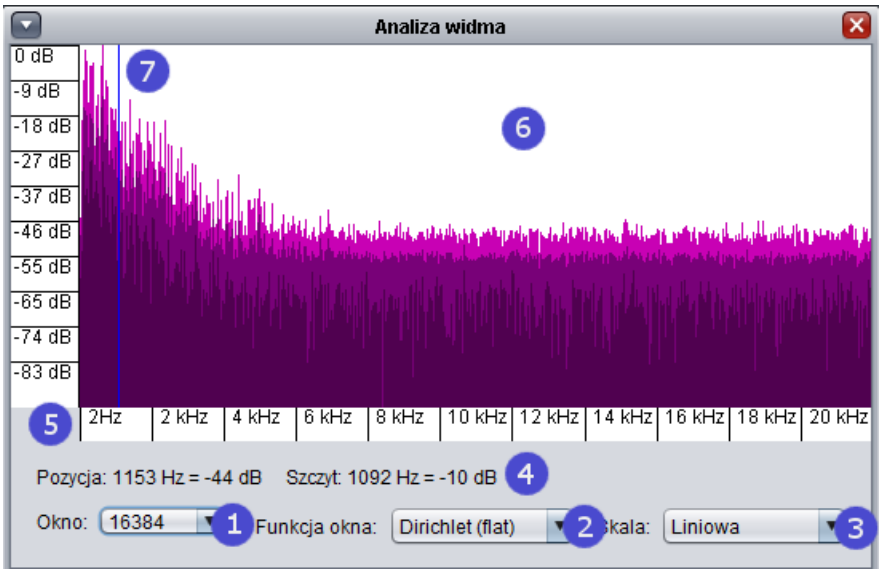
Rozpoznawanie w czasie rzeczywistym



Rys. 2.5: Okno rozpoznawania w czasie rzeczywistym.

1: Rozpoczęcie nowego nagrywania.
2: Zatrzymanie nagrywania i skopiowanie wynikowych nut na okno z notacją.
3: Czas nagranych danych w sekundach.
4: Wynik transkrypcji strumienia audio na nuty.
5: Podgląd widma aktualnie rozpoznawanego strumienia. Im jaśniejsza linia, tym większa dynamika danego dźwięku.
6: Rozpoczęcie odtwarzania nut.
7: Wstrzymanie odtwarzania nut.
8: Zatrzymanie odtwarzania nut.
9: Wybór instrumentu MIDI.
10: Włączenie lub wyłączenie przejrzystości w zależności od dynamiki danej nuty.

Widmo (FFT) dźwięku



Rys. 2.6: Okno widma.

Powyższe okno rysuje widmo zaznaczonego wcześniej fragmentu ścieżki dźwiękowej. Oto poszczególne elementy okna:

1: Rozmiar okna widma w ilości próbek – kolejne potęgi dwójki od 512 do 16384.
2: Wybór funkcji okna – Dirichlet, Blackman, Blackman-Nutall lub Hamming.
3: Skala wykresu – liniowa lub logarytmiczna.
4: Informacje o danym miejscu widma zaznaczone linią **7**.
5: Opis skali wykresu: oś OX – kolejne częstotliwości w hercach; oś OY – różnice w natężeniu między daną częstotliwością a maksimum.
6: Pole wykresu. Najjaśniejsze pole oznacza wartość maksymalną w tym miejscu, pole ciemniejsze oznacza średnią, zaś najciemniejsze – wartość minimalną.
7: Linia o pozycji ustalanej ruchem myszy.

Licencja