

ТЕОРІЯ ТА МЕТОДИ ОБРОБЛЕННЯ СИГНАЛІВ

УДК 004.4'277.02 (045)

¹В. С. Яцковський, асист.,
²Д. М. Бруслік**АЛГОРИТМ ОЦІНЮВАННЯ ТЕМПУ МУЗИКАЛЬНИХ СИГНАЛІВ**Інститут аерокосмічних систем управління НАУ, e-mail: ¹yatskovsky@gmail.com,
²pacification@list.ru

Розглянуто алгоритм розпізнавання темпу аудіо-сигналів (mp3 файлів). Запропоновано метод підвищення точності оцінювання.

Ключові слова: обчислення темпу, спектрограма, мелчастотний кепстральний аналіз, автокореляція, квадратична апроксимація.

Вступ і постановка завдання. Методи і алгоритми розпізнавання музики є складовою частиною систем і засобів штучного інтелекту. Одним із напрямів штучного інтелекту, що активно розвиваються на сучасному етапі, є створення апаратно-програмних комплексів, що імітують процеси сприйняття та розпізнавання музики та музичних творів. Пошук темпу в музичному звукозаписі може стати основою для прикладних програм, таких як автоматичний акомпанемент, транскрипція, комп'ютерний монтаж аудіо, музикальна подібність та ін.

Метою цієї роботи є створення програмної реалізації алгоритму розпізнавання темпу mp3-файлів, яка 1) має точність, достатню для практичних прикладень ($\pm 0,5$ bpm – ударів за хвилину); 2) виконується не більше як за хвилину на сучасному персональному комп'ютері; 3) споживає не більше декількох десятків мегабайтів оперативної пам'яті; 4) може бути застосований в деякому іншому програмному продукті.

Прототип алгоритму ґрунтується на алгоритмі, запропонованому Даніелем Елліном у статті [1]. Автор люб'язно надав робочий варіант у вигляді *m*-файлу (мова системи MatLab), що дозволило виконати тести та оцінити придатність алгоритму для розв'язання поставлених завдань. Щоб задовольнити вимогу 4), було вирішено створити програмну динамічну бібліотеку (.dll) мовою C# шляхом портування коду з мови MatLab. Це є нетривіальним завданням, оскільки мова C# не має вбудованих функцій для перетворення Фур'є, автокореляції, цифрової фільтрації та інших математичних обчислень, а також не надає такої розвиненої системи візуалізації та відлагодження математичної інформації, як MatLab.

Темп музикальної композиції оцінюється кількістю ударів метронома за одну хвилину і позначається скорочено bpm (beats per minute).

Ініціалізація алгоритму та спектрограма. Перший етап роботи алгоритму – це декомпресія mp3 файлу у формат pcm (pulse-code modulation), тобто масив значень амплітуд звукового сигналу, виміряних через однакові проміжки часу $T=1/f$, де f – частота звукового файлу. Тут можливі дві оптимізації: по-перше, алгоритм не потребує інформації, що міститься в стереосигналі, тому сигнал конвертується в моно. По-друге, згідно з теоремою Котельнікова аналоговий сигнал можна відновити з дискретизованого сигналу, якщо частота дискретизації вдвоє вища за верхню частоту його спектру. Це означає, що стандартний mp3-файлів з частотою 44100 Гц відтворює звук до 22050 Гц, що вище від порога чутливості людського слуху. Тим самим економиться значна кількість обчислюваних ресурсів зі зниженням частоти дискретизації аудіосигналу до 8000 Гц. Програмно це забезпечується фільтруванням сигналу НЧ-фільтром з частотою зрізу близько 4 кГц (фільтр 5-го порядку дає достатню крутизну характеристики) та відкиданням «зайвих» відліків.

Далі сигнал з часової перводиться в частотно-часову область й обчислюється спектрограма. Для цього короткий фрагмент аудіосигналу (256 відліків, або 32мс) перемножується на функцію вікна та обчислюється швидко перетворення Фур'є. Використання «прямокутного» вікна призводить до значних спотворень в обчисленні спектру,

тому використовуємо вікно Хана (а також інші поширені типи вікон, наприклад, Хемінга або Блекмана). Наступний фрагмент аудіосигналу береться зі зсувом у часі, що дорівнює половині або 1/4 ширини вікна, тобто фрагменти перекриваються. Величина зсуву між фрагментами (або «стрибка») безпосередньо впливає на точність алгоритму – чим вона менша, тим вища «роздільна здатність» алгоритму, але більше потрібно пам'яті та часу на функціонування програми.

Мелчастотний кепстральний аналіз. Цей тип частотного аналізу застосовують для оброблення звуку та розпізнаванні мови. Мел – це психофізична одиниця висоти звуку, пов'язана з частотою (в герцах) як

$$mel = 2595 \log_{10}(1 + f / 700),$$

а кепстр є результатом взяття оберненого перетворення Фур'є від логарифма спектру сигналу, тобто нелінійним «спектром спектру». Різниця між кепстром і мелчастотним кепстром полягає в тому, що в мелчастотному кепстрі смуги частот розміщені за шкалою Мела, що більше відповідає людській слуховій системі, ніж лінійно розміщені смуги частот, що використовуються у звичайному кепстрі [2].

Розраховуємо мелчастотні кепстральні коефіцієнти (з яких і складається мелчастотний кепстр) таким чином:

1. Обчислюємо матрицю коефіцієнтів перетворення розмірністю 40 (кількість частотних смуг спектрограми) на 256 (ширина вікна). Якщо ці два параметри не будуть змінюватися, то доцільно виконати обчислення один раз та зберегти матрицю як масив констант.

2. Перемножуємо матрицю модулів значень амплітуд спектрограми на матрицю коефіцієнтів.

3. Логарифмуємо результат та отримуємо матрицю, яка містить мелчастотні кепстральні коефіцієнти для різних моментів часу.

Для виділення моментів «атаки» аудіосигналу диференціюємо амплітуди в кожній частотній смузі (обчислюючи кінцеву різницю першого порядку), відкидаємо від'ємні значення та усереднюємо результат, таким чином отримуючи двовимірну функцію – залежність «сили атаки» від часу. Подальше оброблення сигналу потребує, щоб сигнал не мав постійного зсуву за всією амплітуд, тому видаляємо «постійну складову» за допомогою НІХ-фільтра другого порядку. Результат аналізу тестового аудіофайлу подано на рис. 1.

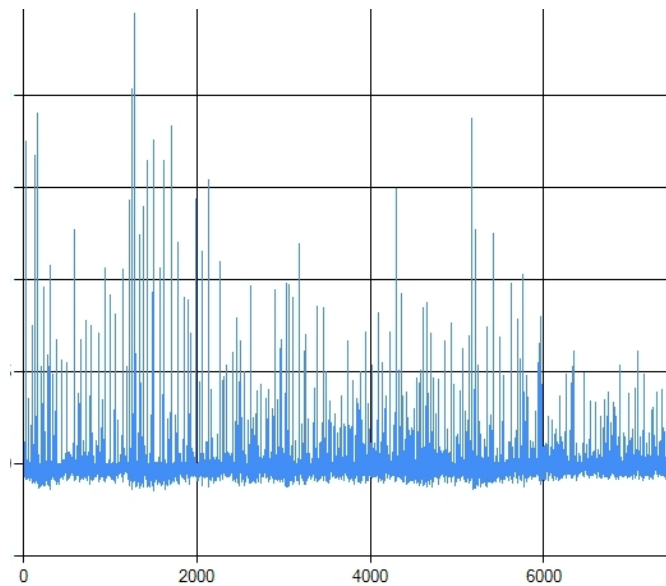


Рис. 1. Залежність «сили атаки» від часу на ділянці аналізованого аудіофайлу

Кореляційний аналіз та визначення темпу. На наступному кроці обчислюємо автокореляційну функцію сигналу «сили атаки» з максимальним зсувом 500 відліків (62,5 мс). Результат подано на рис. 2.

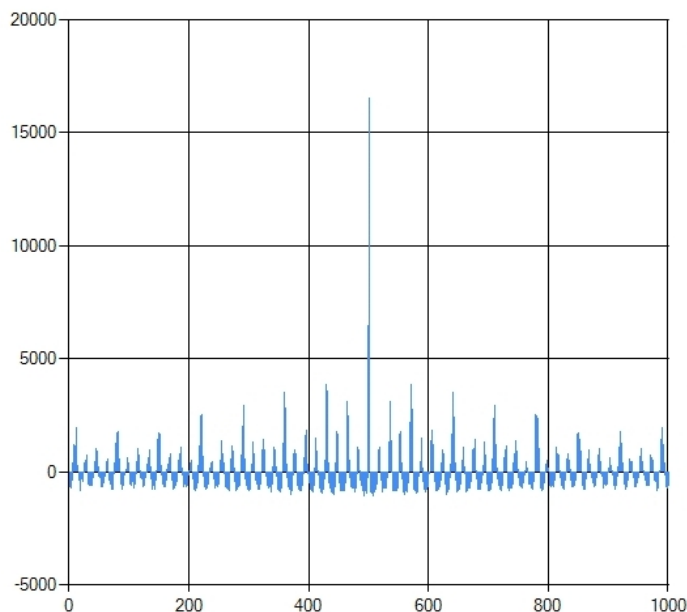


Рис. 2. Автокореляційна функція сигналу «сили атаки»

Піки цієї автокореляційної функції вказують на наявність періодичності у сигналі, а отже, на шуканий темп музичного твору. Вибір серед багатьох піків можна полегшити, якщо перемножити цей сигнал на функцію вікна спеціальної форми, що має максимум в точці 120 bpm (найбільш поширений темп), та плавно спадає по боках. У результаті, як видно на рис. 3, маємо чітко виділений максимум (для цього аудіофрагмента), з якого обчислюємо темп за формулою

$$tempo_{bpm} = \frac{60}{peakindex \cdot hop} \cdot f_s,$$

де $peakindex$ – номер піка автокореляційної функції; f_s – частота дискретизації; hop – ширина «стрибка» при обчисленні спектрограми.

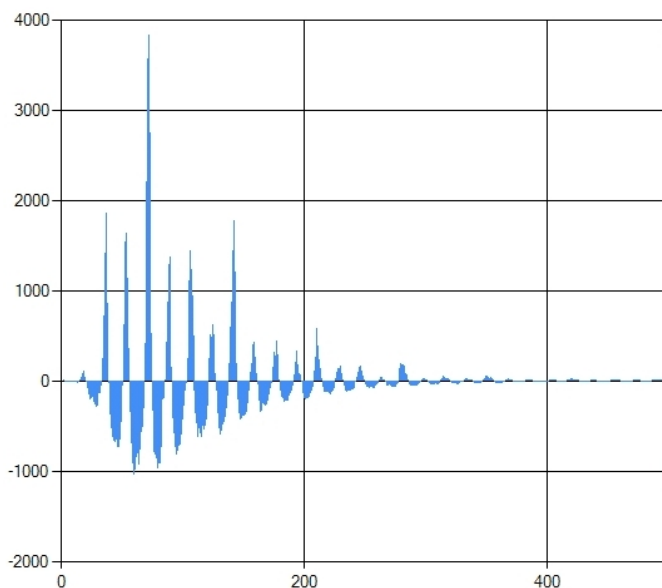


Рис. 3. Зважена автокореляційна функція

У розглядуваному прикладі (композиція Joe Satriani – Moroccan Sunset) максимум функцій припадає на відлік 71, отже оцінений темп становить 105,63 bpm.

Підвищення точності алгоритму. Точність оцінки темпу запропонованого алгоритму обмежена, окрім інших факторів, «роздільною здатністю» автокореляційної функції. Тобто реальне положення максимуму функції між дискретними значеннями зсунує ближче до лівого або правого відліку. Значення темпу становить

$$\frac{60}{\text{peakindex} + 0,5 \text{ hop}} \cdot \frac{f_s}{\text{peakindex} - 0,5 \text{ hop}} < \text{tempo}_{\text{bpm}} < \frac{60}{\text{peakindex} - 0,5 \text{ hop}} \cdot \frac{f_s}{\text{peakindex} + 0,5 \text{ hop}}.$$

Підвищити точність можна, зменшуючи параметр *hop*, але це призводить до збільшення обчислювальних ресурсів. Інший шлях – локально апроксимувати вид кореляційної функції за її дискретними відліками та знайти положення максимуму.

Апроксимацію квадратичною функцією в околі максимуму ілюструє рис. 4. Положення реального максимуму автокореляційної функції позначено зіркою.

Щоб вирахувати це значення, візьмемо квадратичну функцію $f(x) = ax^2 + bx + c$ з умовами $f(0) = y_0$, $f(-1) = y_{-1}$, $f(+1) = y_{+1}$, де y_0 – величина дискретного максимуму, що міститься в центрі координат, а y_{-1} та y_{+1} – зліва і справа відповідно. Як відомо, максимум квадратичної функції досягається у точці $-\frac{b}{2a}$. Розв'язавши систему рівнянь, отримаємо коефіцієнти:

$$c = y_0;$$

$$b = \frac{y_{+1} - y_{-1}}{2};$$

$$a = y_{-1} + b - c.$$

Таким чином, $\text{peakindex}^* = \text{peakindex} - \frac{b}{2a}$. Для розглядуваного прикладу це відповідає темпу 105,96, отже, корекція становить 0,33 bpm.

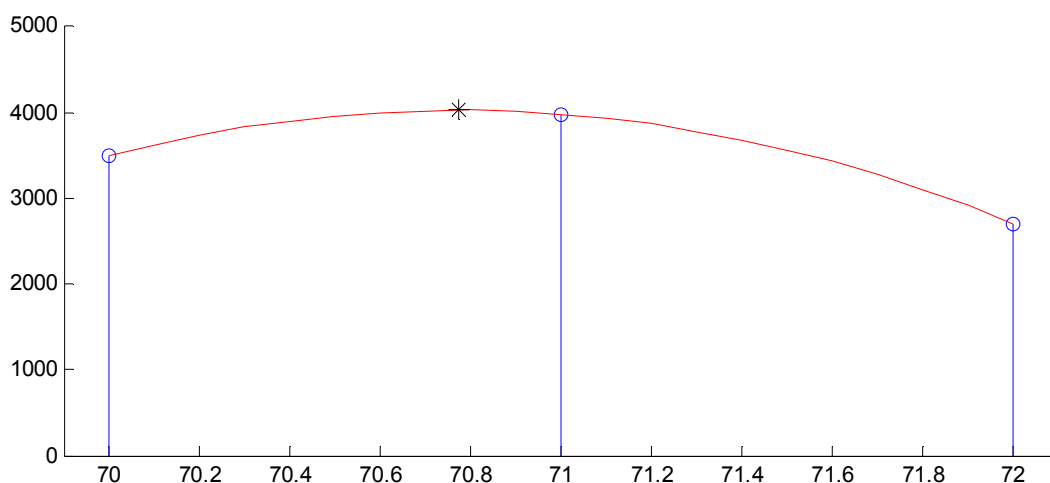


Рис. 4. Локальна апроксимація автокореляційної функції за допомогою квадратичної функції

Висновки. Перевірити темп можна, порівнюючи його з темпом, який визначив професійний музикант, або встановити темп на метрономі та прослуховувати його удари одночасно з музикою. Описаний алгоритм у базовому варіанті, як правило, потребує невеликої корекції, а із запропонованою методикою підвищення точності – визначає достовірний темп, що збігається з музикою (принаймні значний час, бо вимірюється середній темп твору, а темп музики у більшості випадків не є постійним для всього твору). Є деякі

музичні композиції, темп яких визначається неправильно (з точки зору музиканта); це стається у випадку, коли локальні максимуми автокореляційної функції мало відрізняються між собою, а максимум, що відповідає неправильному темпу, є трохи більшим за правильний.

Отже, у більшості випадків поданий алгоритм достовірно обчислює темп музичних творів у будь-якому стилі та жанрі. Він відповідає поставленому завданню і є придатним для використання в науково-дослідних або комерційних системах, що потребують оцінювання темпу музичних творів.

Список літератури

1. *Daniel P. W. Ellis*. Beat Tracking with Dynamic Programming LabROSA, Dept. of Electrical Engineering Columbia University. – New York NY 10027 USA. – 4 с.
2. *Lawrence R. Rabiner*. Introduction to digital speech processing / Lawrence R. Rabiner, Ronald W. Schafer. Now Publishers Inc, 2007 – 200 с.

В. С. Яцковский, Д. Н. Бруслик

Алгоритм оценивания темпа музыкальных сигналов

Рассмотрен алгоритм распознавания темпа аудио-сигналов (mp3 файлов). Предложен метод повышения точности распознавания.

V. S. Yatskovsky, D. N. Bruslik

Algorithm of tempo estimation of musical signals

One of algorithms of music tempo estimation is considered. The method of accuracy improvement is proposed.