**Федеральное государственное автономное образовательное учреждение   
высшего образования**

**КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ИНФОРМАЦИОННЫХ СИСТЕМ**

Направление подготовки: 09.03.03 – Прикладная информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Реализация автоматизированной трансформации ритма музыкального аудио**

**Работа завершена:**

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Студент группы \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Соломатин

**Работа допущена к защите:**

Старший преподаватель Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Кугуракова

Директор Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Ф. Хасьянов

Казань – 2017 г.

Содержание

[Введение 4](#_Toc485364219)

[1. Определение подхода к процессу трансформации ритма 6](#_Toc485364220)

[1.1 Понятие ритмического аудио 6](#_Toc485364221)

[1.2 Масштабирование по времени 7](#_Toc485364222)

[1.3 Представление метода ритмической трансформации 8](#_Toc485364223)

[2. Процесс ритмической сегментации 10](#_Toc485364224)

[2.1 Принципы ритмической сегментации 10](#_Toc485364225)

[2.2 Функция комплексного выявления атак спектральной разницей 11](#_Toc485364226)

[2.3 Определение локаций долей 14](#_Toc485364227)

[2.4 Пересэмплирование и извлечение ритмического паттерна 14](#_Toc485364228)

[3. Сопоставление ритмических паттернов 17](#_Toc485364229)

[3.1 Проблемы при сопоставлении паттернов 17](#_Toc485364230)

[3.2 Обоснованность синхронизированной трансформации 17](#_Toc485364231)

[3.3 Принцип сопоставления паттернов 18](#_Toc485364232)

[4. Масштабирование по времени 20](#_Toc485364233)

[4.1 Дальнейший шаг метода 20](#_Toc485364234)

[4.2 Вокодеры 20](#_Toc485364235)

[4.3 Локировка фазы 21](#_Toc485364236)

[4.4 Применение масштабирования по времени 22](#_Toc485364237)

[5. Результаты 24](#_Toc485364238)

[5.1 Метод оценки метода 24](#_Toc485364239)

[5.2 Алгоритм определения жанров 24](#_Toc485364240)

[5.3. Проведение тестирования 25](#_Toc485364241)

[6. Обсуждение и вывод 27](#_Toc485364242)

[Заключение 28](#_Toc485364243)

[Список литературы 29](#_Toc485364244)

## Введение

Данная работа проводится в рамках лаборатории Digital Media Lab. Digital Media Lab (DML) – лаборатория визуализации и разработки компьютерных игр.

Цель данной дипломной работы заключается в исследовании применения автоматизированной трансформации ритма музыкального аудио  при производстве цифрового музыкального контента. Под «музыкальным аудио» имеется в виду партия ударных инструментов, ведь именно эти инструменты создают основную ритмическую составляющую всей композиции. Для человека, играющего ударную партию, крайне важно поддерживать постоянный темп, бить по инструментам в рамках этого темпа. Иногда ударник играет заранее придуманную партию (например до этого созданную из электронных ударных звуков в какой-либо цифровой рабочей станции – программе, в которой происходит создание музыки, её запись и сведение), в таком случае любое отклонение от этой партии – воспроизведение звука чуть раньше или чуть позже намеченного момента – нежелательно. Отсюда истекает проблема: не всегда записанное аудио музыки соответствует тому, что задумывалось, и приходится затем вручную трансформировать позиции ударных внутри аудиофайла.

Стоит сразу отметить, что в наши дни музыка абсолютно всегда проходит через цифровую обработку, иначе просто невозможно подготовить её к выпуску онлайн или на цифровых носителях. В цифровую обработку как минимум входит сведение (объединение отдельных записанных треков в единую композицию) и мастеринг (обработка сведённой композиции для её выпуска «в мир»), но так же очень часто проводят коррекцию аудио – удаление артефактов (звуковых «ошибок»), изменение длины определённых фрагментов, их громкости. Это самые базовые действия, которые обязательно нужно проводить при коррекции аудиофайлов. Многие из которых входят и в обработку ударных партий. Если требуется трансформировать ритмический рисунок ударной партии, то без этих действий это сделать просто невозможно, и именно эти действия (в данном случае это передвижение фрагментов, изменение их длины, высоты) войдут в автоматизированную обработку ритма. Поэтому смело можно сказать, что проблема актуальна. Она так же практически значима, так как эти действия занимают много времени, если проводить их вручную.

Таким образом, поставлена задача нахождения рационального метода автоматизированной трансформации ритма музыкального аудио.Для достижения поставленной задачи были сформулированы следующие подзадачи:

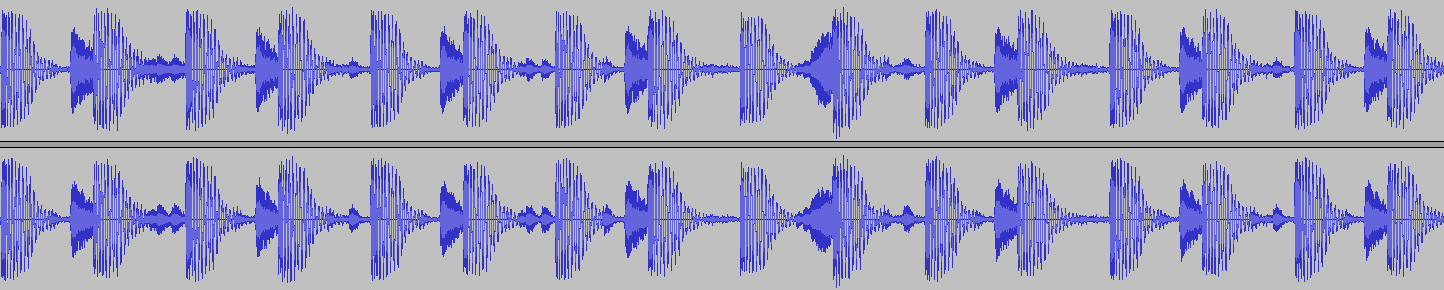
1. Определение подходов к процессу трансформации ритма;
2. Анализ ритмической сегментации;
3. Анализ способов сопоставления ритмических паттернов;
4. Анализ способов масштабирования по времени;
5. Оценка и сравнение результативности методов.

В итоге в данной работе будет описана методика автоматической синхронизации ритмических паттернов между двумя музыкальными сигналами. В трансформации, изначальный сигнал получит темп, метрику, размер, ритмическую структуру модельного сигнала, при этом сохраняя выступы внутри тактов и сильные доли. Модельным сигналом может служить, например, электронная партия ударных, на основе которой и создавалась «живая» ударная партия.

# Определение подхода к процессу трансформации ритма

### 1.1 Понятие ритмического аудио

Ритм состоит из, в большинстве случаев, постоянных пульсов, последовательности пауз и звуков, распределённых по тактам – единицам музыкального метра (см. рисунок 1). Рассматривая их вместе, а именно прослушивая их, создаётся ощущение движения в музыкальной композиции. В рамках автоматизированного извлечения ритма требуется определить структуру этих пульсов, в первую очередь определив музыкальный размер композиции (то есть число долей в такте) и границы тактов, а затем отметив события внутри этих границ.



*Рисунок 1. Файл партии ударных инструментов*

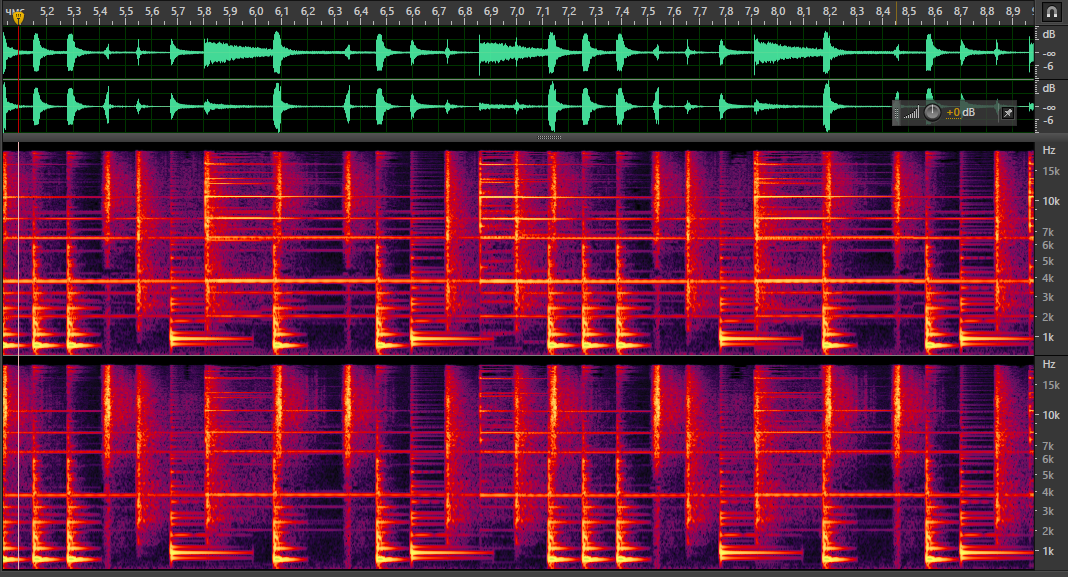
Трансформация ритма вручную проходит в звуковом редакторе и занимает много времени. Звуковой файл разрезается на ударные, которые можно определить по форме звуковой волны на глаз и при прослушивании. После этого требуется каждый фрагмент поставить на новое место внутри такта в соответствие с модельным сигналом. Но со временем возросла потребность автоматической сегментации и трансформации ритма, и в таких цифровых рабочих аудиостанциях как Ableton Live появились инструменты для подправки расположения ударных инструментов в их партии, которые лучше всего работают в монофонических (одноканальных) звуковых файлах или в ограниченных полифонических аудио, но нет методов, которые могли бы решить сложности с трансформацией более комплексного музыкального аудио.

### 1.2 Масштабирование по времени

Часто для автоматизированной синхронизации используется масштабирование по времени – смена скорости и длительности аудио сигнала, не затрагивая высоту тона. Однако, если разница между новым полученным темпом и темпом оригинального файла слишком большая, то транзиентные регионы аудио – очень короткие фрагменты с высокой амплитудой, которые находятся, например, в начале ударных – становятся очень смазанными или наоборот, сжатыми, теряя при этом характеристики, делающие их транзиентными. Также ухудшается качество звука. Ещё одно отрицательное последствие может быть в том, что ритмические структуры оригинального и изменённого аудио не обязательно совпадут, так как эти структуры пропорционально изменяются в процессе масштабирования по времени.

Решения этих проблем частично можно добиться, если масштабировать по времени с помощью адаптивного фазового вокодера [1]. Такой вокодер при масштабировании по времени сохранит атаки (периоды начального нарастания громкости сигнала) и транзиентные места аудио при помощи объединения локального фактора масштаба и процесса локировки фазы. Результатом такого адаптивного метода является возможность изменять темп аудио в большей степени без заметных изменений транзиентных участков.

Автоматизированную ритмическую модификацию можно реализовать посредством определения всех ударных событий в аудиофайле и классифицирования их по высоте тона – высокие, средние и низкие звуки [2] (см. рисунок 2). Алгоритм распознавания паттернов определит наилучшую последовательность, которая бы подошла из сегментов первого сигнала ко второму. Транзиентные регионы с частью последующего аудио сохраняются, как и в предыдущем методе [3]. Тем не менее, этот метод больше подойдёт для незамысловатых партий перкуссии, а процесс извлекания отдельных временных моментов для каждого ударного и их категоризация не подойдёт для полифонического аудио, в котором чёткое разграничение ударных не гарантируется.



*Рисунок 2. Спектральный вид партии перкуссии (колокольчики, маракасы)*

Ещё один подход – перекрёстное сопоставление, в котором синхронизация сигналов музыкального аудио достигается двумя этапами [4]. Сначала определяется темп и музыкальный размер аудио посредством определения сильных долей. Два сигнала затем выравниваются масштабированием по времени регионов между отмеченными ранее сильными долями. Полученная трансформация даёт метрически построенный сигнал, однако никаких выравниваний ударных компонентов, которые находятся внутри самих тактов, не сделано.

### 1.3 Представление метода ритмической трансформации

Метод, описанный в данной дипломной работе, объединит метрическое построение сильных долей перекрёстного сопоставления с сохранением транзиентных регионов локировкой фазы [4], а также включит в себя структурные изменения внутритактовых событий [3].

Эта автоматизированная ритмическая синхронизация достигается несколькими этапами обработки сигнала. Сначала для каждого сигнала мы извлекаем местонахождения слабых и сильных долей, из которых генерируются длительности тактов и доминирующие ритмические рисунки. Затем эти ритмические рисунки оцениваются для обоих сигналов, сравнительный процесс идентифицирует структурно значащие внутритактовые компоненты оригинального сигнала, которые совпадают с модельным сигналом. Эти компоненты потом перестраиваются по модели. Наконец, регионы внутри структурно значащих точек оригинального сигнала адаптивно масштабируются по времени, чтобы вместиться в длину тех же точек в модели, сохраняя при этом транзиентные регионы.

Далее в дипломной работе рассматривается анализ ритмической сегментации, подход к сопоставлению ритмических паттернов и алгоритм временного масштабирования.

# Процесс ритмической сегментации

### 2.1 Принципы ритмической сегментации

Чтобы модифицировать параметры ритма музыкального аудио сигнала требуется сначала локализировать начальные точки событий внутри музыки. К концу этого процесса было бы приемлемо следом применить процесс обнаружения отдельных ударных инструментов [5], так как он используется и в масштабировании по времени [3], и в существующих методах модификации ритма [4, 6]. Обнаружение атак ударных даёт временную сегментацию вводного сигнала. Однако в этом методе не обязательно обнаружить все атаки. В дальнейшем, абсолютно точное представление отдельных атак невозможно внутри музыкального аудио из-за случайного распределения нот во входном сигнале. Вместо этого, требуется только локализовать те атаки ударных, которые затем сами будут образовывать ритмический рисунок, присутствующий в аудио входного сигнала. Цель в том, чтобы получить именно ритмическую сегментацию, для которой мы будем использовать выявление сильных долей, отслеживание ударных и расчёт преобладающих ритмических паттернов [7].

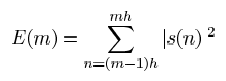
Первая стадия всего этого анализа – трансформировать входящий аудиосигнал в представление среднего уровня, которое больше подойдёт для ритмического анализа. В данном случае имеется в виду, что сигнал среднего уровня должен содержать в себе и показывать пиковые моменты на атаках ударных, где пиковые моменты представляют метрически более сильные события. Для этой цели используется функция комплексного выявления атак спектральной разницей Г(m) [5], где сэмпл m каждой функции выявления равен 11.6 мс. Хотя бывают функции выявляения атак другого типа, которые признаны более действенными, именно этот тип выявления атак ведёт себя лучше, когда требуется локализация долей.

### 2.2 Функция комплексного выявления атак спектральной разницей

Как уже было условлено, ударные ноты начинаются с так называемых транзиентных атак – коротких и непредсказуемых сегментов, охарактеризованных быстрыми изменениями в интенсивности энергии, высоты тона или тэмбра звука. За этими сегментами следует устойчивый сигнал, который легко предсказуем. Атака может быть определена как только начинается транзиент, то есть начинается нота. Обычно, схемы обнаружения атак – это подходы, основанные на работе с энергией сигнала и его частотами.

Обычно, вступление новой ноты ведёт к увеличению энергии сигнала (общей спектральной массы сигнала). В случае сильных, громких перкуссивных атак, таких как атак ударных, увеличение энергии сигнала будет очень резким и ясно видимым при просмотре волны сигнала. По этой причине, энергия сигнала является полезной метрикой для обнаружения перкуссивных транзиентных регионов, а также других типов атак нот.

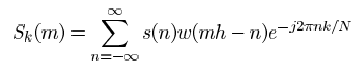
Локальная энергия кадра сигнала s(m) (наименее измеримого отрезка сигнала), можно определить как



, (1)

где h является размером скачка в амплитуде сигнала, m – номером скачка, n – суммарной переменной. Учитывая первую разницу функции E(m), можно обнаружить пиковые участки сигналы, из которых выбираются локации атак. Это один из самых простых способов обнаружения атак нот. Идею можно развить, чтобы учесть кадры быстрого преобразования Фурье (FFT).

Следует учесть расчёт сигнала s(n) принципом оконного преобразования Фурье (STFT):

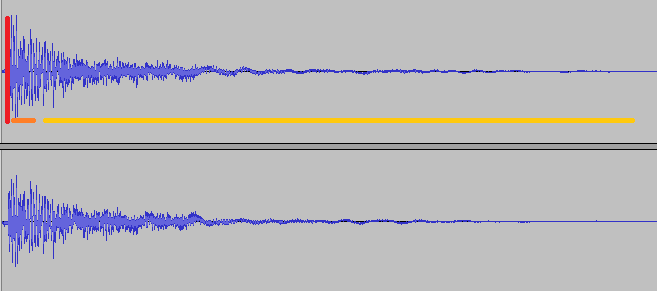


, (2)

где k = 0, 1, .., N-1 является частотным индексом и w(n) является окном определённой длины. Из этого следует, что величина сигнала равна

(3)

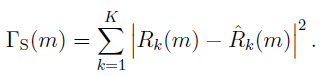
.



*Рисунок 3. Красная линия отображает атаку ударного инструмента, оранжевая – транзиентный регион, жёлтая – стабильный регион без резких изменений в спектре или громкости.*

Следом за транзиентным регионом ударного звука следует устойчивый легко предсказуемый отрезок сигнала, у которого величина спектра должна быть примерно постоянной (см. рисунок 3). Поэтому спектральную величину R^k(m) на момент кадра m можно приблизить к значению спектральной величины предыдущего кадра, Rk(m - 1). Функция спектральной разницы тогда будет равна

(4)



Эта мера может быть использована, чтобы построить эффективную функцию определения атак, если связать её с методом фазового отклонения, так как требуется также проанализировать фазовый спектр.

На устойчивых регионах, фазовая скорость на k-ой ячейке должна быть примерно постоянной, как и в случае спектра:

(5)



Тогда можно сделать предположение о фазе k-ой ячейки для кадра m, учитывая значения предыдущих двух кадров:

, (6)



где princarg принимает значение фазы в рамках отрезка [-π, π]. Разница между предполагаемым фазовым значением и наблюдаемым фазовым значением для кадра m требуется для формирования функции фазового отклонения Гф(m):



(7)

Теперь свяжем этот метод и спектральный метод для определения функции комплексного выявления атак спектральной разницей Г(m), которая будет чувствительна к перкуссивным и тональным атакам.

Предположения по спектральной величине R^k(m) и по фазовому спектру ф^ k(m) и функции спектральной разницы Гs(m) и фазового отклонения Гф(m) дают следующую объединённую формулу:

(8)



Сравнивая эту формулу с формулой наблюдаемого спектра , образуется функция Г(m):



(9)

Подобные алгоритмы работают быстро и легко встраиваются. Однако их эффективность не так высока, когда алгоритмы имеют дело с неперкуссивными сигналами, или когда транзиентный спектр сочетается с другими элементами в музыкальной композиции. Всплески энергии, соотносящиеся с транзиентной информацией, более заметны на высоких частотах, так как «тональная» энергия обычно сконцентрировала на более низких частотах, маскируя эффект этих вариаций на cодержимом сигнала [8].

### 2.3 Определение локаций долей

Следующий этап – требуется определить локации долей γb. Снова используется существующий алгоритм отслеживания долей [7].

Данный алгоритм действует сначала находя ритмические периодичности, исходя из функции автокорреляции кадров алгоритма нахождения атак, и затем, учитывая подходящую оценку периодичности сигнала, происходит выравнивание отрывков импульсов отображаюзих периодичность долей для функции местонахождения, чтобы восстановить фазы их появления. Самая важная особенность этого алгоритма в использовании двухэтапной модели. Идея состоит в том что алгоритм следует следующей интуитивной идее: когда слушатель набивает темп долей под музыку, будто метроном (задание, которое пытается повторить сам алгоритм), он пытается успевать за темпом и подстраиваться под ритмические события в музыке. Однако, в регионах где темп воспринимается как примерно постоянный, внимание перемещается на «набивку» долей ровно в то время когда мы ожидаем их услышать.

Доли извлекаются двухшаговым процессом. Сначала дольный период (время между долями) определяется с помощью гребёнчатой фильтрации автокорреляции функции Г(m). (Гребёнчатый фильтр — в обработке сигналов электронный фильтр, при прохождении сигнала через который к нему добавляется он сам с некоторой задержкой. В результате получается фазовая компенсация). Затем расположение долей оценивается гребёнчатой фильтрацией Г(m), учитывая дольный период, то есть предыдущий результат. В расчёт берётся представление сигнала среднего уровня. Затем описывается включение контекстно-связанной информации по поводу темпа, музыкального размера и местонахождения предыдущих долей для выявления этих параметров.

Когда локации долей были определены (γb), сильные доли (γd) выявляются измерением спектрального различия между синхронизованными с долями спектральными кадрами, например один спектральный кадр в доле. Долевые переходы, постепенно ведущие к наибольшему спектральному изменению, идентифицируются как сильные доли.

Чтобы извлечь преобладающий ритмический паттерн Гp(m), нужно переанализировать функцию выявления атак Г(m), используя локации долей γb и γd. Затем требуется разделить Г(m) на индивидуальные такты, Гd(m):

*Гd(m) = Г(m) γd ≤ m < γd+1* (10)

### 2.4 Пересэмплирование и извлечение ритмического паттерна

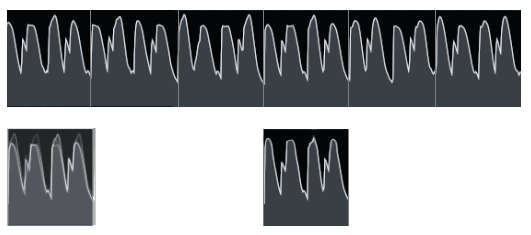
После этого требуется пересэмплировать каждый выявленный такт функции Г(m), чтобы у них была нормализованная длина L (где L = 144 сэмплов функции выявления). Частота сэмплирования аудиофайлов составляла 44100, 16000 или 11025 Гц. Кадровая частота была установлена так, чтобы такт содержал в себе определённое количество сэмплов любого темпа (темп известен заранее); это количество обозначается переменной b. Пусть, например, x(n) является входным сигналом с частотой сэмплирования r и длиной такта l секунд, тогда его огибающая амплитуды вычисляется с частотой сэмплирования b сэмплов в такте, используя размером скачка h, где h = r \* l / b.

Огибающая амплитуды y(n) вычисляется как:



, (11)

где k является фактором наслоения (когда спектр сигнала наслаивается на другой спектр). Длительности тактов l ранжировались от 0.97 до 3.30 секунд. Наилучшие результаты были получены при b = 72 и k = 2, хотя значения b от 48 до 144 давали похожие результаты [9].



*Рисунок 4. Наверху: функция ритмического анализа выявления атак Г(m) с сильными долями γd , нарисованными непрерывистыми линиями. Слева снизу: кластер паттернов длительностью в такт. Справа снизу: преобладающий ритмический паттерн Гp(m).*

Результат Гd(m) затем переформировывается в кластер, используя метод k-means, где k – заранее известное число кластеров, на которое разобъётся множество элементов векторного пространства – равно трём. Основная идея метода k-means заключается в том, что на каждом шаге перевычисляется центр масс каждого кластера, полученного на предыдущей итерации. Затем векторы разбиваются на кластеры опять в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения центра масс кластеров. Это происходит за конечное число итераций. Преобладающий паттерн выявляется в виде временного среднего наибольшего кластера [9] (см. рисунок 4).

Чтобы найти события ритмической последовательности p внутри последовательности длиною в такт, нужно произвести процесс извлекания частот пиковых значений функции Гp(m), используя подход выявления атак [5]. На последнем шаге события паттернов привязываются с извлечёнными сильными долями γd посредством линейной интерполяции:

*γd,p = γd + p\*((γd+1 - γd)/L) при d = 1, … , D-1.* (12)

# Сопоставление ритмических паттернов

### 3.1 Проблемы при сопоставлении паттернов

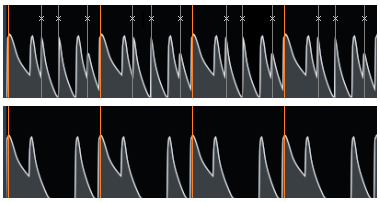
Изначальная цель состоит в том, чтобы привязать события ритмического паттерна изначального сигнала А с событиями модельного сигнала B и адаптивно масштабировать регионы, размеченные точками паттернов (ритмическими частями). Поэтому ранее упомянутая ритмическая сегментация производится на обоих сигналах. Так как реализация осуществлена таким образом, что она оперирует по произвольным сигналам, между сигналами часто несбалансированное число внутритактовых сегментов паттерна. Поэтому до модификации временных локаций точек паттерна обязательно требуется обратиться к внутритактовым событиям паттерна внутри двух сигналов. Если, например, выбранный размер из А содержит на n больше событий ритмического паттерна чем в В, и никаких модификаций не сделано чтобы решить это несоответствие, эти n точек паттерна станут первой последовательностью событий внутри последующего измерения. Этот нежелательный эффект так называемого завёртывания (wrapping) станет причиной того, что длина трансформированных тактов сигнала А будет отличаться от тактов целевого сигнала В, так как сильные доли изначальной последовательности не сохранятся.

### 3.2 Обоснованность синхронизированной трансформации

Обоснованность метрически синхронизированной трансформации подкрепляется расширением ритмического паттерна для включения метрических событий и событий на уровне тактов, так как вывод должен в наилучшем случае содержать в себе пульс, схожий с модельным. Локации долей определяются как периодические, сильные события внутри функции выявления, и поэтому крайне вероятно, что времена долей уже встроены в ритмический паттерн. Объединение времён долей также позволяет убрать метрический компонент из стадии сопоставления, упрощая модель, перемещая приоритет на анализ по уровню долей и количеству событий паттерна в доле.

### 3.3 Принцип сопоставления паттернов

Чтобы удостовериться в синхронизации сильных долей, аудиосигналы усекаются таким образом, чтобы оба сигнала А и В начинались именно со старта такта и заканчивались именно на конце такта. Понадобится изменить индексацию равенства (12), чтобы обращаться к индивидуальным внутритактовым сегментам. Для каждого такта в А будет Q событий паттерна в доле, где q-ое событие паттерна в b-ой доле сигнала A будет называться Ab,q; аналогично, если R событий в доле сигнала B, r-ое событие паттерна в b-ой доле сигнала B будет обозначаться как Bb,r. Местонахождения долей являются первыми событиями паттерна в доле, то есть Ab,1 (см. рисунок 5).



*Рисунок 5. Сопоставление паттернов ритма. События долей сохраняются. Там, где в одном из сигналов в такте больше событий паттерна чем в другом, слабейшие события удаляются.*

Как было сказано, внутри каждой доли b в сигналах A и B нет гарантии, что будет одинаковое количество паттерновых событий (то есть возможно, что R ≠ Q). Решить эту проблему можно игнорируя лишние события, выбирая их по относительной силе. Измерение силы определяется сэмплированием функции Г(m), с числом лишним событий паттерна в доле на каждое паттерновое событие γb,p. Слабейшие события px потом итеративно удаляются при обнаружении, используя следующую формулу:

*px = arg minp Г(γb,p)* (13)

На рисунке 4 отмечен процесс удаления слабых позиций паттерна из каждого внутридолевого интервала и последующие выравнивание остаточных выступов. Местонахождения долей сохраняются без исключений, как показано вертикальными линиями в обоих паттернах А и В, в то время как лишние события паттерна, обнаруженные с помощью равенства (13) отмечаются «х».

Когда события паттерна на долю были сбалансированы, изменяется нотация, чтобы отразить события паттерна P на долю в обоих сигналах, которые теперь называются Ab,p и Вb,p и проходят на стадию временного масштабирования, чтобы потом реализовать ритмическую трансформацию.

# Масштабирование по времени

### 4.1 Дальнейший шаг метода

Теперь требуется итеративно выделить каждый сегмент ритма, как было определено в событиях ритмического паттерна сигнала А, и масштабировать его по времени для совпадения по времени с соответствующим сегментом в сигнале В. Чтобы произвести это масштабирование по времени используется адаптивный метод локировки фазы посредством, как уже упоминалось ранее, фазового вокодера. Требуется рассмотреть принцип работы вокодеров и метод локировки фазы.

### 4.2 Вокодеры

Вокодеры (от англ. voice coder – кодировщик голоса) изначально использовались для синтеза речи на основе произвольного сигнала с богатым спектром (спектр звука графически представляет распределение энергии звука по частотным компонентам). В кодировщике, входной сигнал проходит через многополосной фильтр, затем каждая полоса проходит через детектор огибающей. Этот детектор берёт высокочастотный сигнал в качестве ввода и затем предоставляет огибающую оригинального сигнала. Контрольные сигналы из детекторов передаются через декодер. Декодер затем применяет эти контрольные (амплитудные) сигналы соответствующим фильтрам для ресинтеза [10].

Если рассматривать вокодер как музыкальный эффект, то он способен перенести свойства одного (модулирующего) сигнала на другой сигнал. Фазовый вокодер – это такой тип вокодера, который может масштабировать одновременно частотные и временные области, используя информацию о фазе. Его компьютерный алгоритм позволяет реализовывать на звуковом файле изменения по частотным областям, например изменение длительности и высоты сигнала, которые нужны для масштабирования по времени.

В основе фазового вокодера лежит принцип оконного преобразования Фурье (STFT). В дискретном варианте такого принципа, преобразуемые данные могут быть разделены на кадры. Каждый кадр преобразуется посредством STFT, и сформированный результат добавляется в матрицу, которая записывает амплитуду и фазу в каждый момент времени и частоты. По сути, этот принцип преобразует временное представление звука в частотно-временное представление («аналитическую фазу»), на котором можно производить модификации амплитуд или фаз определённых частотных компонентов звука до ресинтеза частотного представления во временное представление принцип обратного преобразования STFT. Развитие временного параметра ресинтезированного звука может быть изменено модификацией временной позиции кадров STFT, на которые делится звуковой сигнал, до самого ресинтеза, и именно это позволяет масштабировать по времени оригинальный звуковой файл [11].

### 4.3 Локировка фазы

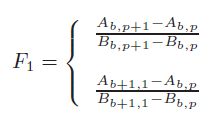
Чтобы решить проблему «смазанности» транзиентных регионов, на них производится алгоритм временно маскированной локировки фазы. Алгоритм локирует, иначе говоря связывает, фазу синтеза с аналитической фазой во временно маскированном транзиентном регионе. Преимущество этого алгоритма в том, что для многих сигналов фазовая ошибка не распространяется на такое количество кадров, чтобы она стала заметной или важной до того как фаза снова залокируется. Этот подход особенно полезен когда требуется расстянуть (увеличить длительность) сигналов, которые включают в себя и атаки как перкуссивного, так и более цельного характера, для которых многие другие методы не дают желательного результата.

Сама локировка фазы производится с помощью предыдущей фазы синтеза psi(sRa; k) вместо предыдущей аналитической фазы fi(sRa; k) в рамках расчёта нераскрытой фазовой разницы (т.е. разницы между фазой текущего рассматриваемого момента и фазой предыдущего рассматриваемого момента). Это значение затем используется в первом кадре (там, где сигнал не расстянут) перкуссивного транзиентного региона чтобы локировать фазу, таким образом сохраняя транзиентные характеристики атаки.

Временная маскировка применяется до процесса, чтобы отделить транзиентные и нетранзиентные регионы от входного сигнала. Дискретное выявление атак ударных применяется на входном аудио сигнале; транзиентные регионы определяются как первая треть каждого внутреннего атакового интервала, при условии, что минимум транзиентного региона всегда выше чем 11.6 мс и меньше 50 мс. Временное масштабирование потом применяется только на устойчивых регионах входного сигнала, чтобы предотвратить смазывание транзиентных регионов [3].

### 4.4 Применение масштабирования по времени

Учитывая, какая ритмическая сегментация получилась на соответствующем этапе, этап дискретного выявления атак не требуется. Каждая ритмическая часть имеет выявленную определённую атаку Ab,p и атаку, определённую либо как следующее событие паттерна для текущей доли Ab,p+1, либо следующей доли Ab+1,1. Внутри каждой части высчитывается линейный фактор масштабирования F1 в качестве пропорции длительностей между текущими событиями паттерна в А и событиями в В:



, (14)

Первый случай при p<P, второй случай при p>P или p=P.

Нелинейный фактор масштабирования F2, который ответственен за временную маскировку, высчитывается таким образом:



, (15)

где Wnt является шириной нетранзиентного региона, Wt является шириной транзиентного региона. Каждая ритмическая часть, масштабированная по времени, затем сцепляется с предыдущей частью, чтобы в итоге сформировать итоговый трансформированный сигнал [3].

# Результаты

### 5.1 Метод оценки метода

Так как оценка методов предварительной обработки и аспектов масштабирования по времени уже была проведена [3, 7], требуется лишь оценить аспект ритмической трансформации, что, впрочем, является самой важной частью всей работы. Оценка направлена на задачу классификации ритмических фрагментов по жанрам. Требуется понять, каким образом алгоритм определения жанра аудиофрагмента выберет жанр у изменённого ритмического отрывка. Например, если жанр модельного сигнала – самба, а у трансформируемого – танго, то какой жанр определится у изменённого сигнала – жанр оригинального сигнала или изменяемого?

Используется существующий алгоритм жанрового анализа [8], который использует ритмические последовательности длительностью в такт чтобы охарактеризовать ритмические свойства входящего сигнала, чтобы максимизировать похожесть между отрывками того же жанра и при этом минимизировать похожесть среди разных жанров. Вычисляемой характеристикой в данном случае является автокорреляционная функция функции определения атак, описанная во второй главе. Затем она усекается с тактовой периодичностью (которая определяется темпом и числом долей в такте) и пересэмплируется в фиксированную длительность. Каждому отрывку в тестовой базе данных затем даётся определённый жанр, и затем отрывки передаются в алгоритм под названием WEKA [12], который и совершит жанровую классификацию [8].

### 5.2 Алгоритм определения жанров

Weka предоставляет реализацию обучающихся алгоритмов которые можно применять на базах данных. Сюда также включён набор инструментов для трансформирования баз данных. Возможно провести предварительную обработку данных, загрузить результаты в обучающуюся схему и проанализировать получающийся классификатор без написания и единой строки кода.

Один из способов использовать Weka – это применять обучающийся метод на БД и анализировать результат чтобы узнать больше о данных. Также можно использовать изученные модели чтобы делать предположения о иных случаях каких-либо ситуаций, или использовать для анализа сразу несколько классификаторов и сравнить их производительность. Классификаторами в данном случае являются обучающиеся методы. У многих классификаторов есть настраиваемые параметры, и для всех классификаторов есть оценочный модуль [12].

### 5.3. Проведение тестирования

Для проведения тестирования была создана база из 120 танцевальных композиций шести жанров: Jive, Quick Step, Tango, Samba, Cha Cha, Rumba (по 20 композиций каждого жанра). Все эти шесть жанров являются жанрами латиноамериканской танцевальной музыки, но они достаточно различаются чтобы с ними мог работать алгоритм определения жанров. Для каждого жанра случайно выбирается обрабатываемый файл и модельный файл и производится трансформация (см. таблицу 1). Для аудиофрагментов указываются сильные и слабые доли. Чтобы предотвратить возможные неточности в ритмической предварительной обработке, алгоритму предоставляются метрические аннотации.

Жанр трансформированных сигналов (ХВА) обозначается так – сначала обозначается один из целевых жанров (GA), затем, в качестве контроля, обозначается так же жанр входящего сигнала (GВ). Затем эти данные передаются WEKA [12], который проводит классификацию. Общие результаты подведены в таблице.

Классификационная точность трансформированных сигналов, учитывая жанр модельных сигналов, гораздо выше, если сравнивать с точностью, учитывая жанр изначальных сигналов – 51,5% против 13,5%. Это означает, что трансформированные жанры ближе по жанру к модельным сигналам, чем до того как они были трансформированы. Также согласно результатам было выяснено, что самым тяжелым для обработки жанром является Rumba, а самым лёгким – Quick Step; скорее всего потому, что в этом жанре есть ритмическая характеристика под названием «свинг», которой не присутствует в остальных жанрах.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Жанр | Jive | Quick | Tango | Samba | Cha | Rumba |
| Jive | - | 19 | 12 | 8 | 13 | 5 |
| Quick | 7 | - | 10 | 3 | 1 | 2 |
| Tango | 7 | 14 | - | 14 | 10 | 7 |
| Samba | 13 | 16 | 5 | - | 15 | 8 |
| Cha | 12 | 9 | 14 | 17 | - | 9 |
| Rumba | 13 | 6 | 9 | 14 | 16 | - |
| Общий % | 53 | 64 | 50 | 56 | 55 | 31 |

*Таблица 1. Результаты классификации трансформированных композиций по жанрам. В каждом случае пробуется по 20 трансформаций. «Quick» сокращённо от «Quick Step»*

# Обсуждение и вывод

Изначально сложность метода состоит в предварительном ритмическом анализе. В случаях когда анализ ритма происходит успешно, трансформация ритма тоже ожидается успешной. Однако если, например, некорректно извлекаются доли, то весь последующий анализ скорее всего будет уже бесполезным. Если неправильно определились сильные доли, достойная ритмическая трансформация возможна, однако фазовая разница между сигналами будет более заметна, если произвести общий микс.

Обе проблемы можно решить при полуавтоматической версии описанной в дипломной работе трансформации, в которой пользователь мог бы вручную скорректировать местонахождения сильных и слабых долей; всё остальное бы произвелось самим компьютером. Если использовать этот инструмент в музыкальном пост-процессинге, это задача будет совсем несложной и не займёт много времени и сил.

# Заключение

В результате выполнения дипломной работы было исследовано применение автоматизированной трансформации ритма музыкального аудио  при производстве цифрового музыкального контента.

В процессе выполнения дипломной работы были решены следующие задачи:

1. Определены подходы к процессу трансформации ритма;
2. Изучена ритмическая сегментация;
3. Изучены способы сопоставления ритмических паттернов;
4. Изучены способы масштабирования по времени;
5. Оценены и сравнены результативности методов.

# Список литературы

[1] Дюксбери Ч. Signal models for polyphonic music, Ph.D. thesis - Department of Electronic Engineering, Queen Mary, University of London, 2014.

[2] Равелли Э., Белло Д. П., Сэндлер M. Б., “Automatic rhythm modification of drum loops”: IEEE Signal Processing Letters, том 14, № 4, C. 228–231. – 2017.

[3] Равелли Э., Белло Д. П., Сэндлер M. Б., “Fast implementation for non-linear time-scaling of stereo audio signals,” in Proceedings of the 8th International Conference on Digital Audio Effects (DAFx05) – Madrid, Spain, 2015 – С. 182–185.

[4] Джехан Т. Creating music by listening, Ph.D. thesis - School of Architecture and Planning, Massachusetts Institute of Technology, 2015.

[5] Белло Д. П., Дюксбери Ч., М. И. Дэвис, Сэндлер M. Б. “On the use of phase and energy for musical onset detection in the complex domain,” IEEE Signal Processing Letters, том. 11, номер. 6, С. 553–556. – 2014.

[6] Гоуйон Ф, Фабиг Л., “Rhythmic expressiveness transformation of audio recordings: swing modifications,” in Proceedings of the 6th International Conference on Digital Audio Effects (DAFx03) – London, United Kingdom, 2013 – С. 94–99.

[7] Дэвис М. Э. П. Towards automatic rhythmic accompaniment, Ph.D. thesis – Department of Electronic Engineering Queen Mary, University of London, 2013.

[8] Дэвис М. Э. П., Пламбли М. Д., “Exploring the effect of rhythmic style classification on automatic tempo estimation,” – 2012.

[9] Диксон С., Гоуйон Ф., and Уидмер Г., “Towards characterization of music via rhythmic patterns,” in Proceedings of 5th International Conference on Music Information Retrieval – Barcelona, Spain, 2004 – С. 509–517.

[10] Вокодер [Электронный ресурс]. - Режим доступа: <https://en.wikipedia.org/wiki/Vocoder>

[11] Принцип оконного преобразования Фурье [Электронный ресурс]. - Режим доступа: <https://en.wikipedia.org/wiki/Short-time_Fourier_transform>

[12] Уиттен И.Х., Франк Е., Data Mining: Practical machine learning tools and techniques – Morgan Kaufmann, San Francisco, 2nd edition. – 2005.