



ИПМ им.М.В.Келдыша РАН • [Электронная библиотека](#)

[Препринты ИПМ](#) • [Препринт № 48 за 2013 г.](#)



Корухова Ю.С., [Мытрова М.В.](#)

Информационный поиск для
нотных записей
музыкальных произведений

Рекомендуемая форма библиографической ссылки: Корухова Ю.С., Мытрова М.В. Информационный поиск для нотных записей музыкальных произведений // Препринты ИПМ им. М.В.Келдыша. 2013. № 48. 16 с. URL: <http://library.keldysh.ru/preprint.asp?id=2013-48>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

Ю.С. Корухова, М.В. Мытрова

**Информационный поиск
для нотных записей
музыкальных произведений**

Москва — 2013

Корухова Ю.С., Мытрова М.В.

Информационный поиск для нотных записей музыкальных произведений

Работа посвящена проблеме поиска нот музыкальных произведений, записанных в формате MusicXML. Данный формат является одним из самых распространенных и поддерживается многими программами – нотными редакторами. В большинстве современных поисковых систем имеется возможность искать ноты по названию и автору композиции. В данной работе предлагается подход к решению задачи поиска произведений по нотной записи фрагмента мелодии. Для поиска использованы волновые правила, традиционно применяемые в методах автоматизированного доказательства теорем.

Ключевые слова: поиск нотных записей, методы автоматических рассуждений, волновые правила

Yulia Stanislavovna Korukhova, Marina Vyacheslavovna Mytrova

Music scores information retrieval

The music information retrieval problem for sheet music stored as MusicXML files is considered. MusicXML is a widespread format for music scores; it is supported by various music composition and notation software. Most of the systems take a composer name and/or a name of composition as a query, and we are interested in content based retrieval, having a sequence of notes as their input data. We propose to adapt the automated reasoning method rippling for the task of music information retrieval.

Key words: music information retrieval, automated reasoning, rippling

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 12-01-31109-мол_а.

Оглавление

Введение	3
Внутреннее представление нотных записей.....	5
Сравнение музыкальных фрагментов	6
Индексирование библиотеки musicXML файлов.....	10
Реализация метода.....	11
Оценка работы метода	14
Заключение.....	16
Список литературы	16

Введение

В век информационных технологий практически в любой области данные начинают храниться и распространяться в электронном виде. Не является исключением и музыкальная область. Множество аудио, видео и текстовой музыкальной информации хранится на просторах сети Интернет, в библиотеках файлов, и закономерно возникает задача поиска такой информации. В данной работе мы ограничиваемся поиском в библиотеке файлов, содержащих нотные записи музыкальных произведений. Такие файлы, как правило, создаются с помощью программ – нотных редакторов, либо представляют собой отсканированные и распознанные специальной программой ноты. Многие нотные редакторы кроме своего формата предоставляют возможность сохранять нотные записи в формате MusicXML¹. Данный формат поддерживается более чем 150 программами-редакторами (MuseScore, Finale, Guitar Pro и др.), которые позволяют не только читать нотные записи, но и вносить в них изменения. Например, перевод мелодии в другую тональность может быть выполнен по запросу пользователя автоматически.

В настоящее время поисковые системы позволяют находить текстовые документы по разным параметрам: не только по их явно заданным характеристикам (название, автор, жанр, год издания), но и по содержимому. В электронных нотных библиотеках возможности поиска обычно скромнее: поиск ведется, как правило, по названию музыкального произведения или по фамилии автора². Но хочется большего: искать музыкальные файлы на основе анализа их содержимого. Для аудиофайлов уже реализованы методы поиска по звуковому фрагменту (акустическому отпечатку)³. Однако данные методы имеют ограничение на минимальную длительность аудиозаписи – входного фрагмента, по которому ведется поиск.

Поиск на основе анализа содержимого музыкальных файлов реализован в системе Musipedia⁴ [3]. Исходными данными для поиска в ней является фрагмент нотной записи, мелодический контур или ритм. В качестве результата выдается набор midi-файлов, в которых содержится заданный (или похожий на него) фрагмент. Система осуществляет сравнение внутреннего представления музыкальных фрагментов, что позволяет, в частности, находить мелодию, записанную в другой тональности, исходное представление которой отличается от запроса.

Для библиотек, хранящих ноты, также представляется полезной возможность поиска по содержимому файлов. Подход к анализу содержимого файлов с нотными записями предложен в системе Music Ngram Viewer⁵ [5],

¹ www.makemusic.com/musicxml

² Такой вид поиска присутствует, например, на сайтах www.wikifonia.org, www.notomania.ru

³ echoprint.me

⁴ www.musipedia.org

⁵ www.peachnote.com

анализирующей вхождение n -грамм (последовательностей из n музыкальных символов) в нотную запись произведений нескольких цифровых библиотек. На основе обнаруженного количества вхождений n -грамм из запроса в конкретное произведение принимается решение о степени его соответствия запросу.

Исходными данными для предлагаемого в данной работе метода поиска является запись фрагмента произведения в виде MusicXML файла, которая может быть сформирована заранее в нотном редакторе или же непосредственно введена с помощью web-интерфейса (виртуальной клавиатуры фортепиано). В качестве результата выдаются найденные в библиотеке файлы, содержащие искомые нотные записи, с указанием степени их соответствия запросу.

Запись нот в виде MusicXML файла представляет собой структурированный документ, однако традиционные методы поиска, разработанные для структурированных файлов [9], оказываются неприменимыми. В методе поиска должна учитываться специфика предметной области. Например, одна и та же мелодия, записанная в разных тональностях, представляется файлами с (формально) разным содержанием. При записи мелодии в более быстром темпе можно поменять информацию о темпе произведения, а можно записать его нотами другой длительности, что также приведет к появлению файлов с разным содержанием для одной мелодии. Поэтому для поиска предлагается использовать специальное представление нотных записей, позволяющее учесть эти особенности.

В информационном поиске мы сталкиваемся с задачей сравнения двух фрагментов (запроса и библиотечного файла), которые в чем-то являются похожими. Требуется оценить, насколько они похожи и допустимо ли считать их представлением одного произведения. Сходная ситуация возникает в математике при проведении доказательства методом математической индукции. При выполнении шага индукции имеется предположение, которое считается истинным, и требуется доказать истинность заключения индукции, которое похоже на гипотезу. Одним из методов, с помощью которого можно автоматизировать такое доказательство, является метод волновых правил [1, 2]. Метод предлагает сначала выделить общие части (основу) гипотезы и заключения и отметить различия, этот процесс называется аннотированием выражений. Затем применять в доказательстве только те правила, которые уменьшают различия в соответствии с некоторой мерой. При успешном исходе доказательства все различия будут устранены, заключение индукции сведется к гипотезе, и его истинность тем самым будет доказана. Для использования данного метода при поиске необходимо определить правила аннотирования запроса и библиотечного файла, определить правила, с помощью которых производятся преобразования, и задать меру.

Далее мы последовательно рассмотрим основные составляющие предлагаемого метода и его реализацию.

Внутреннее представление нотных записей

Предлагаемый метод поиска работает с файлами в формате MusicXML. Данный вид файлов является подмножеством XML файлов, в котором зафиксирован набор тегов разметки содержимого. Для описания информации о нотах в произведении введены, например, теги

`<note>`, `<octave>`, `<pitch>`, `<duration>`,

позволяющие выделить ноту, октаву, в которой она находится, высоту и длительность ноты соответственно. В данном формате переписывание имеющейся мелодии, например, на два тона выше приведет к изменению содержимого файла. Однако, с музыкальной точки зрения, мелодия остается той же. Поскольку задачей является именно нахождение похожих мелодий, необходимо такое внутреннее представление, которое оказывалось бы одинаковым для разных файлов, содержащих одну мелодию. Подходы к решению данной проблемы предложены в [3, 4, 7].

Музыкальный фрагмент предлагается представлять в виде нотных расстояний и пауз. Нотные расстояния хранятся в виде пар:

(относительная высота, относительная длительность),

паузы характеризуются только относительной длительностью.

Относительная высота ноты – это число, равное смещению (в полутонах) относительно предыдущей ноты фрагмента. Для первой ноты фрагмента понятие относительной длительности не вводится, но она участвует в вычислении относительной длительности второй ноты. Такое представление позволяет, во-первых, сделать одинаковыми представления одной мелодии в разных тональностях и, во-вторых, ошибка при вводе одной из нот проявится только в двух относительных высотах, но не повлияет на остальные ноты фрагмента.

Рассмотрим вычисление относительных высот на примере фрагмента "Польки" С.В. Рахманинова (рис. 1). Относительные высоты всех нот, начиная со второй, подписаны под соответствующими нотами: положительные числа появляются при движении мелодии вверх, отрицательные – в случае, если высота следующей ноты ниже предыдущей.



Рис. 1. Относительные высоты нот

Относительная длительность ноты – это отношение длительности ноты к длительности предыдущей ноты или паузы. Первая нота фрагмента участвует в вычислении относительной длительности следующей ноты, а для нее самой относительная длительность не вычисляется. Если ноты одной высоты связаны лигой, то во внутреннем представлении они считаются одной нотой, их

длительности складываются и относительная длительность вычисляется как общая для такой группы нот. Относительные длительности пауз вычисляются по таким же правилам.

Расстановка относительных длительностей для фрагмента "Польки" С.В. Рахманинова приведена на рис. 2.



Рис. 2. Относительные длительности нот

Данное представление позволяет получить одинаковую формальную запись для одинаковых мелодий, таким образом упрощая их сравнение. Однако одного лишь изменения представления оказывается недостаточно для решения задачи поиска. Наличие несовпадающих внутренних представлений не обязательно означает, что рассматриваемые музыкальные записи не относятся к одному произведению. Во-первых, несовпадение может быть вызвано разными длинами запроса и файла из нотной библиотеки. Во-вторых, при формировании запроса по слуху в нем могут отразиться особенности, характерные для конкретного инструмента, и если в библиотеке хранятся ноты для другого инструмента, запись будет выглядеть по-другому. Наконец, в некоторых нотах запроса могут появиться случайно сделанные пользователем ошибки. Поэтому поиска мелодии по точному соответствию на практике оказывается недостаточно, требуется искать также и похожие, не во всем совпадающие с запросом фрагменты.

Сравнение музыкальных фрагментов

Сравнение музыкальных фрагментов, уже переведенных во внутреннее представление, происходит в три этапа. Сначала выполняется поиск наибольшей общей подпоследовательности нот и специальная разметка (аннотация) фрагментов. Затем выполняются преобразования с помощью волновых правил. На последнем этапе вычисляется оценка похожести фрагментов.

Чтобы выяснить, имеет ли смысл доказывать похожесть двух фрагментов, они проходят предварительную проверку. Пусть у нас есть фрагменты F_1 и F_2 . Найдем их самую длинную общую подпоследовательность, используя алгоритм, описанный в [8]. При этом ноты последовательности не обязательно стоят подряд: общая подпоследовательность может быть образована из нескольких фрагментов. Длиной нотной последовательности будем называть количество нот и пауз в ее внутреннем представлении. Если длина самой длинной общей подпоследовательности нот (Longest Common Subsequence of

Notes или LCSN) не превышает половины длины меньшего из фрагментов, то похожесть фрагментов считается равной нулю, и дальнейшие действия не производятся. В противном случае составляется аннотированная формула

$$F_1[LCSN] = F_2[LCSN] \quad (1)$$

где $F[L]$ обозначает последовательность, содержащую L . Выражение $LCSN$, состоящее из общих символов левой и правой частей равенства, называется основой, а все остальные последовательности символов отмечаются как волновой фронт. Далее ставится задача устранения волнового фронта (различий). В процессе такого преобразования изменение основы ($LCSN$) не допускается, а волновые фронты можно изменять, перемещать или удалять в соответствии с заданными правилами преобразования – **волновыми правилами**.

Волновые правила представляют собой правила переписывания, в левой и правой частях которых отмечены общие части (основа) и различия (волновой фронт). Из набора волновых правил выбирается то, левая часть которого может быть унифицирована с подвыражением в рассматриваемой формуле, и это подвыражение заменяется на правую часть волнового правила. Существует два источника расхождений в сравниваемых фрагментах: это различия в способах записи нот для разных инструментов, разные аранжировки и ошибки, сделанные при вводе запроса. Соответственно, есть два источника появления правил: закономерности, известные из теории музыки, правила гармонии [11, 12] и анализ типичных ошибок, которые возникают при записи мелодии по слуху.

В данной работе рассматриваются три категории правил: правила альтернативной записи, правила исправления ошибок и правила сокращения.

1. *Правила альтернативной записи* позволяют учитывать сходство мелодии в вариациях и аранжировках. Один и тот же музыкальный фрагмент может быть записан и исполнен по-разному, при этом мелодия будет узнаваема. Для того чтобы привести к единому виду различные вариации одной мелодии, сформулированы правила, преобразующие альтернативные способы записи фрагмента друг к другу. Они предусматривают приведение друг к другу различных вариантов расстановки пауз, удаление “неосновных” звуков [11]. Таким, в частности, является правило, позволяющее заменить ноту и стоящую после нее паузу нотой большей длительности

$$[\dots](x,y)(y)[\dots] \implies [\dots](x,2*y)[\dots]$$

а также правила устранения вспомогательного неаккордового звука

$$[\dots](x,y)(x+1,y)(x,y)[\dots] \implies [\dots](x,3*y)[\dots]$$

$$[\dots](x,y)(x-1,y)(x,y)[\dots] \implies [\dots](x,3*y)[\dots]$$

и правила, устраняющие опевание

$$\begin{aligned} [\dots](x,y)(x-1,y)(x+1,y)(x,y)[\dots] & \implies [\dots](x,4*y)[\dots] \\ [\dots](x,y)(x+1,y)(x-1,y)(x,y)[\dots] & \implies [\dots](x,4*y)[\dots] \end{aligned}$$

Здесь $[\dots]$ – любая последовательность нот, (x,y) – нота волнового фронта относительной высоты x и относительной длительности y , (y) – пауза волнового фронта относительной длительности y .

2. *Правила исправления ошибок* предназначены для того, чтобы устранить возможные ошибки, допущенные пользователем при записи мелодии по слуху. Так, например, при подборе достаточно большого интервала (секста, септима) пользователь может ошибиться на полтона как в сторону уменьшения интервала, так и в сторону его увеличения. Поэтому такие расхождения в крайних нотах волновых фронтов, прилегающих к одинаковым частям основы, считаются опечаткой и приводятся к одной ноте. Для случая, когда такая ошибка возникает в волновом фронте слева от основы, правила записываются следующим образом:

$$\begin{aligned} [\dots] F(x,y) S[\dots] &= [\dots] F(x+1,y) S[\dots] \implies \\ & [\dots] F(x,y) S[\dots] = [\dots] F(x,y) S[\dots] \\ [\dots] F(x,y) S[\dots] &= [\dots] F(x-1,y) S[\dots] \implies \\ & [\dots] F(x,y) S[\dots] = [\dots] F(x,y) S[\dots] \end{aligned}$$

здесь $[\dots]$ – любая последовательность нот, S – последовательность нот основы, F – последовательность нот волнового фронта, (x,y) – нота волнового фронта относительной высоты x и относительной длительности y .

Аналогичные правила сформулированы и для ошибки, сделанной в волновом фронте справа от основы.

3. *Правила сокращения* оказываются полезными после применения других волновых правил, в результате которых могла увеличиться наибольшая общая подпоследовательность нот в заданных фрагментах. Так как по определению волновые правила должны сохранять основу, мы не можем добавить в нее появившиеся общие ноты (хотя они и стали во фрагментах одинаковыми), поэтому такие ноты удаляются специальными правилами. Как и в случае применения правил исправления ошибок, происходит удаление не всех общих нот волновых фронтов, а только тех, которые граничат с одинаковыми частями основы:

$$[\dots]aXYZc[\dots] = [\dots]aXFZc[\dots] \implies [\dots]aYc[\dots] = [\dots]aFc[\dots]$$

здесь $[\dots]$ – любая последовательность нот, a, c – последовательности нот основы, X, Y, Z, F – последовательности нот волнового фронта.

Также при преобразовании используются правила, удаляющее произвольное количество пауз [pauses], находящихся в начале или конце фрагмента:

$$[\dots] X [\text{pauses}] \implies [\dots] X$$

[pauses] X [...] ==> X[...]

При работе поисковой системы в ее базе содержится большое количество правил, многие из которых можно применять как в прямом, так и в обратном направлении. Случайный выбор правил может сделать бесконечным процесс преобразования или сильно увеличить время его выполнения. Поэтому нужно определить меру, с помощью которой можно “измерять” количество оставшихся различий. На каждом шаге после применения правила мера должна уменьшаться. Процесс преобразования останавливается в одном из двух случаев: либо когда мера становится равной нулю (то есть все различия в выражениях устранены), либо когда ни одно из правил преобразования не применимо. При работе с музыкальными фрагментами в качестве меры предлагается использовать количество нот в волновом фронте. В процессе преобразования выбираются и применяются только те правила, которые уменьшают длину волнового фронта, то есть делают выражения более похожими.

Оценка похожести фрагментов выполняется после завершения применения волновых правил. Похожесть фрагментов обратно пропорциональна длине волнового фронта, который остался после применения правил, а также зависит от количества примененных правил, поскольку правила применялись для устранения различий, то есть исходные фрагменты не были одинаковыми. Для оценки похожести фрагментов F_1 и F_2 мы предлагаем формулу

$$\text{similarity}(F_1, F_2) = \frac{L_1 - W_1 + L_2 - W_2}{L_1 + L_2 + \alpha * N} \quad (2)$$

где L_1 – длина фрагмента F_1 ,

L_2 – длина фрагмента F_2 ,

W_1 – длина волнового фронта фрагмента F_1 ,

W_2 – длина волнового фронта фрагмента F_2 ,

N – число примененных правил преобразования,

α – эвристический коэффициент.

Поскольку в музыкальной области не формализована оценка сходства мелодий (в основном оно определяется на слух, хотя часто и основано на наличии общих фрагментов), эвристический коэффициент подбирался экспериментально. Чем больше этот коэффициент, тем большую роль играет количество примененных правил преобразования. Практическое использование метода показало, что при $\alpha = 0.5$ оценка похожести, как правило, совпадает с оценкой, сделанной специалистом.

Результатом вычисления похожести фрагментов является значение из отрезка $[0, 1]$, где 0 соответствует полному несовпадению фрагментов, а 1 – их полному совпадению. При практическом использовании метода диапазон

возможных значений оказывается меньше, близкие к 0 значения получены не будут, так как сильно различающиеся музыкальные фрагменты были устранены ранее, после определения самой длинной общей подпоследовательности нот.

После прохождения указанных этапов поиска получается список композиций, которые сортируются в порядке убывания значений, вычисленных по формуле (2), и объявляются результатами поиска.

Индексирование библиотеки MusicXML файлов

Одна из задач, которую, как правило, требуется решить при создании поисковой системы, – это индексирование содержимого библиотеки для ускорения поиска.

Проблема индексирования музыкальных файлов заключается в том, что нельзя автоматически разбить произведение на фрагменты и построить полноценный словарь. Если для текстовых коллекций можно определить понятие "слово" как текст между пробельными символами, то аналогом его в музыкальной области была бы "музыкальная фраза" или "мелодия", границы которой сложнее определить формально. Мелодия может быть у одного из инструментов, она не всегда начинается с первого такта и не всегда ее границами являются тактовые черты. Если при работе с текстовым файлом разбить его на отдельные слова не составляет труда, то автоматическое разбиение музыкального произведения на осмысленные фрагменты является нетривиальной задачей. Для разбиения музыкального произведения используется метод "скользящего окна" [6]. Задается размер (суммарная длительность нот) окна и размер шага. В общем случае размер шага меньше или равен размеру окна, то есть последовательно просматриваемые окна могут перекрываться. Окно проходит с заданным шагом от начала произведения до его конца, тем самым формируя фрагменты, совпадающие по размеру с окном. Фрагменты музыкального произведения, содержащие репризы и вольты, проходятся окном столько раз, сколько это необходимо в соответствии с правилами игры по нотам.

Часть мелодии, выражающая законченную музыкальную мысль, в элементарной теории музыки называется периодом. Простейший тип периода состоит из восьми тактов. Период делится на две части по четыре такта, называемые предложениями [12]. Предложения, в свою очередь, можно разбить на двухтактные фразы [10]. Следуя такому принципу деления мелодии, произведение проходит трижды – по одному разу окнами размером в 8, 4 и 2 такта.

Полученные фрагменты переводятся во внутреннее представление и заносятся в словарь. В словаре они хранятся в виде инвертированного файла [9], то есть для каждого фрагмента указывается список произведений, в которых он встречается, и число вхождений в каждое из них. В силу особенностей сравнения музыкальных фрагментов, при занесении в базу и

подсчете вхождений фрагмента в произведение рассматривается не точное совпадение нового фрагмента и имеющегося в базе, а такое совпадение, при котором величина, полученная при вычислении формулы (2), не меньше порогового значения. В результате экспериментов для текущей программной реализации системы выбрано пороговое значение 0.9.

Реализация метода

Предложенный метод реализован в системе, выполняющей поиск нотных записей музыкальных произведений. В настоящее время система работает в тестовом режиме с библиотекой нотных записей на локальном компьютере. Система доступна по адресу music.keldysh.ru

Для ввода поискового запроса система имеет web-интерфейс, позволяющий набрать последовательность нот запроса на виртуальной клавиатуре (рис. 3). Также пользователь имеет возможность прослушать набранный фрагмент, чтобы проверить, правильно ли он записал интересующую его мелодию. Интерфейс предоставляет одну нотную строку, длины которой в большинстве случаев достаточно для ввода запроса. Кроме того, имеется возможность загрузить заранее созданный (например, с помощью нотного редактора) файл с поисковым запросом.

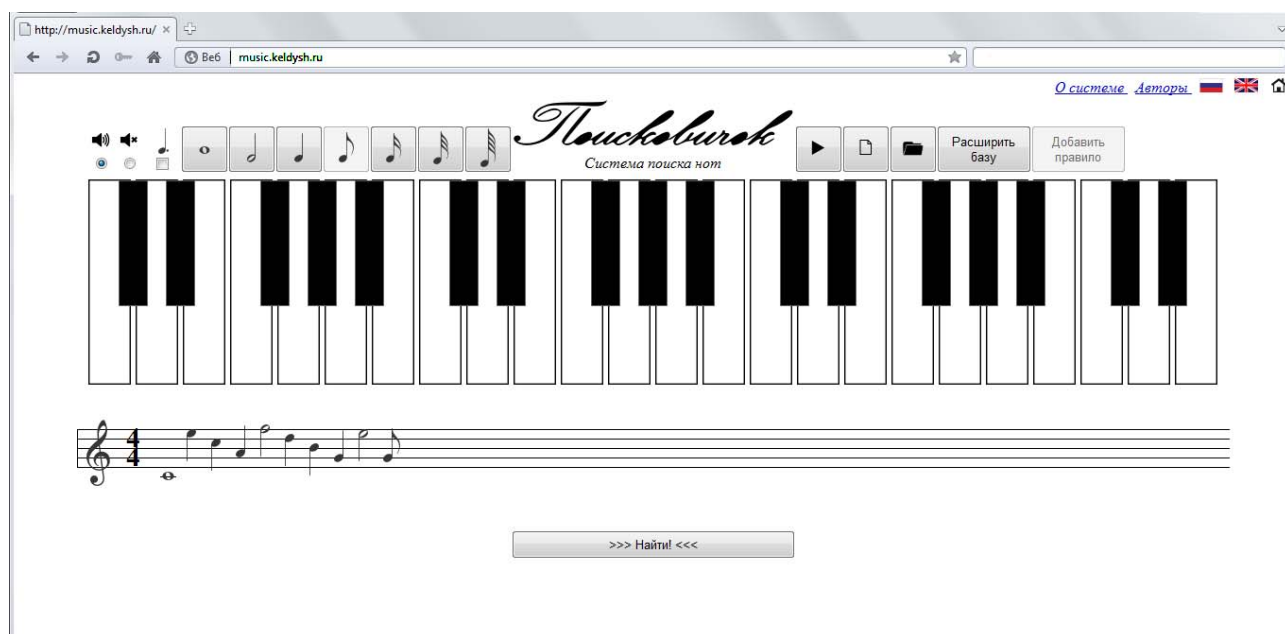


Рис. 3. Web-интерфейс для ввода поискового запроса

В качестве результата поиска на экране отображается список файлов с указанием степени их соответствия запросу. При добавлении нового файла с нотами в библиотеку запускается процедура индексирования: фрагменты произведения добавляются в базу данных.

Текущая реализация системы выполнена на языке PHP, для хранения фрагментов мелодий (словаря) используется база MySQL.

При обработке запроса происходит его сравнение с фрагментами, находящимися в словаре, с помощью волновых правил. Рассмотрим работу алгоритма сравнения мелодий на примере двух фрагментов "Europe – The Final Countdown", один из которых задан как запрос, второй хранится в словаре системы (рис. 4).



Рис. 4. Фрагменты "Europe – The Final Countdown" из разных источников

После вычисления относительных высот и относительных длительностей запишем их, начиная от одной и той же первой ноты (рис. 5).



Рис. 5. Внутреннее представление фрагментов "Europe – The Final Countdown", записанное от одной ноты

Далее система находит наибольшую общую подпоследовательность нот в заданных фрагментах и проводит их аннотирование – отмечает различия (рис. 6).



Рис. 6. Аннотированные фрагменты. На белом фоне – ноты основы, на сером фоне – волновой фронт

После аннотирования начинаем применять к фрагментам правила преобразования, цель которых – уменьшить различия. К первому фрагменту

(рис. 6) применимо правило альтернативной записи (рис. 7). Правило устраняет ошибку, которая могла возникнуть при восприятии мелодии на слух: при исполнении ноты *staccato* она звучит короче и может быть записана как нота меньшей длительности и пауза.

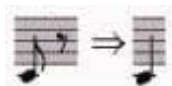


Рис. 7. Правило альтернативной записи

В результате пятикратного применения правила первый фрагмент будет переписан (рис. 8).



Рис. 8. Первый фрагмент после пятикратного применения правила альтернативной записи

Далее к паре фрагментов применяются правила сокращения. Они уменьшают длину волнового фронта, удаляя в нем те одинаковые ноты и паузы, которые граничат с основой (рис. 9).



Рис. 9. Фрагменты после применения правил сокращения

Теперь среди известных системе правил не осталось таких, которые были бы применимы к полученным фрагментам. Процесс преобразования на этом завершается. Наилучшим вариантом завершения преобразований является такой, при котором полученные фрагменты полностью совпадают (мера их различий становится равной нулю). Однако на практике не всегда удается достичь нулевой меры. В рассмотренном примере совпадающие фрагменты не были получены, но после преобразования мера их различий стала заметно меньше: длина волнового фронта у исходных фрагментов равна 22, у преобразованных – 5.

Оценка работы метода

Для оценки предложенного метода поиска была выполнена серия запросов. В базе системы, состоящей из 30500 фрагментов, находились 15 похожих, с точки зрения пользователя, фрагментов (рис. 10). Фрагменты *a* и *d* являются записью одной и той же мелодии от разных начальных нот. Несмотря на их формально разную запись, они имеют одинаковое внутреннее представление и для системы являются одной мелодией. Остальные фрагменты отличаются от первого высотой одной (*b*) или нескольких (*e, f*) нот, либо отличаются длительностью одной или нескольких нот (*c*), либо имеют большее количество различий.

Последовательно был выполнен поиск данных пятнадцати фрагментов. При поиске по запросу *a* были найдены 13 похожих мелодий (все из представленных на рис. 10, кроме *h* и *i*). Результаты поиска для запроса *a* представлены в таблице 1. Для запроса *d* были получены те же результаты (и те же значения функции похожести мелодий).

Таблица 1

Результаты работы системы

Запрос	Результат	Значение функции похожести
а	а	1
	д	1
	б	0.97
	с	0.9
	е	0.9
	ф	0.87
	ж	0.85
	г	0.83
	о	0.82
	к	0.81
	м	0.8
	н	0.73

При всех 15 запросах не было найдено «лишних» мелодий (непохожих на запрос с точки зрения пользователя), но не все похожие фрагменты удавалось найти: в среднем полнота результатов поиска составила около 75%. Некоторые из мелодий не находятся, так как в системе нет подходящих волновых правил, которые позволили бы привести их друг к другу. Однако добавление новых правил может, вообще говоря, снизить точность поиска.

The image displays a musical score consisting of 15 staves, labeled 'a' through 'o'. Each staff contains a melodic line in 4/4 time, primarily using eighth and quarter notes. The key signature is one sharp (F#). The fragments are highly similar in their rhythmic and melodic patterns, illustrating a set of related musical motifs. Some staves, such as 'd', 'h', and 'm', include chromatic alterations (sharps and flats) to create variations. Additionally, there are two short, isolated melodic phrases, one starting at measure 7 on staff 'd' and another on staff 'h', which appear to be related to the main motifs.

Рис. 10. Набор похожих фрагментов, использованный для оценки метода

Заключение

В работе предложен метод поиска нотных записей, хранящихся в формате MusicXML. В рамках предложенного метода решены следующие задачи: выбрано внутреннее представление для хранения нотных записей, позволяющее проводить их сравнение в соответствии со спецификой предметной области, составлены правила преобразования, применяющиеся при оценке степени совпадения нотных фрагментов, и предложен способ индексирования библиотеки нотных файлов для эффективного поиска. Эксперименты с системой позволили подобрать эвристические коэффициенты для оценки сходства запроса и библиотечного фрагмента. Метод позволяет находить ноты музыкальных произведений по заданному фрагменту и оценивать соответствие результатов поиска введенному запросу.

В перспективе метод может быть использован для решения смежных задач, например, для поиска оригинального нотного текста по тексту, который был автоматически построен программой по аудиозаписи, или для поиска заимствований в нотных записях музыкальных произведений.

Список литературы

1. Bundy A., Basin D., Hutter D., Ireland A. Rippling: Meta-level Guidance for Mathematical Reasoning. Cambridge University Press, 2005.
2. Rippling: Meta-level Guidance for Mathematical Reasoning.
URL: <http://www.rippling.org>
3. Typke R. Music Retrieval Based on Melodic Similarity. Ph.D. Thesis, University of Utrecht, 2007
URL: <https://www.cs.uu.nl/groups/MG/multimedia/publications/art/typke-thesis.pdf>
4. Typke R., Giannopoulos P., Veltkamp R.C., Wiering F., van Oostrum R. Using Transportation Distances for Measuring Melodic Similarity // Proceedings of the Fourth International Conference on Music Information Retrieval. 2003. pp. 107–114.
5. Viro V., Peachnote: Music Score Search and Analysis Platform/// Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011. pp. 359–362.
6. Xing Wei, Bruce Croft W. Modeling Term Associations for Ad-hoc Retrieval Performance within Language Modeling Framework // Proceedings of the 29-th European conference on IR research, Springer-Verlag Berlin, Heidelberg, 2007, pp. 52–63.
7. Yang X., Chen Q., Wang X. A Novel Approach Based on Fault Tolerance and Recursive Segmentation to Query by Humming // Advances in Computer Science and Information Technology: AST/UCMA/ISA/ACN. Conferences Joint Proceedings, Springer-Verlag Berlin, Heidelberg, 2010. pp. 544–557
8. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание: Пер. с англ. – М.: Издательский дом "Вильямс", 2005. 1296 с.
9. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск.: Пер. с англ. – М.: ООО "И.Д. Вильямс", 2011. – 528 с.
10. Зарипов Р.Х. Кибернетика и музыка. М: Изд-во "Знание", 1963.
11. Дубовский И., Евсеев С., Способин И., Соколов В. Учебник гармонии. М: Изд-во "Музыка", 1965.
12. Вахромеев В.А. Элементарная теория музыки. М.: МУЗГИЗ, 1961.
13. MusicXML format. URL: <http://www.makemusic.com/musicxml>