

# Применение алгоритма Витерби к восстановлению стёртого фрагмента музыкального произведения

Ботхолов А.Ж.

Приведены методы восстановления стёртого фрагмента музыкальной композиции, основанные на алгоритме Витерби, а именно метод, основанный только на алгоритме Витерби, метод, использующий кроме алгоритма Витерби, также смещённую высоту и смещённую длительность, метод, основанный на алгоритме Витерби и принципах музыкальной гармонии. Выяснен метод с наилучшим результатом (алгоритм Витерби, совмещённый с методом музыкальной гармонии), введён способ оценки схожести фрагментов, приведены примеры мелодий, такты которых были лучше всего восстановлены. Все методы вписаны в программу на языке Java, которая решает заданную задачу, кроме того используется программа GuitarPro, помогающая перевести всю необходимую информацию о мелодии в текстовый формат.

**Ключевые слова:** алгоритм Витерби, смещённая высота, смещённая длительность, гармонизация аккордами.

## Введение

За последнее время было проведено много исследований, связанных с музыкальными произведениями — поиск закономерностей в мелодиях, генерация последовательностей нот на основе выбранного стиля, использование математических моделей для создания музыкальных отрывков и другие. Всегда казалось, что последовательность звучащих нот в мелодии подчиняется закону. Возник вопрос: насколько хорошо пригоден алгоритм Витерби к восстановлению стёртого фрагмента музыкальной композиции?

В данной работе мы имеем дело с файлами, содержащими нотные записи музыкальных произведений (мелодии одноголосные). Существуют специальные программы (редакторы нот), с помощью которых со-

здаются такие файлы в формате MusicXML. Этот формат достаточно популярен среди большого числа программ-редакторов, которые дают возможность читать и вносить изменения в нотные записи. Для проверки пригодности алгоритма Витерби к восстановлению стертого фрагмента будет использоваться запись музыкального произведения в виде MusicXML файла, обладающий определенной структурой.

## Структура MusicXML файла

Используемые методы восстановления фрагмента музыкальной композиции работают с файлами в формате MusicXML. Данный вид файлов состоит из набора тегов. Перечислим некоторые из них: `<note>`, `<octave>`, `<type>`, `<alter>`, `<string>`, `<fret>`, `<measurenumber =”2”>`, отвечающие за ноту, октаву, длительность, присутствие или отсутствие диеза, название струны, лад, номер такта соответственно. Всего существует 55 нот плюс пауза, которую считаем за отдельную ноту.

## Методы восстановления фрагмента

Музыкальная композиция представляет собой последовательность тактов. Такт есть множество нот и пауз, находящееся между 2-мя вертикальными линиями (тактовыми чертами) на нотном стане. Нотный стан есть некая разметка, где записываются ноты.

Допустим в музыкальной композиции стерли некоторое множество идущих друг за другом тактов. Наша задача восстановить эти такты с наименьшей погрешностью, то есть узнать количество стертых нот и значения этих нот. В работе были испробованы два способа построения недостающих нот, на каждый из которых накладывались различные методы.

Первый способ состоит в том, что каждый стертый такт мы рассматриваем как последовательность 16-ти нот длительностью, равной размеру такта, деленного на 16. После получения в каждом стертом такте 16-ти нот есть возможность объединять одинаковые ноты, идущие друг за другом в одну ноту длительностью, равной сумме длительностей суммированных нот. Способ же реализуется одним из двух вариантов: первый учитывает длительности нот в оставшихся не стертых тактах, то есть если в оставшихся тактах есть нота C длительностью не больше  $1/2$ , то

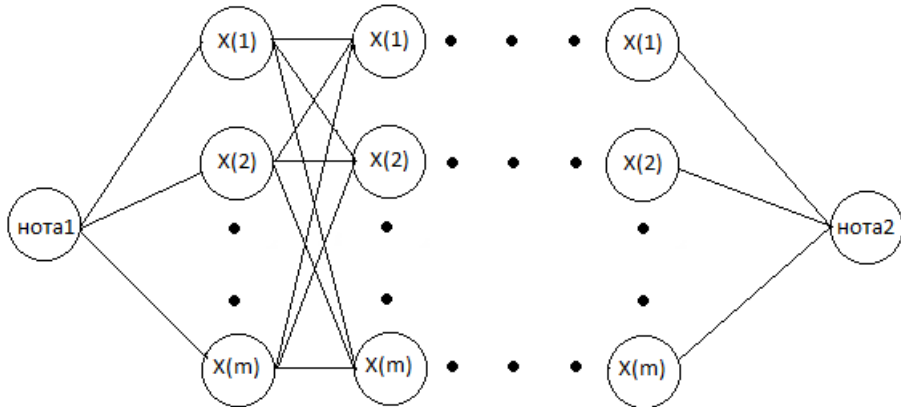
и в восстанавливаемых тактах длительность ноты  $C$  не должна превышать  $1/2$ ; второй вариант не учитывает длительности нот.

Второй способ основан на том, что количество нот в каждом такте стертого фрагмента равно тому количеству нот, который наиболее часто встречается в не стертых тактах. После выяснения данного количества определяем наиболее вероятные длительности восстановленных нот. (Как казалось, второй способ лучше решает задачу, нежели первый, поэтому в дальнейшем используем только его.)

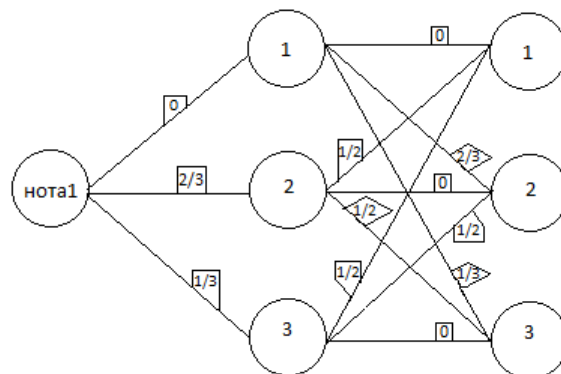
Как было сказано, на эти способы накладывались различные методы решения задачи, причем был испробован некий «симбиоз» методов, то есть их одновременное использование. Рассмотрим эти методы.

## Алгоритм Витерби

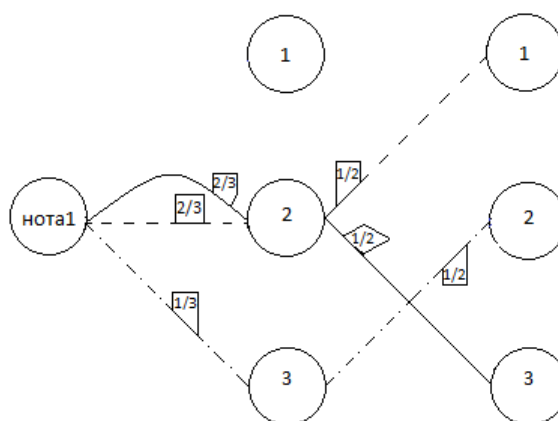
Допустим в мелодии стерли фрагмент, состоящий из нескольких тактов (от 1 до 7). Чтобы восстановить фрагмент, исследуем оставшиеся в мелодии такты с нотами. Предполагаем, что мы оценили, сколько находилось нот во всех тактах фрагмента. Пусть это количество равно  $N$ , а ноты, которые есть в оставшихся фрагментах —  $x(1), x(2), \dots, x(m)$ ; то есть разных нот в оставшихся тактах мелодии равно  $m$ . Нам известна нота перед фрагментом и после него, обозначим их нота1 и нота2 соответственно. После нота1 может стоять одна из  $m$  нот причем с определенной вероятностью (программа вычисляет для каждой пары нот, идущих друг за другом, сколько раз она встречается в не стертых тактах, а затем находит вероятности появления этих пар), после каждой из этих  $m$  нот может идти одна из этих же  $m$  нот и т. д. Фрагмент же заканчивается нотой, которая переходит в нота2. Таким образом существует большое число последовательностей нот, начинающихся в нота1 и заканчивающихся в нота2. Представить эту схему можно с помощью следующего рисунка



Ясно, что количество столбцов с нотами  $x(1), x(2), \dots, x(m)$  на рисунке равно числу  $N$ . Наша задача состоит в нахождении наиболее вероятной последовательности (пути). Для восстановления используется алгоритм Витерби. Алгоритм состоит том, что на каждом шаге восстановления фрагмента (шаг — переход с одного столбца на другой) для каждой ноты из столбца находим наиболее вероятный путь, выходящий из нота1 и заканчивающийся в данной ноте (следует отметить, что путей может быть несколько). На следующем шаге для вычисления для каждой ноты наиболее вероятного пути используем вычисленные пути на предыдущем шаге. Для понимания процесса разберем пример двух шагов алгоритма для трёх нот. На рисунке на ребрах стоят вероятности появления ноты на правом конце ребра после ноты на левом конце.

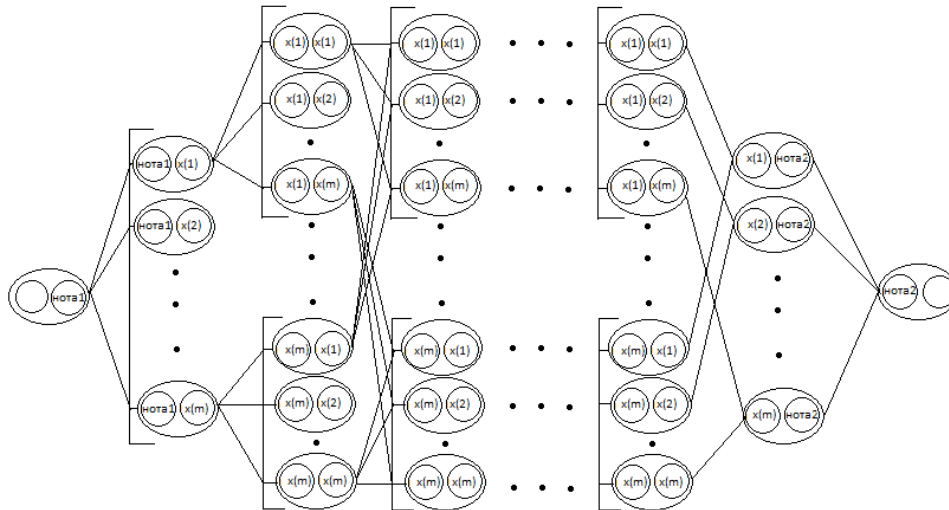


На первом шаге алгоритма для каждой ноты из первого столбца наиболее вероятным путем является ребро, соединяющее нота1 и рассматриваемую ноту, на следующем же шаге для нот второго столбца имеем следующее:



Для ноты 1 обозначен путь с наибольшей вероятностью пунктиром, для ноты 2 — штрихпунктиром, для ноты 3 — обычными линиями. Таким образом, мы рассмотрели вариант метода, который учитывает вероятности появления одной ноты после другой. Этот вариант назовем 1-грамм.

По аналогии определяем вариант 2-грамм, который учитывает вероятности появления одной ноты после двух других. Схематично этот метод можно изобразить так

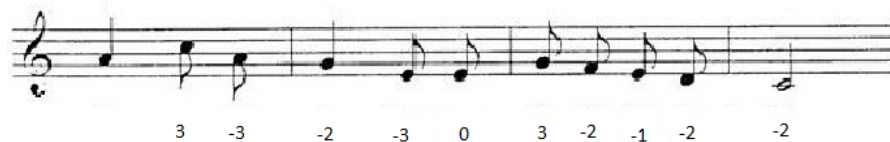


В работе рассмотрены 4 типа реализации алгоритма: 1-грамм, 2-грамм, 3-грамм, 4-грамм.

## Метод, основанный на использовании смещенной высоты и смещенной длительности ноты с алгоритмом Витерби

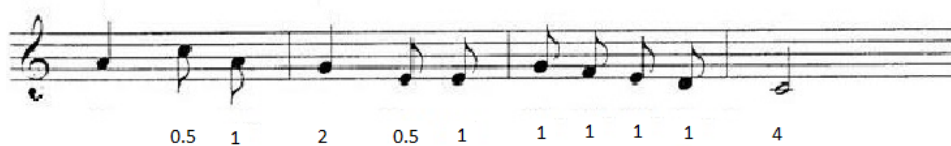
Смещенная высота ноты — это число, которое вычисляется как разность полутонов данной ноты и предыдущей ноты. Для всех нот мелодии, за исключением первой ноты, вводится понятие смещенной высоты. Подход для описания нот, где вместо высот указывается смещенная высота, дает следующее преимущество — в случае неправильного восстановления в мелодии какой-то отдельной ноты ошибка отразится только на двух соседних смещенных высотах, но не повлияет на оставшуюся часть мелодии.

Покажем на примере отрывка песни "Елочка" (рис.1), как вычисляется смещенная высота. Под каждой нотой кроме первой указаны смещенные высоты. Положительные числа стоят, если предыдущая нота ниже данной, а отрицательные — в противном случае.



**Рис.1 Смещенные высоты нот**

Смещенная длительность ноты — это число, которое вычисляется как отношение длительности данной ноты к длительности предыдущей ноты или паузы. Для первой ноты фрагмента смещенная длительность не вычисляется (по аналогии со смещенной высотой). Как определяется смещенная длительность для фрагмента песни "Елочка" показано на рис. 2.



**Рис.2 Смещенные длительности нот**

Здесь же реализация алгоритма Витерби происходит так же как и в предыдущем пункте, только вместо нот используются смещенные высоты и смещенные длительности нот.

## **Алгоритм Витерби плюс метод музыкальной гармонии**

В теории музыки различают два главных способа изложения мелодии: полифонический — мелодия выполняется несколькими голосами, и каждый отдельный голос может выполнять свою мелодию независимо от других голосов, гомофонный склад — мелодия выполняется одним из голосов, а все остальные голоса выполняют роль аккомпанемента.

Так как мы работаем с одnogолосными мелодиями, то у всех рассматриваемых мелодий присутствует гомофонный склад. Наша задача подобрать аккомпанемент к основной мелодии или, другими словами, гармонизировать мелодию аккордами.

Из теории музыкальной гармонии известно, что почти любая мелодия может быть гармонизирована аккордами. Гармонизация мелодии начинается с вычисления тональности мелодии. Понятие тональности состоит из тоники — главной ноты мелодии и гармонического закона, с помощью которого по данной ноте достраивается шесть основных нот. Остальные же ноты могут встречаться в мелодии как дополнительные. Дальше для каждого такта мелодии подбираем аккорды, которые бы гармонично звучали вместе с мелодией. С помощью алгоритма Витерби вычисляем наиболее вероятные аккорды стертого фрагмента и, исходя из полученных аккордов, для восстановления тактов рассматриваем только те ноты, которые соответствуют определенному аккорду. То есть в процессе построения фрагмента мы используем гораздо меньшее число нот.

## Схема реализации методов

Предложенные методы реализованы в программе на языке Java. Выбранные мелодии в формате midi переводятся с помощью программы GuitarPro в формат MusicXML, с которым, собственно, и работает программа.

Программа считывает с файла всю необходимую информацию про ноты: значение, длительность, октава и т.д. В зависимости от метода, находит нужные вероятности нотных последовательностей, получает некоторое множество чисел, которые заменяет на теги для нового файла в формате MusicXML. С помощью GuitarPro переводим полученный файл в файл формата Midi.

Процесс можно схематически изобразить следующим образом:

$$\begin{aligned} \text{Midi(1)} &\longrightarrow \text{GuitarPro} \longrightarrow \text{MusicXML(1)} \longrightarrow \text{Programm} \longrightarrow \\ &\longrightarrow \text{MusicXML(2)} \longrightarrow \text{GuitarPro} \longrightarrow \text{Midi(2)} \end{aligned}$$

Программа в отдельный файл записывает вероятности появления ноты после одной ноты, двух, трех, четырех нот в зависимости от варианта алгоритма и работает с этим файлом для получения результата.



## Результаты работы методов

Так как в музыкальной области нет формальной оценки сходства мелодий (в основном оно определяется на слух), для оценки точности восстановления фрагмента была введена формула коэффициента похожести

$$k = \frac{m \cdot \min(n_1, n_2)}{n_1 \cdot \max(n_1, n_2)},$$

$k$  — коэффициент похожести фрагментов,

$m$  — число совпавших нот,

$n_1$  — число нот фрагмента оригинальной музыкальной композиции,

$n_2$  — число нот фрагмента, восстановленного программой.

Результатом вычисления коэффициента похожести фрагментов является значение из отрезка  $[0, 1]$ , где 0 соответствует полному несовпадению фрагментов, а 1 — их полному совпадению. Как видно коэффициент зависит не только от количества «угаданных» нот, но и от того, из скольких нот, как предполагалось, состоит восстанавливаемый фрагмент. Таким образом, если программа указала достаточно много правильных нот, но полученное количество нот, из которых состоит фрагмент, оказалось далеким от истинного, то коэффициент может оказаться достаточно малым по сравнению с коэффициентом в случае правильно указанного количества нот. Этот факт нужно учитывать и понимать, что коэффициент может получиться «плохим» несмотря на то, что количество угаданных нот велико.

В таблице 1 представлены 5 мелодий, в которых лучше всего восстановились ноты (были удалены 2 такта). Мелодии из таблицы можно оценить по полученным коэффициентам при использовании различных вариаций Алгоритма Витерби, метода смещенной высоты и смещенной длительности нот с алгоритмом Витерби, совмещения Алгоритма Витерби и метода музыкальной гармонии.

Таблица 1

Композиции	Алгоритм Витерби				Смещенн.+ Витерби	Вит.+муз. гармония
	1-грамм	2-грамм	3-грамм	4-грамм		
1	0,57248	0,62355	0,62124	0,75913	0,51012	0,85624
2	0,23779	0,37933	0,39408	0,41068	0,36560	0,57890
3	0,23933	0,20453	0,37692	0,37892	0,34494	0,31222
4	0,12672	0,48744	0,24265	0,35827	0,21447	0,39062
5	0,11681	0,17479	0,31733	0,29333	0,25462	0,29444

Как можно заметить, наибольшие коэффициенты похожести имеет способ, совмещающий Алгоритм Витерби и метод музыкальной гармонии. В таблице использованы следующие мелодии: мелодия «Led Zeppelin — Stairway to heaven», мелодия песни «AC/DC — Back In Black», мелодия песни «Metallica — Nothing else matters», мелодия «Бах — Токката», мелодия «Моцарт — Свадьба Фигаро».

Наименьшие коэффициенты имеет метод алгоритм Витерби 1-Грамм. Также можно сделать вывод, что в большинстве случаев с увеличением количества стертых тактов фрагмента уменьшается точность восстановления нот.

Также хотелось бы отметить, что в данной работе было проведено исследование пригодности именно алгоритма Витерби для восстановления стертого фрагмента. Но если рассмотреть другие методы для восстановления, то как минимум есть один способ, который при дополнительно наложенных условиях может полностью восстановить стертый фрагмент. А именно метод, где мы повторяем мелодию несколько раз и стираем несколько тактов в пределах одной мелодии. Затем если перед стертым фрагментом есть не стертый кусок, то находим данный кусок в мелодии (он обязательно найдется, так как мы повторили мелодию несколько раз) и смотрим какая нота стоит после него и ставим эту ноту в стертый фрагмент, потом находим следующую ноту фрагмента аналогичным способом. Если же перед стертым фрагментом нет нот, то смотрим на не стертый кусок после данного фрагмента и также находим это кусок в мелодии, затем восстанавливаем ноты с конца стертого фрагмента.

## Список литературы

- [1] Рябинович Е. В. Сжатое представление музыкальных сигналов для поиска музыкальных произведений, 2002.
- [2] Славщик А.А. История алгоритмической музыкальной композиции.
- [3] MusicXML format. URL: <http://www.makemusic.com/musicxml>
- [4] Тупке R. Music Retrieval Based on Melodic Similarity. Ph.D. Thesis, University of Utrecht, 2007.
- [5] Rippling: Meta-level Guidance for Mathematical Reasoning.
- [6] Зарипов Р.Х. Кибернетика и музыка. М: Изд-во «Знание», 1963.
- [7] Вахромеев В.А. Элементарная теория музыки. М.: МУЗГИЗ, 1961.

## **Applying of the Viterbi algorithm to the recovering of an erased fragment of a musical composition**

**Botkholov A.J.**

The problem of the recovering of an erased fragment of a musical composition based on the Viterbi algorithm, namely, the method based only on the Viterbi algorithm, the method using the Viterbi algorithm and also shifted height and shifted duration, the method based on the Viterbi algorithm and the principles of musical harmony are obtained. The method with the best result was found (Viterbi algorithm combined with the method of musical harmony), the method for estimating of the similarity of fragments was introduced, examples of melodies, the tacts of which were best recovered, are given. All methods are written into the program in the Java language, which solves the given task, in addition, the program GuitarPro is used, which helps to translate all the necessary information about the melody into a text format.

Keywords: the Viterbi algorithm, shifted height, shifted duration, chord harmonization.

