



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

*«Методы генерации фрагмента музыкального
произведения»*

Студент ИУ7-52Б
(Группа)

(Подпись, дата)

А. А. Аверьянов
(И. О. Фамилия)

Студент ИУ7-52Б
(Группа)

(Подпись, дата)

П. В. Маслюков
(И. О. Фамилия)

Студент ИУ7-52Б
(Группа)

(Подпись, дата)

С. С. Котов
(И. О. Фамилия)

Студент ИУ7-56Б
(Группа)

(Подпись, дата)

М. Ю. Вольняга
(И. О. Фамилия)

Руководитель НИР

(Подпись, дата)

А. С. Кострицкий
(И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ	3
ВВЕДЕНИЕ	5
1 Анализ предметной области	6
2 Описание существующих решений	7
2.1 Математический подход	7
2.1.1 Алгоритм Бьорклунда	7
2.1.2 Алгоритм, основанный на клеточных автоматах	10
2.2 Статистический подход	12
2.2.1 Метод случайного блуждания	12
2.2.2 Цепи Маркова	14
2.3 Грамматический подход	17
2.3.1 Порождающая грамматика	18
2.3.2 L-системы	19
2.4 Трансляционный подход	21
2.4.1 Алгоритм анализа изображения	22
2.4.2 Алгоритм генерации музыкального фрагмента	25
3 Классификация существующих решений	30
ЗАКЛЮЧЕНИЕ	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

ОПРЕДЕЛЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие термины с соответствующими определениями.

Музыкальный тон — это устойчивый периодический звук. Музыкальный тон характеризуется его длительностью, высотой, интенсивностью (или громкостью) и тембром (или качеством) [1].

Темп — это скорость движения в музыке, мера времени в музыке [1].

HSB (от англ. Hue, Saturation, Brightness — тон, насыщенность, яркость) — цветовая модель, в которой координатами цвета являются: цветовой тон (например, красный, зеленый или сине-голубой), насыщенность (варьируется в пределах 0—100), яркость (варьируется в пределах 0—100) [2].

RGB (от англ. Red, Green, Blue — красный, зеленый, синий) — цветовая модель, описывающая способ кодирования цвета для цветовоспроизведения с помощью трех цветов, которые принято называть основными [2].

Кластеризация — это разбиение элементов некоторого множества на группы по принципу схожести. Эти группы принято называть кластерами [3].

Объект — элементарная группа данных, с которой оперируют алгоритмы кластеризации. Каждый объект описывается вектором характеристик [3]: $X = \{x_1, x_2, \dots, x_m\}$, где m — количество характеристик объекта. Компоненты x_i , где $(i = 1, 2, \dots, m)$ являются отдельными характеристиками объекта, как правило, представленными количественными признаками (например, координаты для точки, цветовые компоненты для цвета и т. д.) [3].

Искусственный интеллект (ИИ) — это технология, которая имитирует человеческое поведение, чтобы выполнять задачи и постепенно обучаться, используя собранную информацию [4].

Компьютерная музыка — музыка, созданная с использованием электромузыкальных инструментов и электронных технологий последних десятилетий XX века [5].

Музыкальная нота — это моментальный снимок периодической звуковой волны определенной частоты [6].

Аккорд — это несколько различных по высоте звуков (более двух) воспроизводимых одновременно [7].

ВВЕДЕНИЕ

Музыка — это удивительное искусство, которое оказывает значительное влияние на наши эмоции и настроение. Она имеет способность переносить нас в другие миры и вызывать самые разнообразные чувства. Изобретение фонографа, радио, телевидения и интернета сделало музыку всеобщей и непреходящей. Звуки и мелодии распространяются между континентами взаимодействуя и обогащая друг друга. Однако, за этой привлекательной формой может скрываться глубокий и сложный процесс написания музыки.

Задача генерации музыкального фрагмента стала актуальной, начиная со второй половины 20 века и является такой и по сей день. Крупные корпорации, такие как Google и Sony разработали нейросети MusicLM и FlowMachine, с помощью которых пользователи могут создавать музыкальные композиции.

Цель данной работы — провести анализ методов генерации музыкального фрагмента.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области;
- провести обзор существующих методов и алгоритмов генерации музыкального фрагмента;
- сформулировать критерии сравнения методов генерации музыкального фрагмента;
- классифицировать описанные методы.

1 Анализ предметной области

В настоящее время выделяют 4 основных подхода к определению тональности текста [8].

Математический подход — создание музыкального фрагмента на основе математических алгоритмов [9].

Статистический подход — создания новых музыкальных фрагментов на основе анализа статистических свойств музыкальных данных таких как жанр, стиль, стиль композитора [10].

Грамматический подход — создание музыкального фрагмента на основе грамматических правил и шаблонов для создания музыкальных структур, таких как мелодии, аккорды или ритмы [11].

Трансляционный подход — это способ генерации музыкальных произведений на основе немusical данных, таких как графические образы или текст. Данный процесс может быть случайным или основываться на определенных правилах мелодической структуры, с использованием нейросетевых технологий для преобразования и распознавания исходных данных [12].

2 Описание существующих решений

2.1 Математический подход

Задача генерации музыкальных фрагментов стала актуальной, начиная со второй половины 20 века. Под музыкальным фрагментом подразумевают последовательность нот и молчания, где нотой является музыкальный звук, а молчанием отсутствие звуков. Музыкальность или гармоничность фрагмента определяется субъективно с помощью социального опроса. Одними из популярных математических методов, использующих математические алгоритмы при генерации музыкальных последовательностей, являются метод Бьорклунда и метод, использующий клеточные автоматы.

2.1.1 Алгоритм Бьорклунда

Музыкальный фрагмент из одного музыкального звука представляет из себя циклическую последовательность нот и молчания, расположенных «равномерно». Для того, чтобы сформировать последовательность используется алгоритм Бьорклунда. Введем два обозначения «X» и «●» где первое обозначает начало ноты, а второе представляет молчание. Тогда музыкальная последовательность длиной N временных интервалов может быть представлена в виде последовательности из N символов. Обозначим количество X за K, а длину последовательности за N. Примерами таких последовательностей являются кубинские ритмы Тресильо ($N = 8$, $K = 3$) и Чинквильо ($N = 8$, $K = 5$) [13], пояснение изображено на рисунке 2.1.

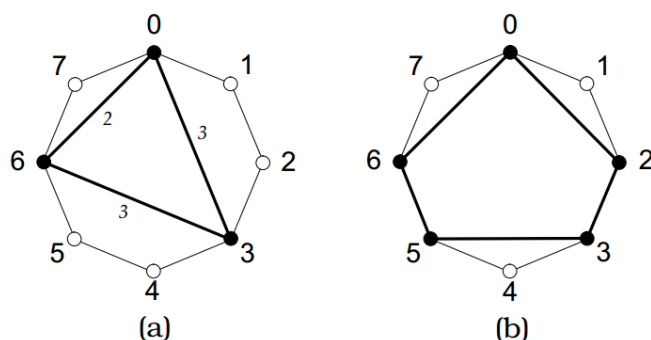


Рисунок 2.1 – Тресильо (a) и Чинквильо (b)

Описание алгоритма генерации музыкального фрагмента с помощью алгоритма Бьорклунда:

- 1) определить три группы: группу А, состоящую из символов «Х», группу В, состоящую из символов «●» и пустую группу С;
- 2) очистить содержимое группы С;
- 3) получить новый элемент, сопоставив элементу группы А в соответствие элемент группы В, новый элемент поместить в группу С;
- 4) повторить шаг 3, если группы А и В не пустые;
- 5) содержимое группы С записать в группу А, а оставшиеся элементы из групп А или В, записать в группу В;
- 6) перейти на шаг 2, если группа В не пустая;
- 7) соединить все элементы из группы А в один. Сформировать музыкальный фрагмент на основе последовательности символов «Х» и «●» в получившемся элементе, размещая вместо «Х» ноту, а вместо «●» молчание.

В случае, если N делится на K без остатка, решение тривиально. Пример для $N = 6$, $K = 2$ приведен в таблице 2.1.

Таблица 2.1 – Пример для $N = 6$, $K = 2$

Номер шага выполнения	Номер шага в алгоритме	A	B	C
1	1	X, X	•, •, •, •	
2	2	X, X	•, •, •, •	
3	3	X	•, •, •	X•
4	4	X	•, •, •	X•
5	3		•, •	X•, X•
6	4		•, •	X•, X•
7	5	X•, X•	•, •	X•, X•
8	6	X•, X•	•, •	X•, X•
9	2	X•, X•	•, •	
10	3	X•	•	X••
11	4	X•	•	X••
12	3			X••, X••
13	4			X••, X••
14	5	X••, X••		X••, X••
15	6	X••, X••		X••, X••
16	7	X••X••		X••, X••

Пример для $N = 5$, $K = 2$ приведен в таблице 2.2.

Таблица 2.2 – Пример для $N = 6$, $K = 2$

Номер шага выполнения	Номер шага в алгоритме	A	B	C
1	1	X, X	•, •, •	
2	2	X, X	•, •, •	
3	3	X	•, •	X•
4	4	X	•, •	X•
5	3		•	X•, X•
6	4		•	X•, X•
7	5	X•, X•	•	X•, X•
8	6	X•, X•	•	X•, X•
9	2	X•, X•	•	
10	3	X•		X••
11	4	X•		X••
12	5	X••	X•	X••
13	6	X••	X•	X••
14	2	X••	X•	
15	3			X••X•
16	4			X••X•
17	5	X••X•		X••X•
18	6	X••X•		X••X•
19	7	X••X•		X••X•

2.1.2 Алгоритм, основанный на клеточных автоматах

Более сложные музыкальные фрагменты можно генерировать с помощью клеточных автоматов. Клеточные автоматы – это модели вычислений, которые состоят из сети ячеек (клеток) [14], каждая из которых может находиться в определенном состоянии, и эти состояния изменяются в соответствии с правилами, определенными для данного автомата. Пример одномерного клеточного автомата приведен на рисунке 2.2.

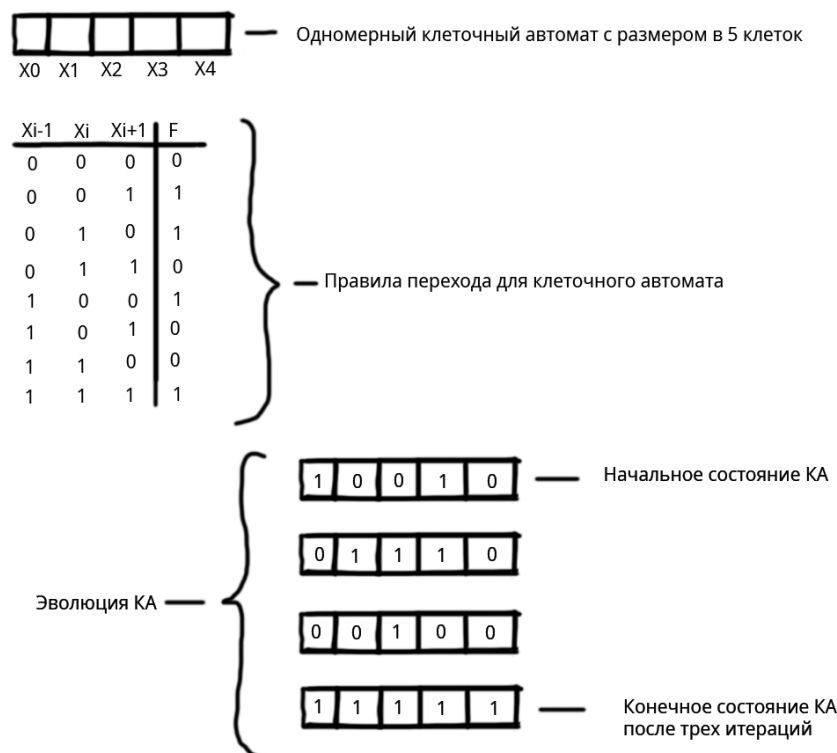


Рисунок 2.2 – Пример одномерного клеточного автомата

В контексте генерации музыки, каждая ячейка клеточного автомата может рассматриваться как нота или аккорд. Описание алгоритма генерации музыкального фрагмента с помощью клеточного автомата:

- 1) определить размер и мерность сетки (1D или 2D) клеточного автомата;
- 2) задать правила, которые определяют, какое состояние будет принимать каждая клетка в следующем поколении, исходя из ее текущего состояния и состояния соседних клеток;
- 3) каждой клетке сопоставить определенную ноту;
- 4) задать условие остановки вычислений;
- 5) выполнить вычисления в клеточном автомате, чтобы определить состояние каждой клетки в следующем поколении, при каждом переходе в следующее поколение записать в результат музыкальное событие на основе состояния клетки;
- 6) продолжить вычисления, пока ложно условие остановки.

2.2 Статистический подход

Музыкальное произведение может быть представлено последовательностью событий, в котором ноты являются музыкальными объектами с длительностью и временем начала [15]. Статистическая модель музыки приписывает каждому возможному музыкальному событию вероятность его появления. Наиболее распространенным типом статистических моделей, встречающихся для музыки, как для анализа, так и для синтеза, являются модели, которые присваивают вероятности событиям, обусловленным только более ранними событиями в последовательности [15]. Создание фрагмента из статистической модели означает выборку фрагмента, который имеет высокую вероятность в соответствии с моделью [15]. Существуют разные методы генерации музыкальных композиций, основывающиеся на статистическую модель. К таким методам относятся метод случайного блуждания и Марковский метод.

2.2.1 Метод случайного блуждания

Это способ создания музыки из модели, состоящий в выборке на каждом этапе случайного события из распределения событий на этом этапе [15]. После того, как событие обработано, оно добавляется к фрагменту, и процесс продолжается до тех пор, пока не будет превышена заданная длительность фрагмента [16]. Рассмотрим схематичное представление модели случайного блуждания, представленную на рисунке 2.3

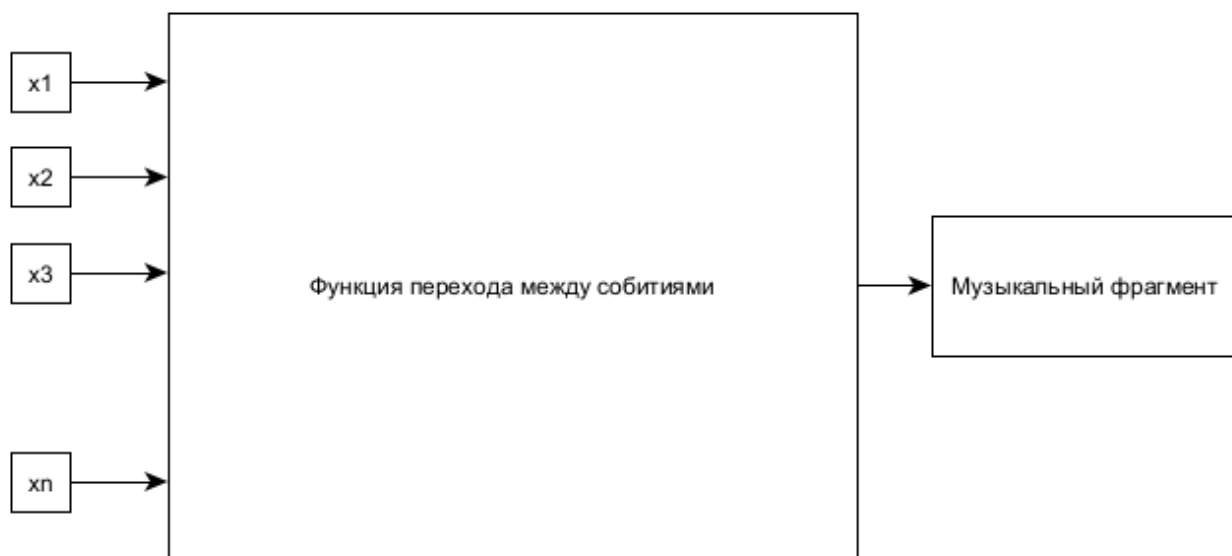


Рисунок 2.3 – Схематичное представление метода случайного блуждания

x_1, x_2, \dots, x_n – множество событий, после обработки путем перехода между событиями получаем музыкальный фрагмент.

Пусть точечная частица может совершать только один тип движений. В дискретные моменты времени t_0, t_1, \dots, t_n частица совершает скачок вдоль прямой так, то в момент времени $t_n + 1$ она оказывается в точке, отстоящей на единичное расстояние вверх или вниз от точки, где она находилась в момент времени. На рисунке 2.4 визуализирована работа метода случайного блуждания.

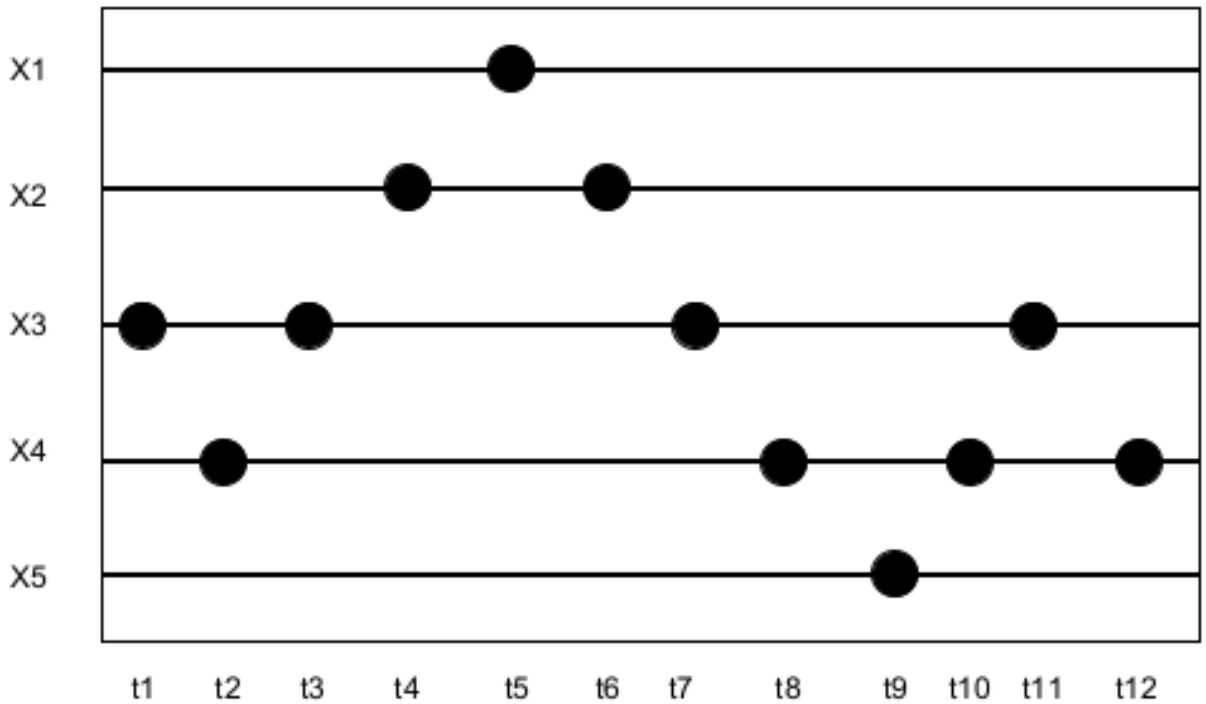


Рисунок 2.4 – Визуализация работы метода случайного блуждания

Без ограничения общности можно считать, что координата частицы в любой момент времени есть целое число. Введем на прямой некоторое начало отсчета и будем писать $\xi * j = m$, если в момент времени t_j частица находилась в точке m ; здесь $j = 1, 2, \dots, n$ и $m = x_1, x_2, \dots, x_n$.

Предположим, что блуждание имеет случайный характер: прыжок вверх частица совершает с вероятностью p , а прыжок вниз — с вероятностью q . При этом любые другие перемещения невозможны [16], так что $p + q = 1$. Примем также, что вероятности скачков не зависят от положения частицы и предыстории ее движения.

При анализе случайных блужданий частицы очень удобно пользоваться

понятием траектории ее движения за n шагов [16]. Она представляет собой набор точек $(j, \xi * j)$, $j = 1, 2, \dots, n$ на двумерной координатной плоскости, в котором первая координата — это номер члена последовательности, т. е. по сути момент времени $t = j$, а вторая — величина, значение которое равно координате частицы в момент времени $t = j$. Для наглядности удобно соединить точки траектории отрезками прямых, на графике получится непрерывная ломаная из n звеньев, координаты узлов которой $(j, \xi * j)$, $j = 0, 1, \dots, n$. Последовательность этих узлов и будет являться искомым фрагментом [16].

Метод случайного блуждания, хотя и применим к музыкальным импровизационным системам в реальном времени, требующим быстрой и немедленной реакции системы имеет недостаток в создании законченных кусков, потому что он не может гарантировать, что будут произведены куски с высокой общей вероятностью [15].

2.2.2 Цепи Маркова

Цепь Маркова — это модель, описывающая последовательность возможных событий [17]. Эта последовательность должна удовлетворять предположению Маркова — вероятность следующего состояния зависит от предыдущего состояния, а не от всех предыдущих состояний в последовательности [17]. Таким образом, композиция разбивается на последовательность состояний, и каждое состояние имеет набор возможных следующих состояний, которые выбираются на основе заданных вероятностей перехода. На рисунке 2.5 изображено схематичное представление модели Марковских цепей.

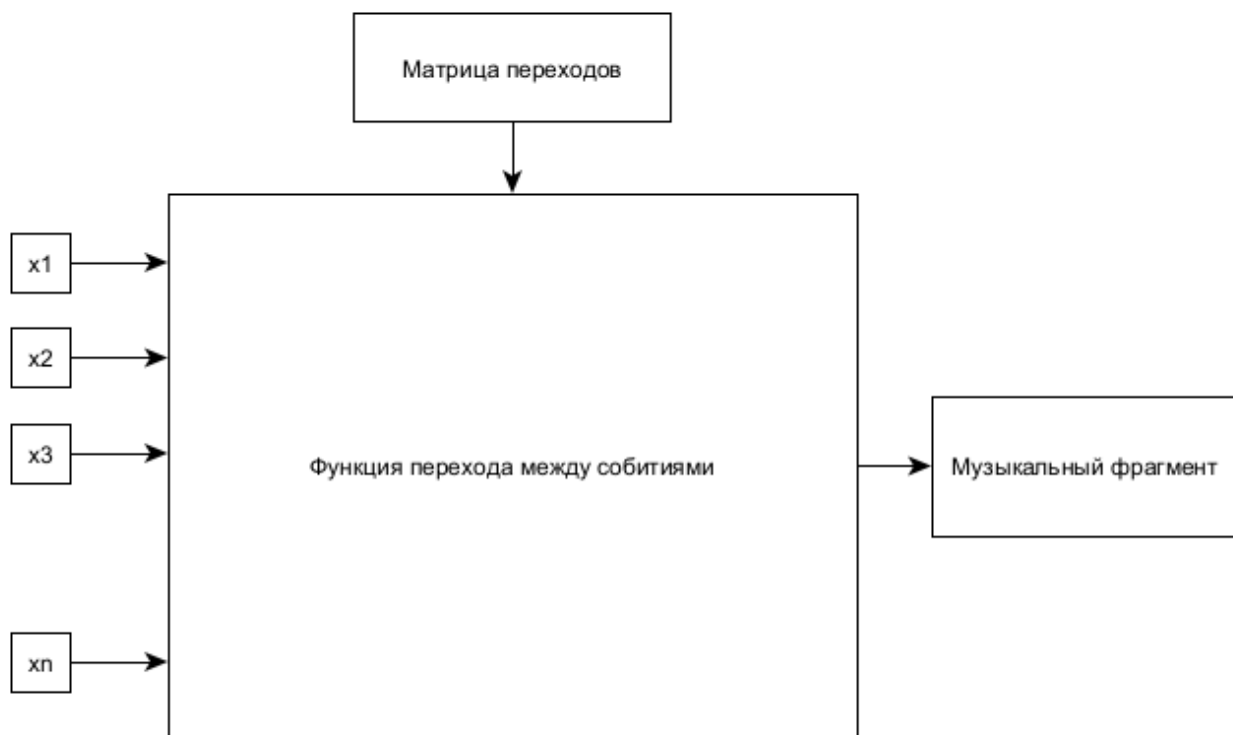


Рисунок 2.5 – Схематичное представление метода Марковских цепей

x_1, x_2, \dots, x_n — множество событий. События могут быть определены различными способами в зависимости от потребностей модели [17]. Например, состоянием может быть нота, аккорд, длительность, интонация и так далее. С каждым состоянием могут быть связаны вероятности перехода к следующим состояниям. Эти вероятности могут быть выведены из обучающего набора данных, на основе которых создается модель. Обучение модели может быть выполнено путем анализа существующих музыкальных композиций, при помощи которых определяются вероятности переходов.

Алгоритм Марковских цепей может быть записан в следующем виде [18]:

- подать на вход множество событий и размер необходимого фрагмента;
- рассчитать распределение вероятностей для событий;
- определить первое событие или сделать случайный выбор;
- сделать случайный выбор следующего события с учетом распределения вероятностей;
- повторять два предыдущих шага для сгенерированного события пока не достигнут нужный размер.

Разберем работу алгоритма на примере. В качестве входного множества событий возьмем корпус аккордов группы Beatles в композиции Yesterday [19]: [«F», «Em7», «A7», «Dm», «Dm7», «Bb», «C7», «F», «C», «Dm7»,...]

Делаем биграммы (последовательность из двух соседних элементов) из соседних аккордов [«F Em7», «Em7 A7», «A7 Dm», «Dm Dm7», «Dm7 Bb», «Bb C7», ...]

В качестве начального аккорда последовательности выбираем аккорд F. Необходимо рассчитать вероятности следующего аккорда. Есть 18 биграмм, которые начинаются с аккорда F. [«F Em7», «F C», «F F», «F Em7», «F C», «F A7sus4», «F A7sus4», ...]

Необходимо рассчитать частоту появления каждого уникального биграмма в последовательности. «F Em7»: 4, «F C»: 4, «F F»: 3, «F A7sus4»: 4, «F Fsus4»: 2, «F G7»: 1

Если нормализовать полученное количество появлений биграмм в последовательности, получим вероятности

«F Em7»: 0.222, «F C»: 0.222, «F F»: 0.167, «F A7sus4»: 0.222, «F Fsus4»: 0.111, «F G7»: 0.056

Данные вероятности можно интерпретировать в виде графа, представленного на рисунке 2.6:

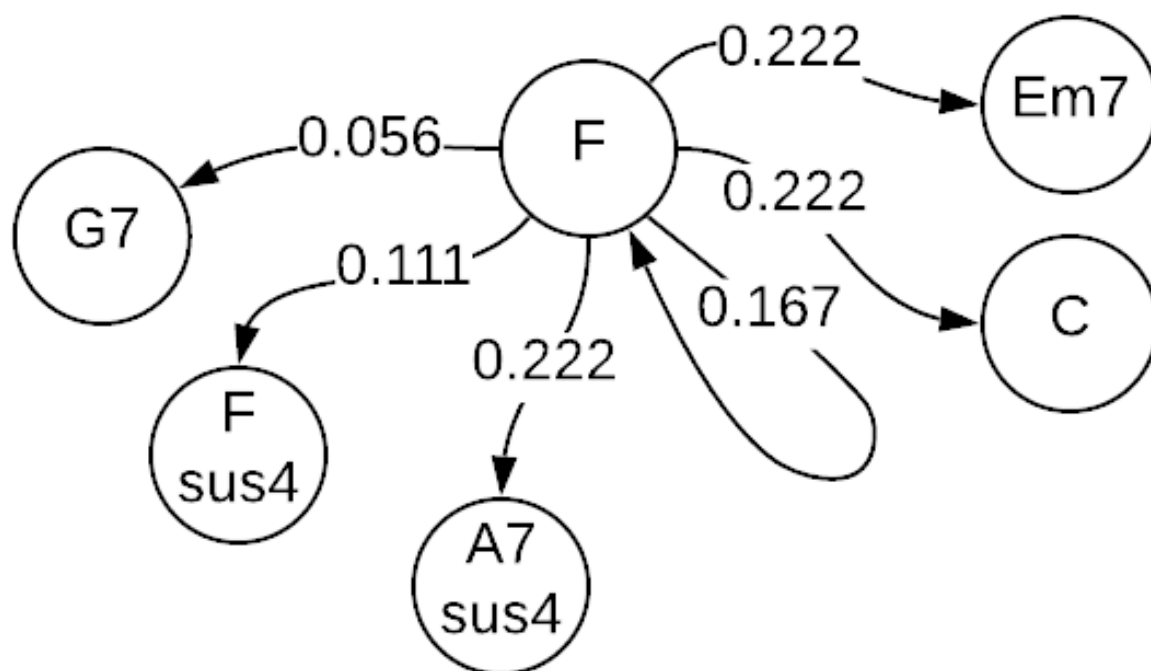


Рисунок 2.6 – Граф вероятностей перехода

Каждый узел этого графа, кроме начального узла F в центре, представляет возможные состояния, которых может достичь наша последовательность, в нашем случае это аккорды, которые могут следовать за F. Некоторые из аккордов имеют более высокую вероятность, чем другие, некоторые аккорды не могут следовать за F. Например, аккорд Am, потому что не было биграмма, которой объединяет этот аккорд с F. Далее нужно выполнить случайную выборку в соответствии с распределением вероятности. Допустим, наш результат этого случайного выбора был Em7. Теперь у нас новое состояние, и мы можем повторить весь процесс снова.

Таким образом, метод генерирует музыку, которая статистически «похожа» на исходные композиции, но в то же время имеет свою уникальность, так как варианты переходов могут быть случайными.

2.3 Грамматический подход

Грамматическая модель может определять, какие ноты, аккорды или мелодические фразы могут быть использованы в композиции, а также как они могут быть комбинированы и повторены. Она может также определять

структуру композиции, такую как введение, куплеты, припевы и заключение. Применение грамматической модели в музыке позволяет создавать композиции, которые соответствуют определенным стилям и жанрам. Грамматическая модель применяется для автоматической генерации музыки, так и как средство для анализа и понимания уже существующих музыкальных произведений. Существуют разные методы генерации музыкальных композиций, основывающиеся на грамматической модели. К таким методом относится порождающая грамматика [20].

2.3.1 Порождающая грамматика

Этот метод является формализмом генеративной лингвистики, связанный с изучением синтаксиса. В рамках подхода порождающей грамматики формулируется система правил, при помощи которых можно определить, какая комбинация слов оформляет грамматически правильное предложение. Термин впервые введен в научный оборот американским лингвистом Ноамом Хомским в конце 1950-х годов. Цель лингвистической теории по Хомскому заключается в том, чтобы объяснить факт поразительно быстрого усвоения родного языка ребенком на основе явно недостаточного внешнего стимула, то есть той информации, которая может быть извлечена из речи окружающих [21]. Данный метод построения грамматики называется методом динамически расширяющегося контекста. В контексте музыки, порождающая грамматика может определять правила для порождения последовательности нот или аккордов. Например, она может иметь правило, которое генерирует последовательность нот в рамках заданного тонального ряда или музыкальной шкалы. Она также может иметь правила для определения длительности нот или мелодических фраз.

Рассмотрим пример составления грамматических правил и генерации новой строки по этим правилам. Берется обычная строка: ABCDEFGIKFHLEFJ. И строится для нее грамматика, начав, например, с символа F [6]. Записывается правило, которое бы указывало, какую букву следует поставить, если встречен символ F. Нельзя создать такое правило, так как лишь по одной букве нельзя определить, что должно идти следом: после F может идти как G, так и H или J. Поэтому добавляется контекст к букве F, контекст — это символы, окружающие F. Взяв по одной букве перед F, получается EF и KF.

Контекстом для буквы F служат буквы E и K. Таким образом, расширяется контекст на один символ, поэтому данный метод построения грамматики называется методом динамически расширяющегося контекста. Это правила для буквы F, в зависимости от ее контекста выбирается какое-то одно правило. Процесс генерации новой строки выглядит следующим образом: дана начальная последовательность, например, ADEF. Буквы берутся с конца. F — нет правила с такой левой частью, расширяется контекст — EF, опять нет, расширяется — DEF, есть такое правило, ставится G, получается ADEFG. Начинается все сначала: берется буква G и т. д. столько раз, сколько нужно.

Порождающая грамматика используется для задачи генерации музыкальной партитуры в виде нот, однако требует определения четких формальных правил построения композиции, что является крайне трудоемким процессом.

2.3.2 L-системы

L-система или система Линденмайера — это параллельная система переписывания и вид формальной грамматики. L-системой (точнее, ее разновидностью, детерминированной контекстно независимой L-системой) называют набор, состоящий из алфавита, аксиомы, и множества правил. Алфавитом называется конечное множество, а его элементы — символами. Природа символов не важна, их единственная функция — отличаться друг от друга. Строкой над алфавитом является конечная последовательность символов алфавита. Аксиома — это некоторая строка над алфавитом. Каждое правило — это пара, состоящая из предшественника и последователя. Предшественник — это символ алфавита, а последователь — строка над алфавитом. Пример пары:

$$C \rightarrow EDEC + FEC + ED - EC,$$

стрелка отделяет предшественника от последователя. В списке правил символы-предшественники должны быть уникальными.

Таблица 2.3 – Правила L-системы

Алфавит	Аксиома	Правила
C,D,E,F,G,A,B	ED	$C \rightarrow EDEC + FEC + ED - EC,$ $D \rightarrow ED + EC - ED - GEDEC,$ $E \rightarrow \langle \rangle$ (последователь – пустая строка), $F \rightarrow B,$ $G \rightarrow A$

Как только L-система определена, она начинает развиваться в соответствии с ее правилами. Начальным состоянием L-системы является ее аксиома. При дальнейшем развитии эта строка, описывающая состояние, будет меняться. Развитие L-системы происходит циклически. В каждом цикле развития строка просматривается от начала к концу, символ за символом. Для каждого символа ищется правило, для которого этот символ служит предшественником. Если такого правила не нашлось, символ оставляется без изменений. Иными словами, для тех символов X , для которых нет явного правила, действует неявное: $X \rightarrow X$. Если же соответствующее правило найдено, символ-предшественник заменяется на строку-последователь из этого правила.

Для иллюстрации рассмотрим следующую L-систему (она называется **Algæ** — водоросль, поскольку ее развитие моделирует рост одного из видов водорослей):

Таблица 2.4 – L-система водоросль

Аксиома	Правила
A	$A \rightarrow B$ $B \rightarrow AB$

Состояния этой L-системы, соответствующие первым десяти циклам развития системы:

Таблица 2.5 – Состояния L-системы

Поколение	Состояние
0	A
1	B
2	AB
3	BAB
4	ABBA
5	BABBA
6	ABBAABBA
7	ABBAABBAABBA
8	ABBAABBAABBAABBA
9	ABBAABBAABBAABBAABBA ABBAABBAABBAABBAABBA

Длины строк, кодирующих состояние такой L-системы, образуют последовательность чисел Фибоначчи, то есть такую числовую последовательность, в которой каждое число равняется сумме двух предыдущих. Последовательностями Фибоначчи будут также количества символов A и B в этих строках. В последовательности строк имеется та же закономерность, что и в последовательности чисел Фибоначчи: каждая строка является «суммой» (конкатенацией) двух предыдущих [22].

2.4 Трансляционный подход

Создание музыки творческий процесс, его автоматизация сложна из-за важной роли композитора и труднопонимаемой эмоциональности в музыке [8]. Для автоматической генерации музыкальных композиций с учетом эмоционального состояния пользователя-композитора можно использовать трансляционные модели [23].

Генерация музыки на основе изображения может применяется в компьютерных играх, рекламе и фильмах для создания фоновой музыки. Автоматизация этого процесса позволит сократить расходы компаний, учитывая небольшие требования к фоновой музыке в этих областях [24].

Формальная постановка задачи:

- 1) пусть A — множество всех возможных изображений, исходные данные

для алгоритма;

- 2) пусть B — множество всех возможных музыкальных композиций, результаты работы алгоритма;
- 3) Тогда задача сводится к поиску приближенной функции, которая каждому элементу (изображению) $a \in A$ сопоставляет множество музыкальных композиций $f(a) \subseteq B$. Формула (2.1) формализует данное многозначное отображение.

$$f : A \rightarrow 2^B \quad (2.1)$$

Генерация фрагмента музыкального произведения по изображению, представляет собой преобразование визуальных данных в последовательности нот с определенным тоном и темпом [25].

Метод генерации фрагмента музыкального произведения по изображению состоит из двух составляющих алгоритмов:

- 1) алгоритм анализа изображения;
- 2) алгоритм генерации музыкального фрагмента.

2.4.1 Алгоритм анализа изображения

Алгоритм анализа изображения состоит из следующих шагов [25].

- 1) Преобразовать входное изображение из цветовой модели RGB в HSB. Цветовая модель HSB более удобна, так как содержит необходимые характеристики.
- 2) Определить тональность произведения.
- 3) Определить жанр произведения.
- 4) Определить схему соотнесения цвета и ноты.
- 5) Используя выбранную схему, найти последовательность первых нот, считанных с изображения.

Алгоритм анализа изображения позволяет извлекать музыкальные характеристики изображения, такие как тональность и жанр создаваемой музыкальной композиции [25]. Тональность и жанр — являются ключевыми

параметрами для трансляции изображения в музыку, поскольку они формируют эмоциональную составляющую произведения, и должны быть определены путем анализа цветовой гаммы изображения [24].

Определение тональности произведения

Тональность произведения определяется путем нахождения преимущественного цвета изображения и сопоставление его с выбранной схемой соответствия тона и цвета. В данном алгоритме была выбрана схема А. Н. Скрябина (таблица 2.6) [26; 27].

Таблица 2.6 – Соответствие цвета и тональности по схеме А. Н. Скрябина

Цвет	Тональность
Красный	C-dur
Оранжево-розовый	G-dur
Желтый, яркий	D-dur
Зеленый	A-dur
Синий, сапфировый	E-dur
Синий, мрачный	H-dur
Сине-яркий	Fis-dur
Фиолетовый	Des-dur
Пурпурно-фиолетовый	As-dur
Красный	F-dur

Для определения преимущественного цвета изображения, традиционно выбирают алгоритм кластеризации К-средних [23; 25]. Данный алгоритм имеет следующие особенности: высокое качество кластеризации, возможность эффективного распараллеливания, существует множество модификаций, число кластеров надо знать заранее [28; 29].

Алгоритм кластеризации К-средних — это неконтролируемый метод обучения. В этом методе данные разбиваются на кластеры на основе их сходства без использования предварительных обозначений [25].

Алгоритм состоит из следующих шагов [3]

- 1) Выбирается число k — количество кластеров.
- 2) Далее случайным образом из заданного изображения выбирается k точек. На первом шаге эти точки будут считаться «центрами» кластеров. Каждому кластеру соответствует один центр.

- 3) Все точки изображения распределяются по кластерам. Вычисляется расстояние от точки до каждого центра кластера (например используя Евклидово расстояние 2.3), и точку относят к тому кластеру, расстояние до центра которого будет наименьшим.
- 4) Когда все точки изображения распределены по кластерам, происходит пересчет центров кластеров. В качестве нового центра кластера берется среднее арифметическое всех точек, принадлежащих кластеру.

Пункты 3 и 4 повторяются до тех пор, пока не будет выполнено условие в соответствии с некоторым критерием остановки:

- кластерные центры стабилизировались, то есть все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации;
- число итераций равно максимальному числу итераций.

В алгоритме k -средних ставится цель минимизировать полную внутри-классовую дисперсию:

$$V = \sum_{i=1}^k \sum_{X_j \in C_i} (X_j - \mu_i)^2 \quad (2.2)$$

где X_j — векторы характеристик, k — количество кластеров, C_i — кластеры, μ_i — центры кластеров.

В качестве метрики для данной задачи традиционно берут евклидово расстояние [3; 30; 31].

Евклидово расстояние между точками x_i и y_i в n -мерном пространстве выражается следующей формулой:

$$\rho(x_i, y_i) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.3)$$

Результаты алгоритма кластеризации K -средних [32]:

- Центроиды кластеров k , которые можно использовать для маркировки новых данных;

- Метки для обучающих данных (каждая точка данных назначается одному кластеру).

Определение жанра произведения

Жанр произведения определяется так же по преимущественному цвету изображения. В статье [26] дано соответствие между преимущественным цветом и жанром произведения (таблица 2.7).

Таблица 2.7 – соответствие цвета и жанра

цвет	Музыкальный жанр
Синий	Блюз
Зеленый	Классика
Красный	Рок
Желтый	Классика

Определение схем соотнесения цвета и ноты

В статье [27] описывается множество подобных схем, например соотнесение цветов и нот по И. Ньютону, он искал связь между солнечным спектром и музыкальной октавой, сопоставляя длины разноцветных участков спектра и частоту колебаний звуков гаммы, таблица 2.8. В алгоритме будет использована эта схема, так как традиционно выбирают ее [8; 23–25].

Таблица 2.8 – Соотнесение цветов и нот по И. Ньютону

Цвет	Нота
Красный	До
Фиолетовый	Ре
Синий	Ми
Голубой	Фа
Зеленый	Соль
Желтый	Ля
Оранжевый	Си

2.4.2 Алгоритм генерации музыкального фрагмента

Для генерации музыкального фрагмента можно применить множество различных математических методов: марковские цепи, генетический алгоритм, порождающие грамматики и другие. В данной работе для решения этой

задачи будет применена рекуррентная архитектура искусственной нейронной сети. Этот тип сетей отлично выявляет и воспроизводит взаимосвязи в неструктурированных данных, что делает его эффективным для генерации музыки [33—35].

Рекуррентные нейронные сети

Рекуррентные нейронные сети содержат обратные связи. На рисунке 2.7 представлен фрагмент нейронной сети A принимает входное значение X_t и возвращает значение ht . Наличие обратной связи позволяет передавать информацию от одного шага сети к другому. Это обеспечивает некоторую форму памяти, что важно для генерации музыки, где каждая нота связана с предыдущими состояниями [33; 36].

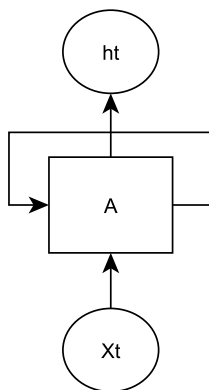


Рисунок 2.7 – Фрагмент рекуррентной нейронной сети

Однако у рекуррентных нейронных сетей возникает проблема долгосрочных зависимостей, которая проявляется в ограниченной способности корректного вычисления текущего состояния на основе далеких прошлых данных [33; 34; 36].

LSTM

Для решения данной проблемы были изобретены сети с долгой краткосрочной памятью или LSTM (от англ. long short-term memory) [36]. В отличие от обычных рекуррентных нейронных сетей, где повторяющийся модуль представляет собой функцию, в сетях LSTM этот модуль содержит четыре взаимодействующих между собой слоя: входной слой (фильтр), выходной слой, забывающий слой и ячейка памяти. Основной целью этих слоев является предотвращение данных от перезаписи и забывания. Такие нейронные сети наиболее хорошо подходят для анализа сложных структур, например,

текстовых и музыкальных данных, а также предсказания временных рядов [33]. Подробное описание и пошаговый разбор сети LSTM, предоставлены в статье [24].

Обучение сети LSTM

В алгоритме использован *midi* формат для обучения и предсказания композиций, так как нейронная сеть работает исключительно с числовой информацией. Особенности данного формата: данные о начале и окончании звучания конкретной ноты или набора нот представлены в числовом виде, при этом высота и ее название специальным образом зашифровано в числовые значения. Такой подход не требует дополнительных преобразований входных обучающих файлов — их сразу можно подготовить в *midi* формате [33].

Сценарий обучения сети описан в статье [33] и выглядит следующим образом: на заранее подготовленном наборе музыкальных композиций в *midi* формате для определенного жанра, обучаем сеть и сохраняем модель в отдельный файл для данного жанра. Обучение для определенного жанра позволит получить более подходящий выход сети. На рисунке 2.8 представлена схема верхнеуровневого алгоритма для генерации фрагмента музыкального произведения по изображению. Учтено, что доступны обученные модели на различных наборах данных по жанрам [33].

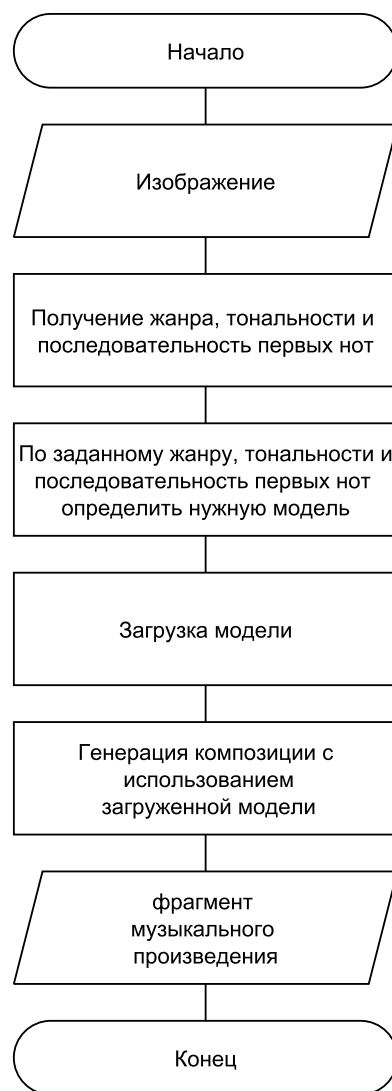


Рисунок 2.8 – Верхнеуровневый алгоритм для генерации фрагмента музыкального произведения по изображению

Вывод

Предложенный метод находит приближенную функцию многозначного отображения, которая описана формулой (2.1).

Метод генерации фрагмента музыкального произведения по изображению включает два основных этапа: анализ изображения и генерация музыкального фрагмента. На этапе анализа изображения получаются следующие характеристики: тональность, жанр и последовательность первых нот. Эти данные необходимы для последующей генерации музыкального фрагмента. Для определения тональности используются алгоритм кластеризации К-средних и соответствие цвета и тональности с учетом схемы А. Н. Скрябина. Определение жанра также базируется на алгоритме кластеризации К-средних и

соответствии цвета и жанра. Генерация музыкального фрагмента выполняется с использованием рекуррентной архитектуры нейронной сети LSTM. Обучение сети осуществляется на подготовленном наборе музыкальных композиций в *midi* формате для конкретного жанра.

3 Классификация существующих решений

Для краткости записи в данной таблице используются следующие обозначения описанных критериев:

- К1 — необходимость обучения перед генерацией фрагмента;
- К2 — возможность генерации в режиме реального времени;
- К3 — входные данные.

Таблица 3.1 – Сравнение различных методов

Метод	К1	К2	К3
Алгоритм Бьорклунда	Нет	Нет	Длина музыкального фрагмента, количество сигналов, аккорд
Генерация с помощью клеточного автомата	Нет	Нет	Набор правил перехода состояний, набор аккордов, соответствующий правилам
Метод случайного блуждания	Нет	Нет	Длина музыкального фрагмента, множество аккордов
Метод Марковских цепей	Да	Да	Длина музыкального фрагмента, множество аккордов
Порождающая грамматика	Нет	Нет	Набор правил перехода состояний, набор аккордов, соответствующий правилам
L-системы	Нет	Нет	Набор правил перехода состояний, соответствующий правилам, набор, состоящий из алфавита, аксиомы
LSTM	Да	Нет	Тональность, жанр и последовательность первых нот

ЗАКЛЮЧЕНИЕ

Цель работы достигнута, проведен анализ методов генерации музыкального фрагмента.

В итоге, в ходе данной работы были выполнены следующие задачи:

- проведен анализ предметной области;
- проведен обзор существующих методов и алгоритмов генерации музыкального фрагмента;
- сформулированы критерии сравнения методов генерации музыкального фрагмента;
- проведена классификация методов генерации музыкального фрагмента.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Миланич Е. Э., Лойко О. Б.* Учебно-методический комплекс по учебной дисциплине «Элементарная теория музыки». — 2017.
2. *Кузьмина Е. К., Катулин М. В., Пашкова М. М.* Сравнительный анализ использования цветовых пространств RGB и HSB в задачах распознавания объектов при определении позиции предметов в роботизированных системах Кузьмина [Электронный ресурс] // Системный анализ в науке и образовании. — 2022. — с. 55–63. — Режим доступа: <https://sanse.ru/index.php/sanse/article/view/520> (дата обращения: 2023-11-09).
3. *Котелина Н. О., Матвийчук Б. Р.* Кластеризация изображения методом К-средних // Вестник Сыктывкарского университета. — 2019. — т. Выпуск 3 (32). — с. 102–106.
4. What is artificial intelligence (AI)? [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/cis/artificial-intelligence/what-is-ai/> (дата обращения: 2023-11-10).
5. Electronic and computer music [Электронный ресурс]. — 2010. — Режим доступа: https://www.gramota.net/articles/issn_1993-5552_2010_1-2_08.pdf (дата обращения: 2023-11-10).
6. The purpose of the notes is their full definition and purpose [Электронный ресурс]. — 2010. — Режим доступа: <https://muzrock.com/teoriya-muzyki/noty> (дата обращения: 2023-11-10).
7. Chords. What is it and how are they built? [Электронный ресурс]. — . — Режим доступа: https://www.otsema.ru/accords/chto_takoe_akkord.php (дата обращения: 2023-11-10).
8. *Никитин Н. А.* Модели, методы и средства компьютерного синтеза музыки по цветовому изображению. — 2022.
9. *Chew E.* Mathematical and Computational Modeling of Tonality [Электронный ресурс]. — 2013. — Режим доступа: <https://link.springer.com/book/10.1007/978-1-4614-9475-1> (дата обращения: 20.11.2023).

10. *Conklin D.* Music Generation from Statistical Models [Электронный ресурс]. — 2016. — Режим доступа: https://www.researchgate.net/publication/2895833_Music_Generation_from_Statistical_Models (дата обращения: 10.10.2023).
11. *Quick D., Hudak P.* An automated combination of music based on computer graphics in Haskell [Электронный ресурс]. — 2013. — Режим доступа: https://www.researchgate.net/publication/261853177_Grammar-based_automated_music_composition_in_Haskell (дата обращения: 20.11.2023).
12. *Малахов В. Е., Rogozинский Г. Г.* Применение декомпозиции для изучения процессов при создании генеративной музыки. — 2022.
13. *Toussaint G.* The Euclidean Algorithm Generates Traditional Musical Rhythms [Электронный ресурс]. — 2005. — Режим доступа: <http://cgm.cs.mcgill.ca/~godfried/publications/banff.pdf> (дата обращения: 2023-10-10).
14. *Lo. M. Y.* Evolving Cellular Automata for Music Composition with Trainable Fitness Functions [Электронный ресурс]. — 2012. — Режим доступа: https://www.researchgate.net/publication/349061035_Evolving_Cellular_Automata_for_Music_Composition_with_Trainable_Fitness_Functions (дата обращения: 2023-10-10).
15. *D C.* Music Generation from Statistical Models [Электронный ресурс]. — 2016. — Режим доступа: https://www.researchgate.net/publication/2895833_Music_Generation_from_Statistical_Models (дата обращения: 2023-11-10).
16. *Niedek T. V.* Random Walk with Restart for Automatic Playlist Continuation and Query-Specific Adaptations [Электронный ресурс]. — 2018. — Режим доступа: https://www.researchgate.net/publication/328090223_Random_Walk_with_Restart_for_Automatic_Playlist_Continuation_and_Query-Specific_Adaptations (дата обращения: 2023-11-10).
17. *B H.* First Links in the Markov Chain. American Scientist [Электронный ресурс]. — 2013. — Режим доступа: https://www.researchgate.net/publication/289803891_First_Links_in_the_Markov_Chain (дата обращения: 2023-11-10).

18. Марковская цепь для генерации музыки [Электронный ресурс]. — Режим доступа: <https://machinelearningmastery.ru/markov-chain-for-music-generation-932ea8a88305/> (дата обращения: 2023-10-25).
19. The Beatles – Yesterday [Электронный ресурс]. — Режим доступа: <https://www.5lad.ru/akkordy/the-beatles/yesterday> (дата обращения: 05.11.2023).
20. *Quick D., Hudak P.* An automated combination of music based on computer graphics in Haskell [Электронный ресурс]. — 2013. — Режим доступа: https://www.researchgate.net/publication/261853177_Grammar-based_automated_music_composition_in_Haskell (дата обращения: 2023-11-10).
21. Review of mathematical methods for generating musical compositions [Электронный ресурс]. — 2020. — Режим доступа: <https://moluch.ru/archive/342/77077> (дата обращения: 2023-11-10).
22. Chapter 44. L-systems [Электронный ресурс]. — 2013. — Режим доступа: http://mech.math.msu.su/~shvetz/54/inf/perl-examples/PerlExamples_LSystems.xhtml (дата обращения: 2023-11-10).
23. *Никитин Н. А., Розалиев В. Л., Орлова Ю. А.* Разработка веб-сервиса для генерации музыкальной последовательности по изображению [Электронный ресурс] // Молодой ученый. — 2019. — т. 51, № 289. — с. 27—30. — Режим доступа: <https://moluch.ru/archive/289/65648/> (дата обращения: 2023-10-06).
24. *Никитин Н. А., Орлова Ю. А., Розалиев В. Л.* Программная генерация звуков по цветовой гамме изображений с использованием рекуррентной нейронной сети [Электронный ресурс] // Первая Всероссийская научно-практическая конференция. — 2017. — с. 105—113. — Режим доступа: http://fti.ulstu.ru/wp-content/uploads/2022/01/Proceedings_FTI_2017_final.pdf (дата обращения: 2023-09-11).
25. *Никитин Н. А., Орлова Ю. А., Розалиев В. Л.* Алгоритм генерации музыкальных композиций с использованием интуитивного и эмоционального подходов [Электронный ресурс] // Молодой ученый. — 2021. — т. 24, № 366. — с. 35—39. — Режим доступа: <https://moluch.ru/archive/366/82308/> (дата обращения: 2023-10-06).

26. *Никитин Н. А., Орлова Ю. А., Розалиев В. Л.* Автоматическая генерация звуков по цветовой гамме изображений // Вестник Волгоградского государственного технического университета. — 2016.
27. *Abdullayev E.* An Experimental Model Of Color And Sound Correlation [Электронный ресурс] // Eurasian music science journal. — 2020. — Режим доступа: <https://core.ac.uk/download/pdf/336866842.pdf> (дата обращения: 2023-10-07).
28. *Jain A., Murty M., Flynn P.* Data clustering: A review // ACM Computing Surveys. — 1999. — т. 31.
29. *Тюрин А. Г., Зуев И. О.* Кластерный анализ, методы и алгоритмы кластеризации [Электронный ресурс] // Вестник МГТУ МИРЭА. — 2014. — т. Выпуск 3. — Режим доступа: <https://rtj.mirea.ru/upload/medialibrary/fba/09-tyurin.pdf> (дата обращения: 2023-11-10).
30. *Ооржак О. Е.* Алгоритм кластеризации К-средних и его реализация в среде MATLAB. — 2015.
31. *Coates A., Ng A. Y.* Learning Feature Representations with K-means [Электронный ресурс]. — 2012. — Режим доступа: https://web.archive.org/web/20150621073741/http://www.cs.stanford.edu/~acoates/papers/coatesng_nntot2012.pdf (дата обращения: 2023-11-10).
32. *Oyelade O., Oladipupo O., Obagbuwa I.* Application of k-Means Clustering Algorithm for Prediction of Students Academic Performance // International Journal of Computer Science and Information Security. — 2010. — т. 7, № 1. — с. 292—295.
33. *Никитин Н. А., Орлова Ю. А., Розалиев В. Л.* Генерация жанровых музыкальных композиций по эмоциональному состоянию человека [Электронный ресурс] // Моделирование, оптимизация и информационные технологии. — 2022. — т. 10, № 2. — с. 1—14. — Режим доступа: https://web.archive.org/web/20220707054841id_/https://moitvivi.ru/journal/pdf?id=1175 (дата обращения: 2023-11-10).
34. *Толстов М. В., Соломатин Д. И.* Применение нейронных сетей для генерации мелодий // сборник студенческих научных работ факультета компьютерных наук ВГУ. — 2020. — т. 14. — с. 185—190.

35. *Токтарбеков М. А., Сарсембаев А. А.* Особенности использования RNN-LSTM и GAN архитектур при генерации музыки // Международный университет информационных технологий. — 2022.
36. *Линдигрин А. Н.* Искусственные нейронные сети как основа глубинного обучения // Известия ТулГУ. Технические науки. — 2019. — т. 12.