информатика

Компьютерные инструменты в образовании, 2023
№ 1: 74–95
УДК: 78.02: 81'322.6
http://cte.eltech.ru
doi:10.32603/2071-2340-2023-1-74-95

MUSIC INFORMATION RETRIEVAL — СОВРЕМЕННЫЕ ЗАДАЧИ И ТЕХНОЛОГИИ

Абросимов К. И.[1], студент, ✉ abrosimov.kirill.1999@mail.ru, orcid.org/0000-0001-9262-0474
Рыбин С. В.[1,2], канд. физ.-мат. наук, доцент, svrybin@itmo.ru,
orcid.org/0000-0002-9095-3168

[1]Санкт-Петербургский национальный исследовательский университет ИТМО,
Кронверкский пр., 49, лит. А, 197101, Санкт-Петербург, Россия
[2]Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова
(Ленина), ул. Профессора Попова, 5, корп. 3, 197022, Санкт-Петербург, Россия

Аннотация

В работе рассматривается Music Information Retrieval — область вычислительного музыковедения, которая активно развивается в современном мире. В рамках статьи описаны некоторые основные задачи и технологии данного направления, такие как генерация музыки, автоматическая музыкальная транскрипция, синтез звуков музыкальных инструментов, поиск музыки.

Особое внимание уделяется одной из интереснейших задач на стыке речевых и музыкальных технологий — синтезу поющего голоса. Рассматриваются различные подходы к этой задаче, существующие проблемы и методы их решения.

Ключевые слова: *вычислительное музыковедение, music information retrieval, генерация музыки, автоматическая музыкальная транскрипция, синтез звуков музыкальных инструментов, поиск музыки, синтез певческого голоса.*

Цитирование: Абросимов К. И., Рыбин С. В. Music information retrieval — современные задачи и технологи // Компьютерные инструменты в образовании. 2023. № 1. С. 74–95. doi:10.32603/2071-2340-2023-1-74-95

1. ВВЕДЕНИЕ

Вычислительная музыковедение (*англ.* Computational musicology) — это современная междисциплинарная наука на стыке музыковедения, информатики и прикладной математики, изучающая музыку с помощью вычислительных и компьютерных методов [1]. Автоматическая обработка музыки позволяет автоматизировать многие процессы в музыкальной индустрии — от автоматической музыкальной транскрипции до синтеза певческого голоса исполнителей прошлых лет.

В рамках вычислительного музыковедения отдельное положение занимает моделирование и представление музыки, анализ звучания музыкальных инструментов с помощью различных вычислительных методов и технологий [2].

Данное научное направление основывается на следующих ключевых областях:

• **Музыковедение**. Вычислительное музыковедение использует методологию анализа, выводы и сравнительные результаты [3].

• **Математическая лингвистика**. В вычислительном музыковедении для анализа нот — формального языка музыки — активно применяются методы математической лингвистики [4–6]. Технологии обработки естественного языка позволяют относиться к нотам, как к некоторому тексту, применять модели дистрибутивной семантики и NLP[1] в целом [7, 8].

• **Машинное обучение**. Применение методов машинного обучения даёт возможность строить модели, которые могут извлекать информацию и обрабатывать музыкальные сигналы, например, определять сходства мелодий [9] или разделять музыку и вокал (подробный обзор методов можно найти в [10]).

• **Цифровая обработка сигналов и речевые технологии**. Методы цифровой обработки сигналов [11, 12] позволяют анализировать звучащую музыку, например, для решения задачи музыкальной транскрипции [13], а речевые технологии, в частности, методы синтеза речи [14–16] активно используются при решении задачи синтеза вокализованного голоса. Обзор технологий синтеза речи можно найти в [17].

• **Методы оптимизации**. С помощью методов оптимизации, в том числе эволюционных и популяционных, процесс генерации музыки можно представить как результат решения некоторой комбинаторной задачи оптимизации [18].

В данной статье основное внимание будет уделено важнейшему направлению вычислительного музыковедения — Music Information Retrieval.

Music Information Retrieval (MIR) — это раздел вычислительного музыковедения, анализирующий процессы и представления данных для извлечения различной информации из музыки. Одно из основных прикладных применений — анализ и построение музыкальных интеллектуальных систем.

В 2000 г. был образован международный форум по исследованию данных, связанных с музыкой, — *международное общество поиска музыкальной информации* (*англ.* International Society for Music Information Retrieval, ISMIR). В 2002 году он был преобразован в ежегодную конференцию, сохранив прежнюю аббревиатуру.

2. ОСНОВНЫЕ НОТАЦИИ

Ноты — это рукописная или печатная форма записи музыки, которая использует символы для обозначения высоты тона, ритма, аккордов песни или инструментального музыкального произведения. Для решения задач вычислительного музыковедения ноты являются одной из самых важных информационных моделей, при этом существуют совершенно различные нотные представления для обработки музыки [19].

Рассмотрим различные форматы нотной записи.

• **Графическое представление** — является классическим, однако для анализа является достаточно трудоемким, так как необходимо применять методы компьютерного зрения, которые могут допускать ошибки при обучении и использовании модели детектирования нот. Пример графического представления представлен на рис. 1.

[1] NLP — Natural Language Processing — обработка естественного языка.

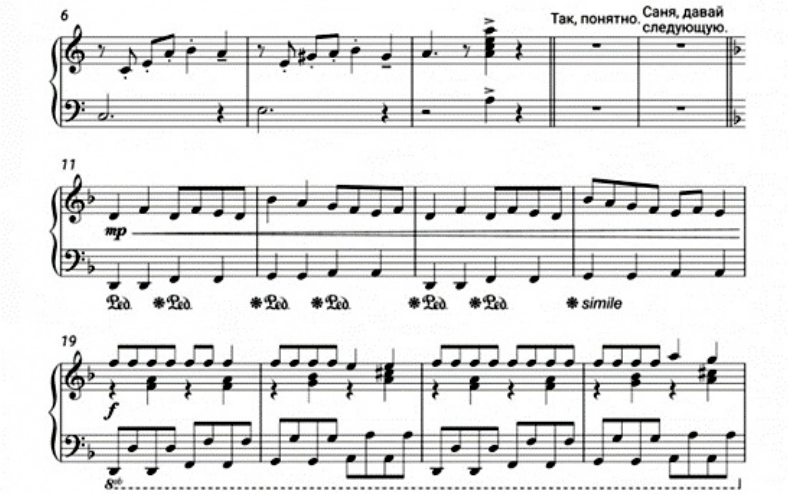

Рис. 1. Графическое представление нот

• **MusicXML** [20] — представление нот с помощью языка разметки XML[2] и строгой спецификации. Тип нотации является удобным для компьютерной обработки. Существует большое количество библиотек для эффективной обработки языка XML (рис. 2).

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 4.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="4.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Рис. 2. Представление нот в MusicXML

[2] XML — eXtensible Markup Language — расширяемый язык разметки.

• **MIDI** (*англ.* Musical Instrument Digital Interface) — цифровой интерфейс музыкальных инструментов [21] — стандарт цифровой звукозаписи для обмена данными между электронными музыкальными инструментами (рис. 3).

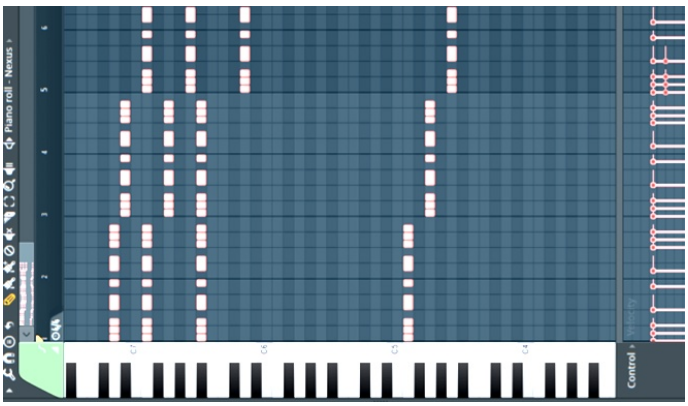

Рис. 3. Пример визуализации MIDI файла

Интерфейс позволяет единообразно кодировать в цифровой форме такие данные, как нажатие клавиш, настройку громкости и других акустических параметров, выбор тембра, темпа, тональности и др. с точной привязкой во времени.

В системе кодировок присутствует множество свободных команд, которые производители, программисты и пользователи могут использовать по своему усмотрению. Поэтому интерфейс MIDI позволяет, помимо исполнения музыки, синхронизировать управление другим оборудованием, например, осветительным, пиротехническим и т. п.

• **ABC-нотация** [22] и **LilyPond нотация** [23] — это текстовые нотации, с определенными правилами оформления нот и различных музыкальных символов. На рисунке 4 представлен пример ABC-нотации и его графическое представление, а на рисунке 5 представлен пример LilyPond нотации.

```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
</score>
```

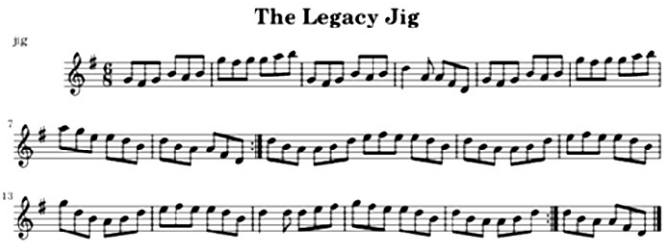

Рис. 4. ABC-нотация

```
    %% Пьеса начинается с затакта длиной в четверть, "\partial 4"
    %% на это и указывает.
    \partial 4
    a4 | e'4.( d8[ c]) r8 | d4.( c8[ h]) r8 | a2. | e2
    a4 | e'4.( d8[ c]) r8 | d4.( e8[ f]) r8 | e2. | r2
    e4 |  f4.( e8[ d]) r8 | d4.( c8[ h]) r8 | a2. | e2
    a4 | e'4.( d8[ c]) r8 | d4.( c8[ h]) r8 | a2. ~ a2 r4 | \bar "|
}
```

Рис. 5. LilyPond нотация

Для компьютерной обработки наиболее удобны форматы MIDI и MusicXML. Чуть сложнее обстоят дела с другими нотациями, но с помощью методов математической лингвистики (автоматные модели и проч.) можно выделять необходимую информацию.

3. МУЗЫКАЛЬНАЯ АЛГОРИТМИЧЕСКАЯ КОМПОЗИЦИЯ

Одной из фронтирных задач, решаемых MIR на основе нотаций, является алгоритмическая композиция, или, по-другому, генерация музыки. Данная задача известна еще с прошлого столетия и по сей день является очень значимой и актуальной. Например, с помощью автоматической генерации музыки можно создавать различные фоновые мелодии для компьютерных игр или видеороликов, при этом данные записи будут дешевле, чем музыка, написанная профессиональными композиторами.

Рассмотрим основные модели алгоритмической композиции, применяющие различные методы и технологии.

• **Трансляционные модели** — это модели, основанные на переносе информации из какого-либо процесса или объекта в музыку. Например, перенос информации о яркости пикселей на картинке в музыку или перенос информации из текста в музыку [24].

• **Математические модели**. На протяжении столетий (начиная с Пифагора) теория музыки была тесно связана с математикой. Здесь будет уместно упомянуть работу Леонарда Эйлера «Опыт новой теории музыки», опубликованную в Петербурге в 1739 г. на латинском языке. В ней Эйлер попытался математическим языком объяснить некоторые явления современной ему музыки. Следует отметить, что применение математического моделирования для музыки является достаточно сложной задачей, как правило, с применением экспертных оценок [25]. В [26] представлена библиотека Gelisp, позволяющая представлять некоторые гармонии с помощью таких моделей.

• **Модели на основе математической лингвистики**. Эти модели работают на основе правил, полученных из теории формальных языков и грамматик, а также логических языков, таких как *Haskell*. Например, порождающая грамматика хорошо решает задачу генерации музыкальной партитуры в классической нотации [27]. Другой пример — применение систем Линденмайера[3] для автоматической генерации музыки [28].

Несмотря на то что эти модели не являются моделями в парадигме машинного обучения, они не уступают моделям на основе современных глубоких нейронных сетей. Например, в соревновании по алгоритмической композиции Yet Another Data Challenge[4]

[3] Системы были предложены в 1968 г. венгерским ботаником Аристидом Линденмайером для изучения развития простых многоклеточных организмов.

[4] Онлайн-соревнование по созданию генеративной музыки, организованное сообществом разработчиков в сфере ИИ — AI Community и облачной платформой Yandex.Cloud. Задача — обучить ML-модель, чтобы она смогла продолжить мелодию по заданным первым 8 тактам.

5 первых мест получили именно модели на основе правил. Оценивали данные произведения эксперты по строгой спецификации [5].

• **Модели на основе оптимизации** — здесь проблема формулируется как задача комбинаторной оптимизации с построением целевой функции с помощью экспертов-музыкальных теоретиков [29].

• **Эволюционные модели генерации музыки**. Данный подход основан на применении эволюционных и популяционных алгоритмов оптимизации. Первым успешным примером применения таких алгоритмов в генерации музыки можно считать программу *GenJam* Джона Бильса [30]. Данная программа работает с MIDI файлами, включающими партии аккордов пианино, баса, ритм–секции, и генерирует соло. Основной сложностью данных моделей является оценка полученных музыкальных фрагментов с помощью экспертов [31].

• **Применение методов баз знаний и экспертных систем**. С помощью экспертных систем можно формализовать априорные теоретические музыкальные знания и использовать их для дальнейшей генерации с помощью логики, нечеткой логики и других инструментов [32]. Вероятно, первой работой по генерации музыки на основе базы знаний является произведение «Сюита Иллиака»[5], в которой разработчики использовали классические правила контрапункта[6].

• **Модели машинного обучения**. Современные методы алгоритмической композиции в основном строятся на этих моделях. Для этого используют различные архитектуры глубоких нейронных сетей, таких как сверточные, рекуррентные и трансформерные нейронные сети. Первоначально, в конце XX в., нейронные сети использовались для анализа музыкальных композиций, но затем стали применяться и для генерации музыки. Один из первых примеров такого применения — модель трехслойной рекурсивной сети [33]. В настоящее время наиболее активно применяются различные современные модели вычислительной лингвистики и NLP [34].

• **Гибридные модели** представляют собой совокупность вышеназванных семейств моделей. Предлагаемая в [35] модель состоит из моделей математической лингвистики и вычислительной. С помощью автоматных моделей происходит разбиение ABC-нотации на мелодические и ритмические конструкции. С помощью трансформерных нейронных сетей генерируется продолжение мелодии и далее с помощью автоматной модели преобразуется в формат ABC-нотации (рис. 6).

Как правило, алгоритмическая композиция состоит не только из генерации основной мелодии, но также и из генерации вспомогательных, например автоматической генерации аккомпанемента.

В работе [36] для этой задачи предложена архитектура нейронной сети, которая на основе рекуррентных блоков просмотра основной мелодии до просматриваемой ноты предсказывает ноты в формате one-hot в определенный момент времени.

Для представления нот используется Piano Roll формат, в котором музыкальный фрагмент разбивается на небольшие отрезки времени, как правило включающие в себя определенные длительности нот (например 1/16). При этом 1 ставится в фрагменте напротив той ноты, которая в данный момент должна играть, и 0 — в другом случае (рис. 7).

В результате модель обучается генерировать аккомпанемент на основе данных, состоящих из 389 Хоралов Баха [37].

[5] Композиция для струнного квартета, написанная в 1957 г. компьютером ILLIAC I в Иллинойском университете. Соавторы — композитор Лейярен Хиллер и программист Леонард Айзексон.

[6] Одновременное сочетание двух или более самостоятельных партий музыкальных инструментов.

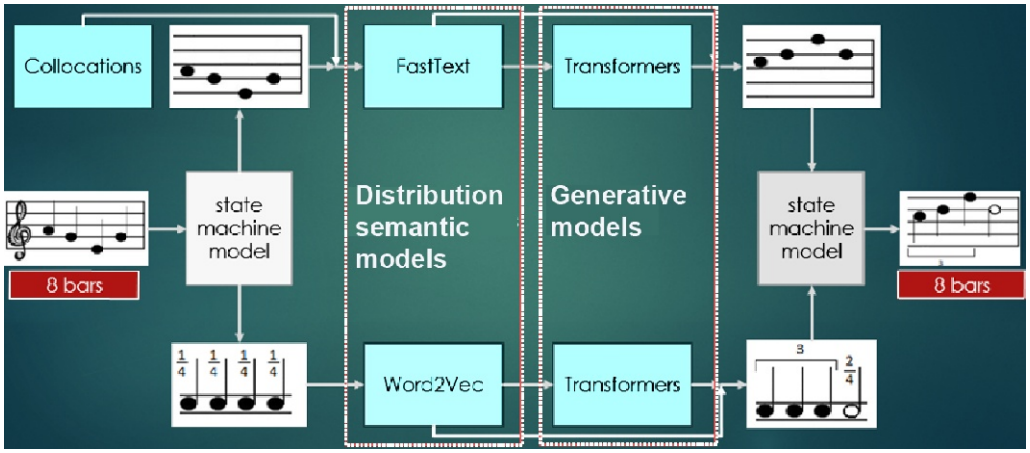

Рис. 6. Гибридная модель алгоритмической композиции

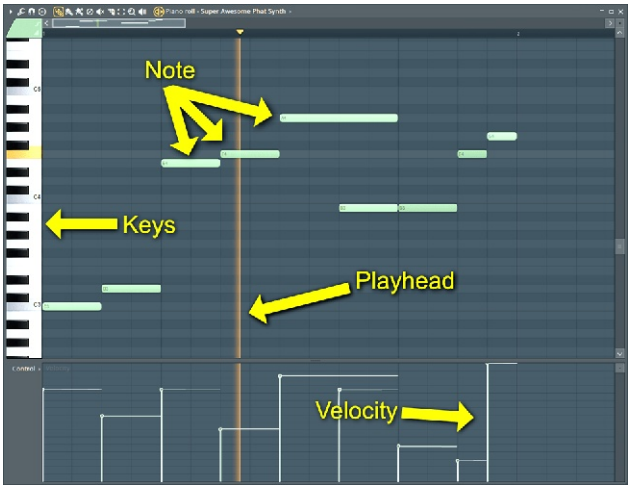

Рис. 7. Piano Roll формат представления нот

Одной из самых сложных задач алгоритмической композиции является генерация полифонической музыки, так как эта музыка имеет несколько мелодий, которые гармонично звучат как сами по себе, так и между собой.

Для решения этой задачи в работе [38] представлена очень интересная архитектура рекуррентной нейронной сети. Она состоит из 4 основных LSTM[7] блоков, причем два LSTM слоя направлены на обработку временной зависимости, а оставшиеся два LSTM слоя — на генерацию полифонии, то есть автор считает, что главная мелодия — верхняя, а остальные подстраиваются под неё. Визуализация работы сети представлена на рисунке 8.

4. АВТОМАТИЧЕСКАЯ ТРАНСКРИПЦИЯ МУЗЫКИ

Одной из главных задач MIR является автоматическая музыкальная транскрипция. Эта задача позволяет автоматически переносить звучащую музыку в формат определен-

[7] Нейронная сеть с долгой краткосрочной памятью (*англ.* Long-Short Term Memory, сокращенно LSTM [39]).

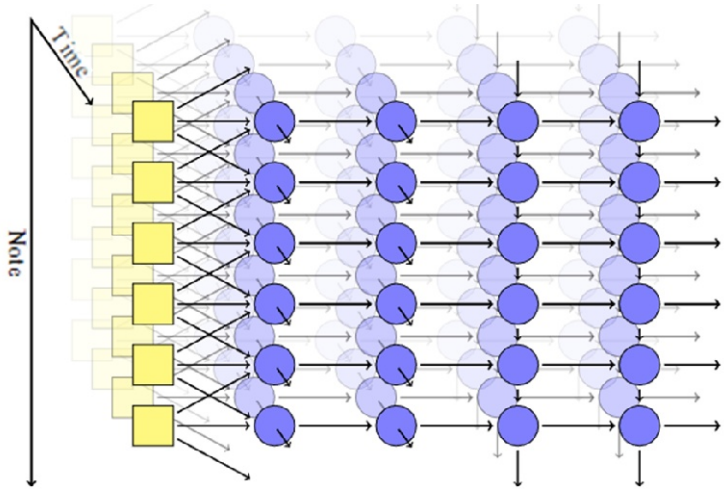

Рис. 8. Модель генерации полифонической музыки [38]

ной нотации. Как правило, из музыкального фрагмента извлекают определенные акустические признаки (mel-спектрограммы[8], хроматограммы[9], MFCC[10]), а далее с помощью статистических и data driven моделей производят предсказание нот.

В работе [41] представлена модель, которая в две основные фазы производит формирование MIDI представления звучащей музыки. Первый этап — это нахождение начала звучания определенного звука, второй этап — предсказание длительности нот. Оба этапа производятся с помощью двунаправленных LSTM-слоев с последующей пост-обработкой. На рисунке 9 схематично представлена архитектура нейросетевой модели.

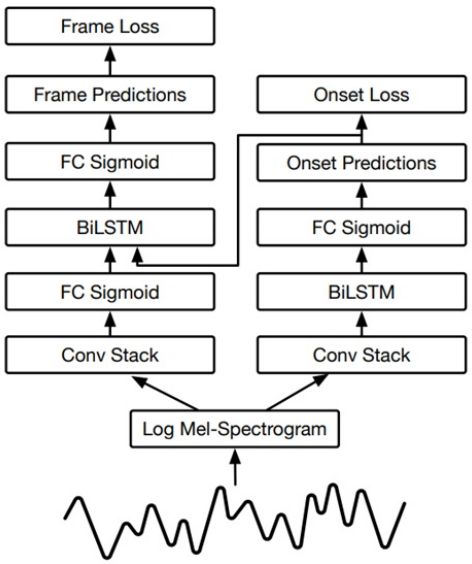

Рис. 9. Двухэтапная модель музыкальной транскрипции [41]

[8] Mel-спектрограмма — это матрица, где каждый столбец является спектром короткого участка сигнала, причем частота выражена не в Гц, а в психофизических единицах высоты звука (мелах).

[9] Цветовое представление гармонических и мелодических характеристик музыки.

[10] Мел-кепстральные коэффициенты (*англ.* Mel Frequency Cepstral Coefficient, сокращенно MFCC [40]).

Похожим образом работает еще одна модель, представленная в работе [42], только обучение проводится не в два отдельных этапа, а последовательно с передачей информации между моделями, подобно передаче информации Residual блоку.

Сначала происходит распознавание и квантизация[11] ноты, разбиение по тактам. Затем — предсказание длительности отдельной ноты, а также предсказание, какой рукой должна играться обрабатываемая нота (рис. 10). Единственное ограничение у системы — она нацелена на обработку фортепианных партий.

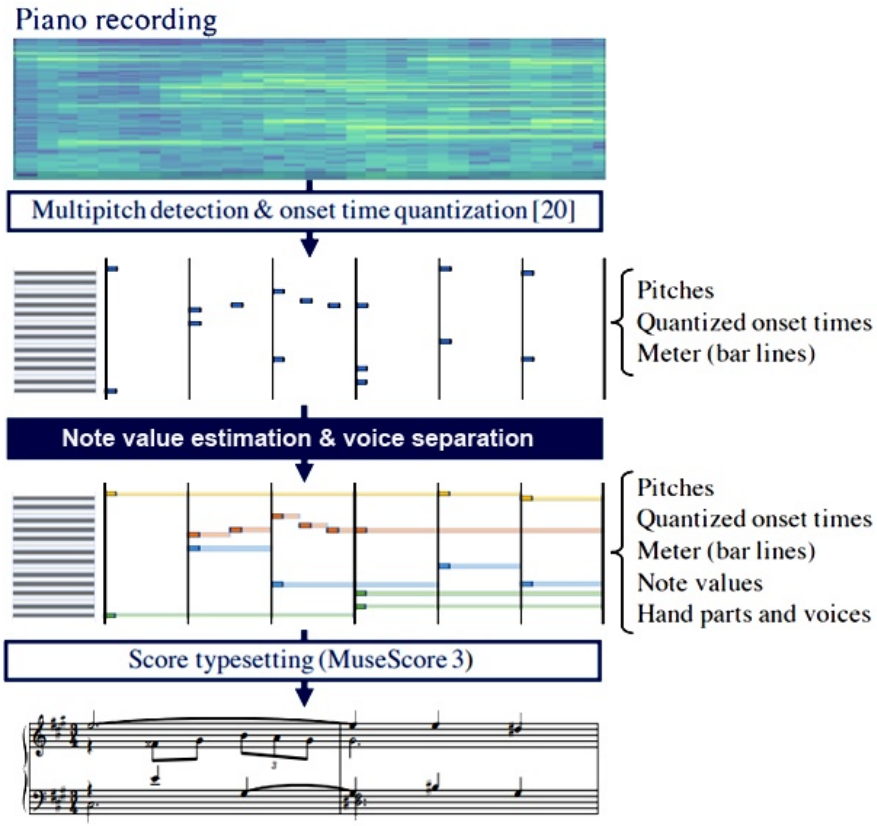

Рис. 10. Модель распознавания фортепианных партий [42]

Несмотря на большое количество различных моделей автоматической транскрипции, эксперименты показали, что ни одна из моделей не является лучше других при их применении к различным музыкальным произведениям и жанрам [43]. Исследователи применили восемь разных моделей и сравнили с истинной разметкой и нотами, полученными экспертами. Последние получили качество более 90 %, в то время как модели — меньше 60 %. Этот факт еще раз подтверждает, что работа с музыкой весьма трудна.

5. СИНТЕЗ ЗВУКОВ МУЗЫКАЛЬНЫХ ИНСТРУМЕНТОВ

Попытки синтеза музыки имеют давнюю историю. Со времен уже упоминавшегося древнегреческого математика Пифагора один из важнейших вопросов в музыкальной

[11] Квантизация — процесс выравнивания множества нот, тонкая настройка их положения относительно такта.

акустике — *«Как данный музыкальный инструмент издает свой характерный звук?»*. В настоящее время имеются подробные исследования о функционировании музыкальных инструментов [44].

Классические методы синтеза звука музыкальных инструментов используют эффекты обработки сигнала, такие как, например, частотная модуляция или семплирование. Первый из них основан на взаимной модуляции по частоте между несколькими гармоническими сигналами. Использование данного метода на практике достаточно сложно ввиду трудностей правил настройки. Семплерный метод работает с записями реального звука, меняя их через определенные интервалы с возможным изменением скорости воспроизведения. Его существенным недостатком является падение естественности звука при изменении параметров. Современные результаты синтеза звучания различных музыкальных инструментов на основе классических методов — высоки, однако слушатели зачастую способны отличить синтезируемые звуки от реальных.

В последние годы новым инструментом для синтеза стали методы машинного обучения, которые ранее были недоступны из-за их высоких вычислительных затрат. Использование методов глубокого машинного обучения для аудио синтеза было названо **нейронным аудио синтезом** (*англ.* Neural Audio Synthesis, NAS).

Одним из первых решений в этой области была интегральная модель на основе автоэнкодера [45], разработанная исследовательской группой в рамках проекта Google Magenta[12].

В работе [46] исследователи предложили использовать перспективную архитектуру GAN [47][13] для генерации новых фрагментов и использования их, например, в задаче алгоритмической композиции.

Однако все чаще современные композиторы и звукорежиссеры хотят перекладывать определенные партии на другие инструменты, при этом затрачивая как можно меньше времени. Для такого типа задач предложена модель [48], которая выделяет частоту основного тона[14] из перезаписанного или спетого фрагмента и перекладывает предсказанную частоту основного тона на один из выбранных инструментов: флейта, скрипка, труба. На рис. 11 представлена схема модели.

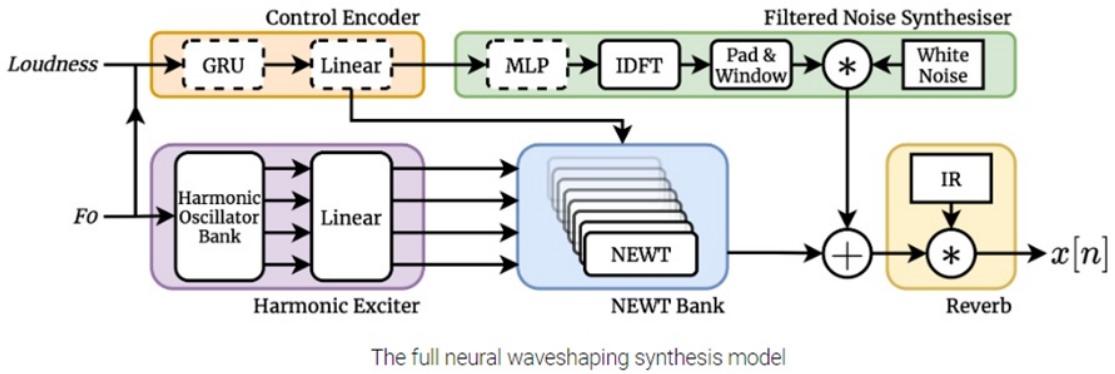

Рис. 11. Предлагаемая модель синтеза звуков [48]

[12] Magenta — это исследовательский проект с открытым исходным кодом, изучающий роль машинного обучения как инструмента в творческом процессе.

[13] Генеративная состязательная сеть (*англ.* Generative adversarial network, сокращенно GAN).

[14] Частота основного тона (F0) определяется как частота вибрации голосовых связок говорящего.

Модель состоит из Harmonic Exciter[15], вызывающего определенный сигнал на основе частоты основного тона, который, в свою очередь, с помощью синусоидального многослойного персептрона производит предсказание частот со специфическими акустическими атрибутами определенного инструмента. Также добавляется шум, который моделирует реальное пространство. Эти блоки и позволяют добиться реальности звучания инструментов.

6. МУЗЫКАЛЬНЫЙ ПОИСК

На данный момент, одной из самых используемых систем MIR является система музыкального поиска Shazam. Впервые эта система появилась в 2003 г.

Основная задача данных систем — на основе звучащего фрагмента музыкального произведения указать пользователю, какая музыкальная композиция воспроизводится [49].

Изначально в основе системы лежат структуры данных, позволяющие быстро находить в базе данных экземпляры музыкальных произведений. Для этого, как правило, используются различные хеш-таблицы. Для разметки музыкальных произведений и поступивших фрагментов производится оконное преобразование Фурье и выделяются так называемые якорные точки — точки, имеющие наибольшую частотную амплитуду на спектрограмме в каждом временном окне. Полученные в результате этих преобразований значения сохраняются в базе данных.

Одним из главных недостатков данной системы является ограниченность в поиске, то есть любой cover[16] на популярное произведение не выдаст никакой информации, так как оригинальное произведение и его cover-версия, как правило, имеют разный «цифровой код», хотя музыкальное произведение одно и то же.

Начиная с 2008 г., в рамках конференций ISMIR ежегодно проводится оценка алгоритмов музыкального информационного поиска (англ. Music Information Retrieval Evaluation eXchange, MIREX).

7. СИНТЕЗ ПЕВЧЕСКОГО ГОЛОСА

Направление синтеза певческого голоса (*англ.* Singing Voice Synthesis, SVS) развивается с конца прошлого столетия. Одним из важных мотивационных факторов для развития SVS является неизбежность ухода известных исполнителей из жизни. При наличии хороших аудиозаписей, правильной разметки данных можно построить модель, которая позволит создавать новые музыкальные произведения любимых актеров. На данный момент эти технологии активно продвигаются азиатскими станами, такими как Южная Корея, Япония, Китай, на западе — Испания и США.

Задача синтеза певческого голоса находится на стыке музыкальных технологий и автоматического синтеза речи (*англ.* Text To Speech, TTS [50, 51]). Еще совсем недавно основными методами SVS, как и в задаче синтезе речи, являлись канкатенативные методы, однако уже сегодня, как правило, используются современные нейросетевые модели.

В настоящее время для генерации информации популярными архитектурами являются различные модификации уже упомянутых GAN. Традиционно GAN состоят из конкурирующих нейронных сетей, которые условно разделяются на генератор и дискриминатор, обучающиеся поочередно. Дискриминатор учится определять, какие параметры

[15] Звуковой усилитель, добавляет небольшие гармонические искажения, звук становится более «прозрачным» и насыщенным.

[16] Обработка оригинального произведения с элементами новой аранжировки.

получены из реального сигнала, а какие — «не настоящие» (сгенерированные). Соответственно, генератор обучается «обманывать» дискриминатор. Такая схема и используется в работе [52]. Пример архитектуры представлен на рисунках 12 (общая схема GAN) и 13 (генератор, акустическая модель.)

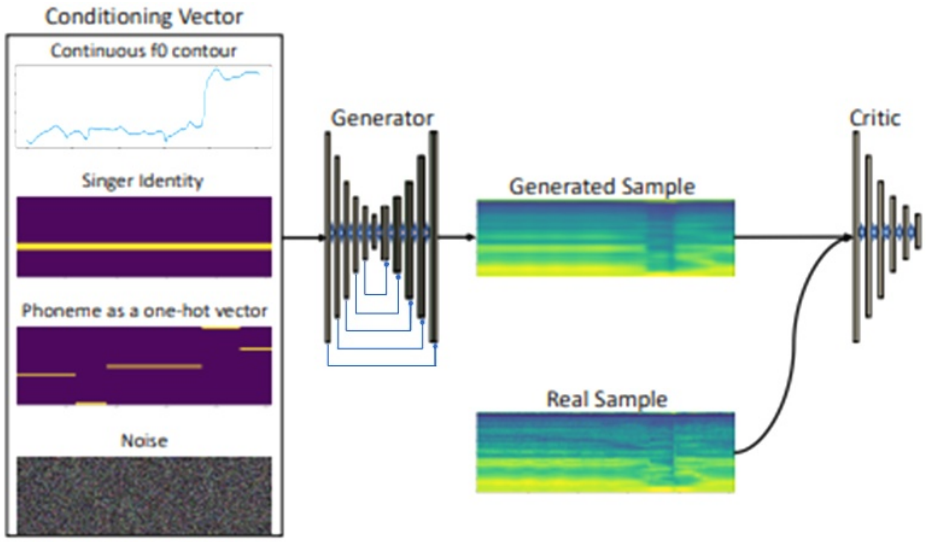

Рис. 12. Общая схема GAN для решения задачи SVS [52]

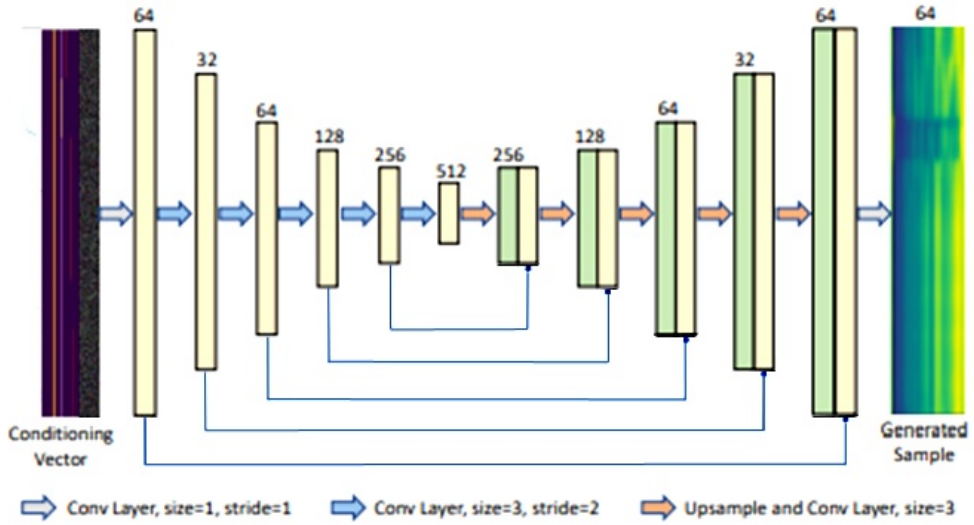

Рис. 13. Генератор в GAN для решения задачи SVS [52]

В SVS, как и в задачах синтеза речи, появляется необходимость клонировать тембр или стиль пения, поэтому возникли модели для решения таких задач. Так, в работе [53] представлена модель для переноса стиля и тембра на основе мел-спектрограмм и сверточных слоев, маскирования для совмещения.

На рисунке 14 представлена архитектура GAN из [53] для клонирования тембра и стиля пения: добавлены специальные блоки-кодировщики, позволяющие выделять из акустических признаков необходимую информацию, характерную для данного исполнителя: тембр и певческий стиль.

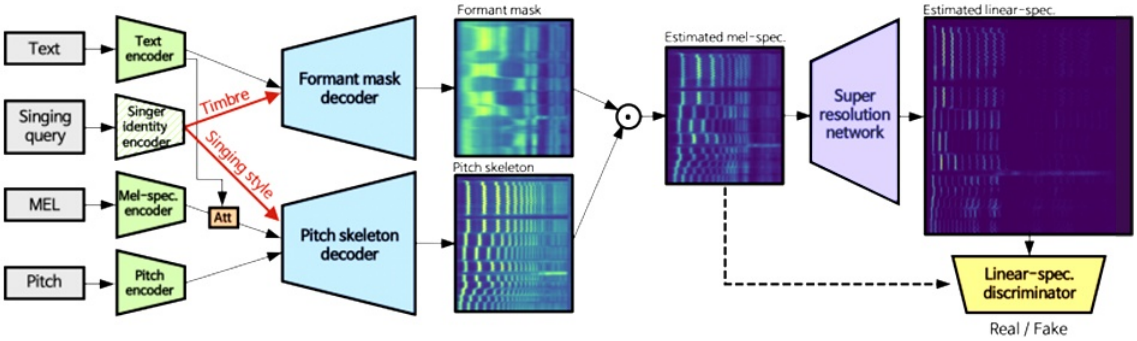

Рис. 14. Архитектура GAN для клонирования тембра [53]

В работе [54] представлена архитектура на основе векторно-квантованного вариационного автоэнкодера (*англ.* Vector Quantized Variational Autoencoders, VQ-VAE) для генерации пения на основе только текста песни, без нотных партитур. Благодаря большому количеству данных, происходит «запоминание» музыкальных интонаций фонем[17], поэтому при генерации на основе текста происходит некоторая «музыкальная аппроксимация». На рисунке 15 представлена общая схема модели.

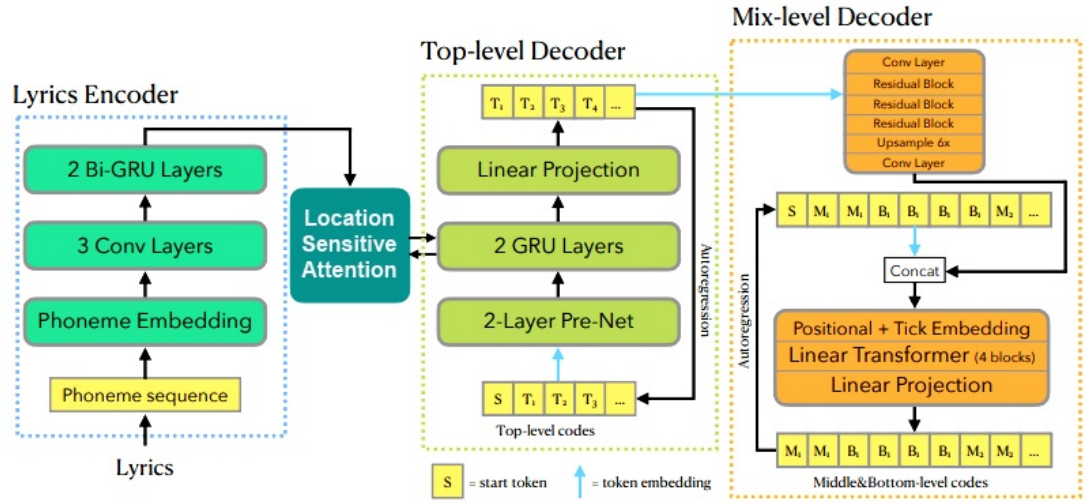

Рис. 15. Схема вокализированной речи без партитур [54]

Модели синтеза пения, как и модели синтеза речи, могут быть как End-to-End[18] системами, так и последовательными pipeline-системами (аналог технологии статистического параметрического синтеза TTS, см. например [55]). В работе [56] представлен пример pipeline-системы. Данная система очень похожа на последовательность задач синтеза речи, только с увеличенным количеством данных. На рисунке 16 представлена общая схема такой pipeline-системы.

[17] Фонема — обозначение звуковой единицы, элемент звуковой системы языка. Это абстрактная лингвистическая сущность, единица в отвлечении от ее конкретных реализаций.

[18] End-to-End (интегральная или сквозная) система объединяет в себе модули pipeline-системы в единую модель и непосредственно связывает вход и выход (end-to-end).

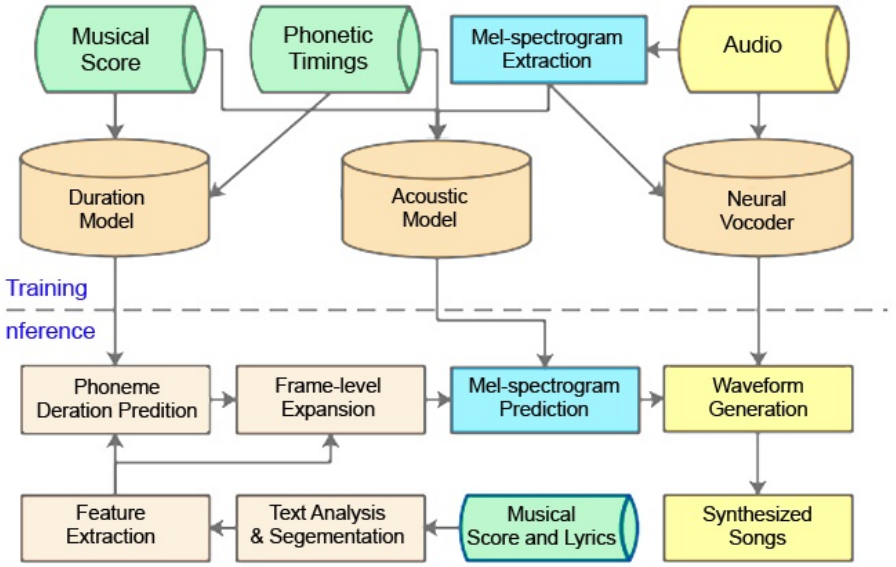

Рис. 16. Схема pipeline-системы синтеза пения [56]

Хорошим примером End-to-End системы является модель из работы [57]. Исследователи применили модель Fast Speech [58] для генерации певческого голоса. Теперь на вход модели приходит не только речь, но и нотная партитура в формате, высоты нот и длительности. Кроме того, применена усложненная постобработка перед модулем вокодера (рис. 17).

Основная сложность при решении задачи SVS — небольшое количество данных, так как, для того чтобы собрать набор данных, особенно современной музыки, нужно решить огромное количество задач, связанных с юридическими тонкостями и авторскими правами.

В работе [59] авторы попытались решить проблему нехватки данных. Предложенная архитектура отличается тем, что может обучаться как с учителем, так и без него. Для того чтобы обучаться без учителя, была создана подсистема, извлекающая из произведения фонемы и частоту основного тона. Здесь фонемы — аналог произносимых слов, а частота основного тона — аналог нотной партитуры. Для обучения с учителем же как раз используются слова и нотные партитуры в том или ином представлении. При анализе предоставленных авторами примеров можно сделать вполне ожидаемые выводы, что обучение без учителя значительно уступает модели, обучаемой с учителем. Выходом здесь видится использование semi-supervised[19], которое объединяет два эти отдельные вида обучения. На рисунке 18 представлена архитектура такой модели.

Альтернативный подход к решению задачи SVS при малом количестве открытых качественных данных — методы аугментации. Основные работы в данном направлении начали появляться только с 2021 г. Рассмотрим некоторые подходы.

В статье [60] авторы пытаются решать задачу с помощью эффективного метода увеличения данных сегментацией длительности произведения — разбивают запись каждой песни на более мелкие фрагменты с тремя различными временными интервалами: (0–5),

[19] Обучение с частичным привлечением учителя (*англ.* Sesupervised learning) — разновидность машинного обучения с учителем, которое использует небольшое количество размеченных и большое количество неразмеченных данных.

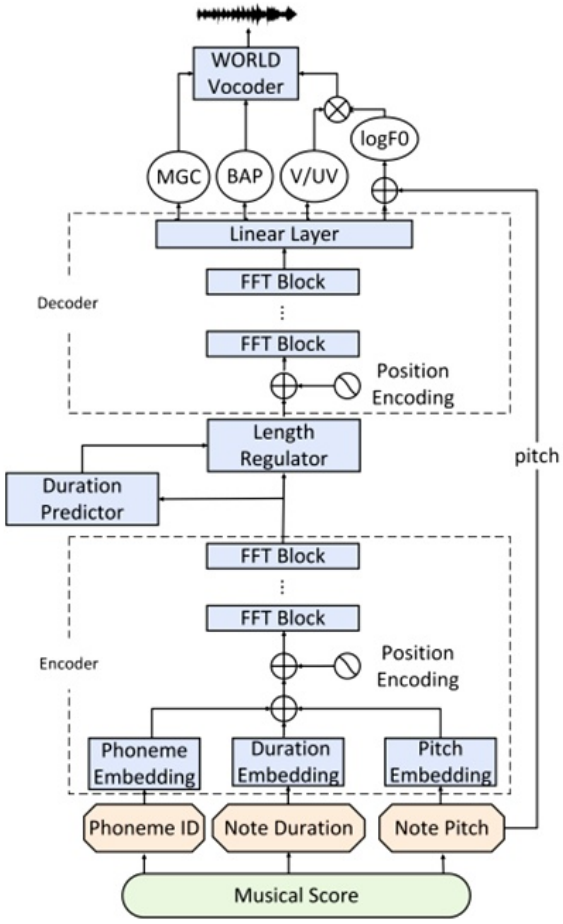

Рис. 17. End-to-End система SVS на основе FastSpeech архитектуры синтеза речи [57]

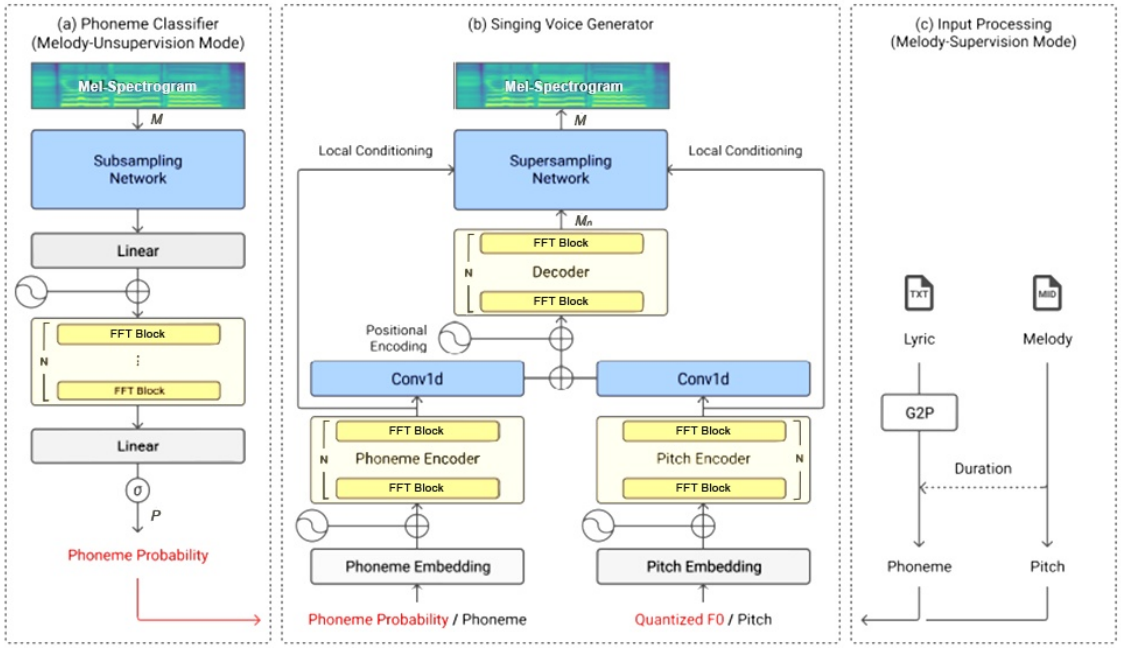

Рис. 18. Архитектура современной модели SVS [59]

(5–8) и (8–12) секунд. Плюсы данного метода — используются исключительно изначально заявленные базы данных, минусы — требуется отдельная модель, позволяющая правильно «сцепить» акустические признаки и лингвистические единицы.

В работе [61] предложено несколько методов аугментации. Авторы используют MixUp аугментацию (MA) — подход, хорошо показавший себя при аугментации изображений и текстов. При её использовании новый объект, добавляемый в обучение, является линейной комбинацией выбранных объектов с коэффициентами $\lambda, 1-\lambda, \lambda \in (0,1)$. На рисунке 19 представлена схема работы этого метода.

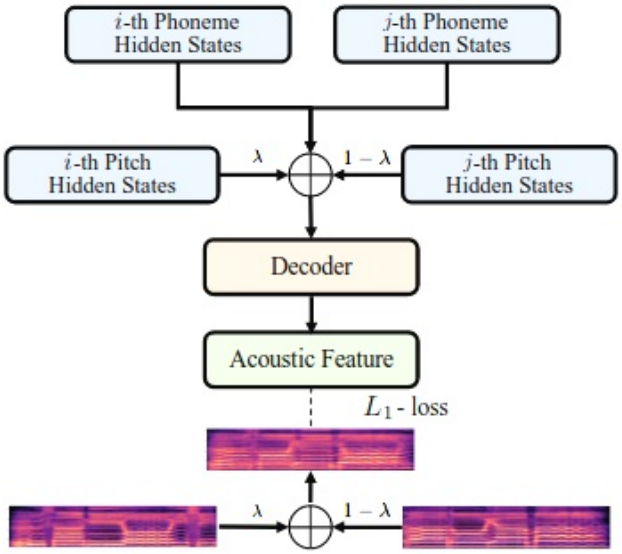

Рис. 19. MixUp аугментация модели SVS [61]

Однако метод MA авторов отличается от классического MA при обработке текстов, поскольку выполняется на расширенных скрытых состояниях фонем вместо входных последовательностей. Основная причина заключается в том, что в случае SVS звуковые фрагменты, выбранные для MA, могут иметь разную длительность для каждой ноты, что приводит к различным несостыковкам при регулировании длины.

Второй подход — Pitch Augmentation связан с изменением высоты тона. Для изменения высоты тона авторы используют вокодер WORLD [62], чтобы получить соответствующий певческий голос после настройки полутона, как показано на рисунке 20.

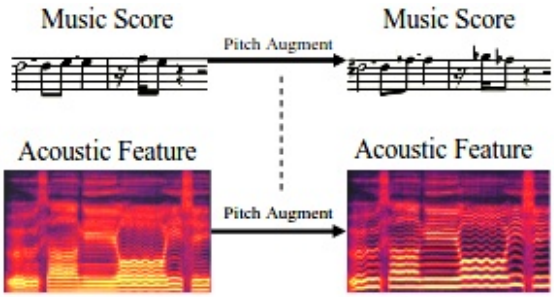

Рис. 20. Pitch аугментация модели SVS [61]

На вход вокодера поступает последовательность основного тона F_0, гармоническая и апериодическая спектральные огибающие спектра. Умножая или деля последовательность F_0 на $\sqrt[12]{2}$ и сохраняя спектральные огибающие неизменными, авторы получают сигнал поющего голоса, соответствующий его полутоновым изменениям.

Список литературы

1. *Yolk A., Wiering F., Kranenburg P.* Unfolding the potential of computational musicology // Proceedings of the 13th International Conference on Informatics and Semiotics in Organisations. July 4–6 2021. Leeuwarden, 2011. P. 137–144.
2. *Müller M.* “Fundamentals of Music Processing Audio, Analysis, Algorithms, Applications”. Berlin: Springer, 2015; doi:10.1007978-3-319-21945-5
3. *Бронфельд М. Ш.* Введение в музыкознание: Учеб. пособие для студ. высш. учеб. заведений. СПб.: Планета музыки, 2022, 308 с.
4. *Гладкий А. В., Мельчук И. А.* Элементы математической лингвистики. М.: Наука, 1969. 192 с.
5. *Пиотровский Р. Г., Бектаев К. Б., Пиотровская А. А.* Математическая лингвистика. Учеб. пособие для пед. институтов. М.: Высшая школа, 1977. 383 с.
6. *Тимофеева М. К.* Введение в математическую лингвистику: Практикум. Новосибирск, 2018.
7. *Большакова Е. И, Клышинский Э. С., Ландэ Д. В., Носков А. А., Пескова О. В., Ягунова Е. В.* Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие. М.: МИЭМ, 2011. 272 с.
8. *Ломакина Л. С., Суркова А. С.* Информационные технологии анализа и моделирования текстовых структур. Воронеж: Научная книга, 2015. 208 с.
9. *Balakrishnan A.* DeepPlaylist: Using Recurrent Neural Networks to Predict Song Similarity. Stanford: Stanford University, 2016.
10. *Rafii Z., Liutkus A., Stöter F. R., Mimilakis S. I., FitzGerald D., Pardo B.* An Overview of Lead and Accompaniment Separation in Music // arXiv preprint arXiv:1804.08300, 2018.
11. *Лайонс Р.* Цифровая обработка сигналов. М.: Бином, 2015, 656 с.
12. *Оппенгейм А. , Шафер Р.* Цифровая обработка сигналов. М.: Техносфера, 2012. 1048 с.
13. *Klapuri A., Davy M.* Processing Methods for Music Transcription. New York: Springer Science and Business Media, 2006.
14. *Taylor P.* Text-to-speech synthesis. Cambridge: Cambridge university press, 2009.
15. *Guennec D.* Study of Unit Selection Text-To-Speech Synthesis Algorithms. [PhD diss.], Université Rennes 1, Rennes, Brittany, France, 2017.
16. *Столбов М. Б.* Основы анализа и обработки речевых сигналов. СПб.: НИУ ИТМО, 2021, 101 с.
17. *Калиев А., Рыбин С. В.* Синтез речи: прошлое и настоящее // Компьютерные инструменты в образовании. 2019. № 1. С. 42–61. doi:10.32603/2071-2340-2019-1-47-55
18. *Miranda E. R, Biles J.* Evolutionary Computer Music. London: Springer Science and Business Media, 200. doi:10.1007/978-1-84628-600-1.
19. *Fingerhut M.* Music Information Retrieval, or how to search for (and maybe find) music and do away with incipits // Proc. of IAML-IASA Congress. August 8–13, 2004. Oslo, Norway, 2004. P. 17.
20. *Good M.* MusicXML: An Internet-Friendly Format for Sheet Music” // Proceedings of XML 2001. December 9-14, 2021. Boston, 2001. P. 03–04.
21. *Huber D.* The MIDI Manual: A Practical Guide to MIDI within Modern Music Production (Audio Engineering Society Presents). 4th edition. Oxfordshire: Routledge, 2020.
22. *Oppenheim I.* The ABC Music standard 2.0. 21 Feb. 2008. [Online]. URL: https://abc.sourceforge.net/standard/abc2-draft.html (date: 16.03.2023).
23. *Blum K.* OOoLilyPond: Creating musical snippets in LibreOffice documents. 2017. [Online]. URL: https://github.com/openlilylib/LO-ly (date: 16.03.2023).
24. *Hannah D., Saif M.* Generating Music from Literature” // Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL). Gothenburg, Sweden, 2014. P. 1–10. doi:10.3115/v1/W14-0901.

25. *Зубарева Н. Б., Куличкин П. А.* Тайны музыки и математическое моделирование: Алгебра или гармония?.. Гармония и алгебра! М., 2022. 254 с.
26. *Toro M., Rueda C., Agón C., Assayag G.* Gelisp: a framework to represent musical constraint satisfaction problems and search strategies // Journal of Theoretical and Applied Information Technology. 2016. Vol. 86, № 2. P. 327–331.
27. *Quick D., Hudak P.* Grammar-based automated music composition in Haskell // Proc. ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design, FARM '13. 2013. P. 59–70.
28. *Koops H. V., Magalhaes J. P., de Haas W. B.* A functional approach to automatic melody harmonisation // Proc. ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design, FARM '13. 2013. P. 47–58.
29. *Cunha, N. dos S., Subramanian A., Herremans D.* Generating guitar solos by integer programming // Journal of the Operational Research Society. 2018. Vol. 69, № 6. P. 971–985. doi:10.1080/01605682.2017.1390528
30. *Biles J. A.* GenJam: A Genetic Algorithm for Generating Jazz Solos // Proc. International Computer Music Conference (ICMC), 1994. P. 131–137.
31. *Fox C.* Genetic Hierarchical Music Structure // Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference. Melbourne Beach, Florida. May 11–13, 2006. Washington: AAAI Press, 2006. P. 243–247.
32. *Silas S. B.* Algorithmic Composition and Reductionist Analysis: Can a Machine Compose? Cambridge: Cambridge University New Music Society, 1997.
33. *Fernández J. D., Vico F.* AI Methods in Algorithmic Composition: A Comprehensive Survey // Journal of Artificial Intelligence Research. 2013. № 48. P. 513–582.
34. *Marchini M., Purwins H.* Unsupervised Analysis and Generation of Audio Percussion Sequences // Lecture Notes in Computer Science book series. Berlin: Springer, 2011. P. 205–218. doi:10.1007/978-3-642-23126-1-14
35. *Абросимов К. И., Суркова А .С.* Алгоритм генерации музыки на основе ABC-нотации и дистрибутивной семантики // Информационные системы и технологии (ИСТ-2021), сборник материалов XXVII Международной научно-технической конференции. Нижний Новгород: Нижегородский государственный технический университет им. Р. Е. Алексеева, 2021. С. 776–781.
36. *Hadjeres G., Pachet Fo., Nielsen F.* DeepBach: a Steerable Model for Bach Chorales Generation // Proceedings of the 34th International Conference on Machine Learning, PMLR, 2017. P. 1362–1371.
37. *Bach J.* 389 Chorales (Choral-Gesange). Los Angeles: Alfred Publishing Company, 1985.
38. *Johnson D. D.* Composing Music With Recurrent Neural Networks. [Online]. URL: https://www.danieldjohnson.com/2015/08/03/composing-music-with-recurrent-neural-networks (date: 16.03.2023).
39. *Zen H., Sak H.* Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis // Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015. P. 4470–4474.
40. *Benesty J., Sondhi M. M., Huang Y. (ed.)* Springer handbook of speech processing. Berlin: Springer, 2008.
41. *Hawthorne C., Elsen E., Song J., Roberts A., Simon I., Raffel C., Engel J., Oore S., Eck D.* Onsets and frames: Dual-objective piano transcription // Proceedings of the International Society for Music Information Retrieval Conference. Sep. 23–27, 2018. Paris, 2018. P. 50–57.
42. *Hiramatsu Y., Nakamura E., and Yoshii K.* Joint Estimation of Note Values and Voices for Audio-to-Score Piano Transcription // Proceedings of the 22nd International Society for Music Information Retrieval Conference ISMIR. Online, November 7-12, 2021. P. 278–284.
43. *Ozaki Y., McBride J., et. al.* Agreement Among Human and Automated Transcriptions of Global Songs // Proceedings of the 22nd International Society for Music Information Retrieval Conference ISMIR 2021. Online, November 7-12, 2021. P. 500–508. ISBN 978-1-7327299-0-2
44. *Fletcher N., Rossing T.* The Physics of Musical Instruments. New York: Springer-Verlag, 1998.
45. *Engel J. H., Resnick C., Roberts A., Dieleman S., Norouzi M., Eck D., Simonyan K.* Neural audio synthesis of musical notes with WaveNet autoencoders // Proceedings of the 34th International Conference on Machine Learning. 2017. Vol. 70. P. 1068–1077.
46. *Nistal J., Lattner S., and Richard G.* DarkGAN: Exploiting Knowledge Distillation for Comprehensible Audio Synthesis with GANs // Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021. Online, November 7–12, 2021. P. 482–494.

47. *Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.* Generative Adversarial Networks. arXiv preprint arXiv:1406.2661, 2014.
48. *Hayes B., Saitis C., Fazekas G.* Neural Waveshaping Synthesis // Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021. Online, November 7–12, 2021. P. 254–261.
49. *Wang A. L.-C.* An Industrial-Strength Audio Search Algorithm // Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003). October 27–30, 2003. Baltimore, 2003. P. 27–30. doi:10.5281/zenodo.1416340.
50. *Taylor P.* Text-to-Speech Synthesis. Cambridge: Cambridge University Press, 2009.
51. *Лобанов Б. М., Цирульник Л. И.* Компьютерный синтез и клонирование речи. Минск: Белорусская Наука, 2008. 316 с.
52. *Chandna P., Blaauw M., Bonada J., Gómez E.* WGANSing: A Multi-Voice Singing Voice Synthesizer Based on the Wasserstein-GAN // Proceedings of the 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, September 2–6, 2019. doi:10.23919/EUSIPCO.2019.8903099.
53. *Lee J., Choi H-S., Koo J. H., Lee K.* Disentangling Timbre and Singing Style with Multi-Singer Singing Synthesis System" // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2020). May 4–8, 2020. Barcelona, 2020. doi:10.1109/ICASSP40776.2020.9054636
54. *Liao C. F., Liu J. Y., Yang Y. H.* KaraSinger: Score-Free Singing Voice Synthesis with VQ-VAE Using Mel-Spectrograms // Proc. ICASSP. 2022, P. 956–960. doi10.1109/ICASSP43922.2022.9747441.
55. *Black A. W.* CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling // 9th Int. Conf. on Spoken Language Processing. September 17–21, 2006. Pittsburgh, 2006. doi:10.21437/Interspeech.2006-488
56. *Gu Y., Yin X., Rao Y. H., Wan Y., Tang B., Zhang Y., Chen J., Wang Y., Ma Z.* ByteSing: A Chinese Singing Voice Synthesis System Using Duration Allocated Encoder-Decoder Acoustic Models and WaveRNN Vocoders // Proceedings of the 12th International Symposium on Chinese Spoken Language Processing (ISCSLP'2021). January 24–27, 2021. Hong Kong, 2021. P. 1–5. doi: 10.1109/ISCSLP49672.2021.9362104.
57. *Lu L., Wu J., Luan J., Tan X., Zhou L.* XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System // Processing 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, October 25–29, 2020. P. 1306–1310. doi:10.21437/Interspeech.2020-1410.
58. *Ren Y., Ruan Y., Tan X., Qin T., Zhao S., Zhao Z., Liu T. Y.* FastSpeech: Fast, robust and controllable text to speech // NeurIPS. 2019. P. 1–13. doi:10.48550/arXiv.1905.09263
59. *Choi S., Nam J.* A Melody-Unsupervision Model for Singing Voice Synthesis // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2022), Singapore, May 23–27, 2022. P. 7242–7246. doi: 10.1109/ICASSP43922.2022.9747422
60. *Zhang Z., Zheng Y., Li X., Lu L.* WeSinger: Data-augmented Singing Voice Synthesis with Auxiliary Losses // arXiv preprint arXiv:2203.10750, 2022.
61. *Guo S., Shi J., Qian T., Watanabe S., Jin Q.* SingAug: Data Augmentation for Singing Voice Synthesis with Cycle-consistent Training Strategy // Proc. INTERSPEECH 2022. P. 4272–4276.
62. *Morise M., Yokomori F., Ozawa K.* WORLD: a vocoder-based high-quality speech synthesis system for real-time applications // IEICE TRANSACTIONS on Information and Systems. 2016. Vol. 99, № 7. P. 1877–1884.

Поступила в редакцию 19.12.2022, окончательный вариант — 16.03.2023.

Абросимов Кирилл Игоревич, студент магистратуры, факультет информационных технологий и программирования университета ИТМО, ✉ abrosimov.kirill.1999@mail.ru

Рыбин Сергей Витальевич, кандидат физико-математических наук, доцент, факультет информационных технологий и программирования, университет ИТМО; факультет компьютерных технологий и информатики СПбГЭТУ «ЛЭТИ», svrybin@itmo.ru

Computer tools in education, 2023
№ 1: 74–95
http://cte.eltech.ru
doi:10.32603/2071-2340-2023-1-74-95

Music Information Retrieval — Modern Challenges and Technology

Abrosimov K. I.[1], Student, ✉ abrosimov.kirill.1999@mail.ru, orcid.org/0000-0001-9262-0474
Rybin S. V.[1,2], Cand. Sc., Associate Professor, ITMO, ETU «LETI», svrybin@itmo.ru, orcid.org/0000-0002-9095-3168

[1]ITMO University, 49 Kronverksky, bldg. A, 197101, Russia Saint Petersburg, Russia
[2]Saint Petersburg Electrotechnical University,
5, building 3, st. Professora Popova, 197022, Saint Petersburg, Russia

Abstract

This paper discusses Music Information Retrieval — a field of computational musicology that is actively developing in the modern world. The paper describes some of the main tasks and technologies of this area, such as music generation, automatic music transcription, synthesis of musical instrument sounds, and music retrieval. Special attention is paid to one of the most interesting tasks at the junction of speech and music technologies —singing voice synthesis. Different approaches to this task, existing problems and methods of their solution are discussed.

Keywords: *computational musicology, music information retrieval, music generation, automatic music transcription, synthesis of musical instrument sounds, music retrieval, synthesis of the singing voice.*

Citation: K. I. Abrosimov and S. V. Rybin, "Music Information Retrieval — Modern Challenges and Technology," *Computer tools in education*, no. 1, pp. 74–95, 2023 (in Russian); doi:10.32603/2071-2340-2023-1-74-95

Список литературы

1. A. Yolk, F. Wiering, and P. Kranenburg, "Unfolding the potential of computational musicology," in *Proc. of the 13th Int. Conf. on Informatics and Semiotics in Organisations, Leeuwarden, The Netherlands, July 4–6, 2011*, pp. 137–144, 2011.
2. M. Müller, "Fundamentals of Music Processing. Audio, Analysis, Algorithms, Applications," Berlin: Springer, 2015; doi:10.1007978-3-319-21945-5
3. M. Sh Bronfeld, *The Introduction to Musicology: Textbook for students and teachers of higher musical educational institutions*, Saint Petersburg, Russia: Planeta Muziki, 2022 (in Russian).
4. A. V. Gladkiy and I. A Melchuk, *Elements of Mathematical Linguistics*, Moscow: Nauka, 1969 (in Russian).
5. R. G. Piotrovsky, K. B. Bektaev, and A. A Piotrovskaya, *Mathematical Linguistics. Textbook for pedagogical institutes*, Moscow: Vysshaia shkola, 1977 (in Russian).
6. M. K. Timofeeva, *Introduction to Mathematical Linguistics: Practicum*, Novosibirsk, Russia: Novosibirsk, 2018 (in Russian).
7. E. I. Bolshakova, E. S. Klyshinsky, D. V. Lande, Noskov A. A., Peskova O. V., and Yagunova E. V, *Automatic natural language processing and computational linguistics: textbook*, Moscow: HSE MIEM, 2011 (in Russian).
8. L. S. Lomakin and A. S. Surkov, *Information technologies for analysis and modeling of text structures*, Voronezh: Scientific Book Publ., 2015 (in Russian).
9. A. Balakrishnan, *DeepPlaylist : Using Recurrent Neural Networks to Predict Song Similarity*, Stanford, CA, USA: Stanford University, 2016.
10. Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An Overview of Lead and Accompaniment Separation in Music" in *arXiv*, [Online], preprint arXiv:1804.08300, 2018.

11. R. G. Lyons, “Understanding digital signal processing,” Moscow: Binom, 2015.
12. A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, 3-th ed., Moscow: Technosfera, 2012 (in Russian).
13. A. Klapuri and M. Davy, *Processing Methods for Music Transcription*, New York, USA: Springer Science and Business Media, 2006.
14. P. Taylor, *Text-to-speech synthesis*, Cambridge, England: Cambridge university press, 2009.
15. D. Guennec, *Study of Unit Selection Text-To-Speech Synthesis Algorithms*, [PhD diss.], Université Rennes 1, Rennes, Brittany, France, 2017.
16. M. B Stolbov, *Fundamentals of analysis and processing of speech signals*, [Textbook], St. Petersburg: ITMO University, 2021 (in Russian).
17. V. Rybin and A. Kaliev, “Speech Synthesis: Past and Present,” *Computer Tools in Education*, no. 1, pp. 5–28, 2019 (in Russian); doi:10.32603/2071-2340-2019-1-5-28
18. E. R Miranda and J. Biles, *Evolutionary Computer Music*, London: Springer Science and Business Media, 2007; doi:10.1007/978-1-84628-600-1
19. M. Fingerhut, “Music Information Retrieval, or how to search for (and maybe find) music and do away with incipits,” in *Proc. of IAML-IASA Congress, Oslo, Norway, August 8–13, 2004*, p. 17, 2004.
20. M. Good, “MusicXML: An Internet-Friendly Format for Sheet Music,” in *Proc. of XML 2001, Boston, USA, December 9-14*, pp. 03–04, 2001.
21. D. Huber, *The MIDI Manual: A Practical Guide to MIDI within Modern Music Production (Audio Engineering Society Presents)*, 4th ed., Oxfordshire, England: Routledge, 2020.
22. I. Oppenheim, “The ABC Music standard 2.0,” in *abc.sourceforge.net*, 21 Feb. 2008, [Online]. Available: https://abc.sourceforge.net/standard/abc2-draft.html
23. K. Blum, “OOoLilyPond: Creating musical snippets in LibreOffice documents,” in *github.com* 2017. [Online]. Available: https://github.com/openlilylib/LO-ly
24. D. Hannah and M. Saif, “Generating Music from Literature,” in *Proc. of the 3rd Workshop on Computational Linguistics for Literature (CLFL), Gothenburg, Sweden, 2014*, pp. 1–10, 2017; doi:10.3115/v1/W14-0901
25. N. B. Zubareva and P. A Kulichkin, *Secrets of music and mathematical modeling: Algebra or harmony?.. Harmony and algebra!*, Moscow: URSS, 2022 (in Russian).
26. M. Toro, C. Rueda, C. Agón, and G. Assayag, “Gelisp: a framework to represent musical constraint satisfaction problems and search strategies,” *J. of Theoretical and Applied Information Technology*, vol. 86, no. 2, pp. 327–331, 2016.
27. D. Quick and P. Hudak, “Grammar-based automated music composition in Haskell,” in *Proc. ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design, FARM’13*, pp. 59–70, 2013.
28. H. V. Koops, J. P. Magalhaes, and W. B de Haas, “A functional approach to automatic melody harmonisation,” in *Proc. ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design, FARM’13*, pp 47–58, 2013.
29. N. dos S. Cunha, A. Subramanian, and D. Herremans, “Generating guitar solos by integer programming,” *Journal of the Operational Research Society*, vol. 69, no. 6, pp. 971–985, 2017; doi:10.1080/01605682.2017.1390528
30. J. A Biles, “GenJam: A Genetic Algorithm for Generating Jazz Solos,” in *Proc. International Computer Music Conference (ICMC), 1994*, pp. 131–137, 1994.
31. C. Fox, “Genetic Hierarchical Music Structure,” in *Proc. of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11–13, 2006*, Washington, DC, USA: AAAI Press, pp 243–247, 2006.
32. S. B Silas, *Algorithmic Composition and Reductionist Analysis: Can a Machine Compose?*, Cambridge, England: Cambridge University New Music Society, 1997.
33. J. D. Fernández and F. Vico, “AI Methods in Algorithmic Composition: A Comprehensive Survey,” *J. of Artificial Intelligence Research*, no. 48, pp. 513–582, 2013.
34. M. Marchini and H. Purwins, “Unsupervised Analysis and Generation of Audio Percussion Sequences,” in *Lecture Notes in Computer Science book series*, Berlin: Springer, pp. 205–218, 2011; doi:10.1007/978-3-642-23126-1-14
35. K. I. Abrosimov and A. S. Surkova, “Music generation algorithm based on abc notation and distributive semantics,” in *Proc. of the XXVII Int. Scientific and Technical ConferenceInformation systems and technologies (IST-2021), Nizhny Novgorod state technical university n.a. R. E. Alekseev*, pp. 776—781, 2021 (in Russian).
36. G. Hadjeres, Fo. Pachet, and F. Nielsen, “DeepBach: a Steerable Model for Bach Chorales Generation,” in *Proc. of the 34th Int. Conf. on Machine Learning, PMLR*, pp. 1362–1371, 2017.
37. J. Bach, *389 Chorales (Choral-Gesange)*, Los Angeles, CA: Alfred Publishing Company, 1985.
38. D. D. Johnson, “Composing Music With Recurrent Neural Networks,” in *www.danieldjohnson.com*. [Online]. Available: https://www.danieldjohnson.com/2015/08/03/composing-music-with-recurrent-neural-networks
39. H. Zen and H. Sak, “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis,” in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 4470–4474, 2015.
40. J. Benesty, M. M. Sondhi, and Y. Huang, eds, *Springer handbook of speech processing*, Berlin: Springer, 2008.
41. C. Hawthorne et al., “Onsets and frames: Dual-objective piano transcription,” in *Proc. of the Int. Society for Music Information Retrieval Conference, Paris, France, Sep. 23–27, 2018*, pp. 50–57, 2018.

42. Y. Hiramatsu, E. Nakamura, and K. Yoshii, “Joint Estimation of Note Values and Voices for Audio-to-Score Piano Transcription,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7–12*, pp. 278–284, 2021.
43. Y. Ozaki, J. McBride, et al., “Agreement Among Human and Automated Transcriptions of Global Songs,” in *Proc. of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7–12*, pp. 500–508, 2021; doi:10.31234/osf.io/jsa4u
44. N. Fletcher and T. Rossing, *The Physics of Musical Instruments*, New York: Springer-Verlag, 1998.
45. J. H. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in *Proc. of the 34th Int. Conf. on Machine Learning*, vol. 70, pp. 1068–1077, 2017.
46. J. Nistal, S. Lattner, and Richard G, “DarkGAN: Exploiting Knowledge Distillation for Comprehensible Audio Synthesis with GANs,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf., ISMIR 2021, Online, November 7–12*, pp. 482–494, 2021.
47. I. J. Goodfellow et al., “Generative Adversarial Networks,” in *arXiv*, [Online], preprint arXiv:1406.2661, 2014.
48. B. Hayes, C. Saitis, and G. Fazekas, “Neural Waveshaping Synthesis,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf., ISMIR 2021, Online, November 7–12*, pp. 254–261, 2021; doi:10.48550/arXiv.2107.05050
49. A. L.-C. Wang, “An Industrial-Strength Audio Search Algorithm,” in *Proc. of the 4th Int. Conf. on Music Information Retrieval (ISMIR 2003), Baltimore, USA, October*, pp. 27–30, 2003; doi:10.5281/zenodo.1416340
50. P. Taylor, *Text-to-Speech Synthesis,* Cambridge, England: Cambridge University Press, 2009.
51. B. M. Lobanov and L. I. Tsirul’nik, *Computer synthesis and speech cloning*, Minsk, Belarusia: Belorusskaya Nauka, 2008 (in Russian).
52. P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, “WGANSing: A Multi-Voice Singing Voice Synthesizer Based on the Wasserstein-GAN,” *Proc. of the 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, September 2–6*, 2019; doi:10.23919/EUSIPCO.2019.8903099
53. J. Lee, H-S. Choi, J. H. Koo, and K. Lee, “Disentangling Timbre and Singing Style with Multi-Singer Singing Synthesis System,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP’2020), Barcelona, Spain, May 4–8*, 2020; doi:10.1109/ICASSP40776.2020.9054636
54. C. F. Liao, J. Y. Liu, and Y. H. Yang, “KaraSinger: Score-Free Singing Voice Synthesis with VQ-VAE Using Mel-Spectrograms,” in *Proc. ICASSP.*, pp. 956–960, 2022; doi:10.1109/ICASSP43922.2022.9747441
55. A. W. Black, “CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling,” in *9th Int. Conf. on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21*, 2006; doi:10.21437/Interspeech.2006-488
56. Y. Gu et al., “ByteSing: A Chinese Singing Voice Synthesis System Using Duration Allocated Encoder-Decoder Acoustic Models and WaveRNN Vocoders,” in *Proc. of the 12th Int. Symposium on Chinese Spoken Language Processing (ISCSLP’2021), Hong Kong, Jan. 24–27*, pp. 1–5, 2021; doi:10.1109/ISCSLP49672.2021.9362104
57. L. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, “XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System,” in *Proc. 21st Annual Conf. of the Int. Speech Communication Association, Virtual Event, Shanghai, China, Oct. 25–29*, pp. 1306–1310, 2020; doi:10.21437/Interspeech.2020-1410
58. Y. Ren et al., “FastSpeech: Fast, robust and controllable text to speech,” in *NeurIPS*, pp. 1–13, 2019; doi:10.48550/arXiv.1905.09263
59. S. Choi and J. Nam, “A Melody-Unsupervision Model for Singing Voice Synthesis,” *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP’2022), Singapore, May 23–27*, pp. 7242–7246. doi:10.1109/ICASSP43922.2022.9747422
60. Z. Zhang, Y. Zheng, X. Li, and L. Lu, “WeSinger: Data-augmented Singing Voice Synthesis with Auxiliary Losses,” in *arXiv*, [Online], preprint arXiv:2203.10750, 2022.
61. S. Guo, J. Shi, T. Qian, S. Watanabe, and Q. Jin, “SingAug: Data Augmentation for Singing Voice Synthesis with Cycle-consistent Training Strategy,” in *Proc. INTERSPEECH 2022*, pp. 4272–4276, 2022; doi:10.21437/Interspeech.2022-
62. M. Morise, F. Yokomori, and K. Ozawa, “WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016; doi:10.1587/transinf.2015EDP7457

Received 19-12-2022, the final version — 16-03-2023.

Kirill Abrosimov, Master’s Degree student, IT and Programming Department, ITMO University, ✉ **abrosimov.kirill.1999@mail.ru**

Sergey Rybin, Candidate of Sciences in Physics and Mathematics, Associate Professor, IT and Programming Department, ITMO University; Faculty of Computer Science and Technology, Saint Petersburg Electrotechnical University, svrybin@itmo.ru