

неиерархические плоские алгоритмы и иерархические древовидные. На выходе иерархических древовидных алгоритмов на выходе получается дерево кластеров с некоторой степенью вложенности. На выходе же плоских неиерархических алгоритмов получается некоторое множество кластеров. *Плоские неиерархические алгоритмы* в свою очередь подразделяются на четкие и нечеткие. В четких алгоритмах кластеризуемый объект может принадлежать исключительно одному кластеру, в нечетких алгоритмах вычисляется вероятность принадлежности объекта к полученным кластерам. *Иерархические алгоритмы* кластеризации подразделяются на агломеративные и дивизивные. В агломеративных алгоритмах на начальной итерации каждому объекту ставится в соответствие кластер, которые объединяются в более крупные кластеры при последующих итерациях алгоритма. Дивизивные алгоритмы представляют из себя противоположность агломеративным: на начальной итерации все объекты состоят в одном кластере, а при последующих итерациях кластеры делятся на вложенные меньшего размера. Графическая визуализация результатов работы иерархических алгоритмов может быть представлена в виде дендрограммы, которая позволяет отобразить взаимные связи объектов для исходного множества.

Список использованных источников:

1. Мандель И. Д. Кластерный анализ. М.: Финансы и статистика, 1988. – 176с
2. Sammut C., Webb J. Encyclopedia of Machine Learning – NY, USA, 2010.

## СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ АНАЛИЗА МУЗЫКАЛЬНОГО РИТМА

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Шабловский К. С.*

*Одинец Д. Н. – к. т. н., доцент*

В настоящее время наблюдается бурный рост числа пользователей сети, которые отдают предпочтение музыкальным сервисам, ориентированным на поиск музыки и онлайн радиовещание, как альтернативу домашним коллекциям. За последнее десятилетие наметилась устойчивая тенденция к росту объемов медиа-трафика в интернете, поэтому научные исследования в области классификации музыки и автоматической генерации метаданных являются весьма актуальными.

Задача анализа музыкального ритма сводится к нахождению последовательности сильных долей(ударов) и слабых долей(интервалов) между ними, с последующим выделением ритмических паттернов на их основе. Ритмическая структура произведения может быть однозначно задана при помощи последовательности обнаруженных ритмических паттернов [1].

В настоящей работе был проведено сравнение точности следующих алгоритмов анализа темпа:

- алгоритм Мигеля Алонсо [2];
- алгоритм Саймона Диксона [3];
- алгоритм Анси Клапури [4];
- алгоритм Эрика Шейрера [5];
- алгоритм Джоджа Тзанетакиса [6];
- алгоритм Кристиана Угле [7];

В результате экспериментального анализа выявлено, что алгоритм А.Клапури является наиболее точным из приведенных. Его точность для не зашумленных аудиозаписей составляет около 80% и незначительно зависит от величины темпа и жанра музыки [8].

Алгоритм Клапури включает следующие этапы:

1. частотно-временной анализ;
2. разбиение сигнала с использованием набора полосовых фильтров [5];
3. определение акцентов, тактов и ритмических паттернов с помощью вероятностного алгоритма для каждого из частотных интервалов;
4. построение фазовой модели на основе шагов 2 и 3;
5. генерация ритмической картины, на основе данных, полученных на шагах 3 и 4;

В результате анализа экспериментальных данных сделано заключение о том, что ключевыми преимуществами алгоритма Клапури, позволяющими добиться высокой точности, являются:

- применение полосовых фильтров, и последующий анализ ритмической картины для различных частотных диапазонов;
- комбинация вероятностного и детерминированного алгоритмов при получении конечного результата;

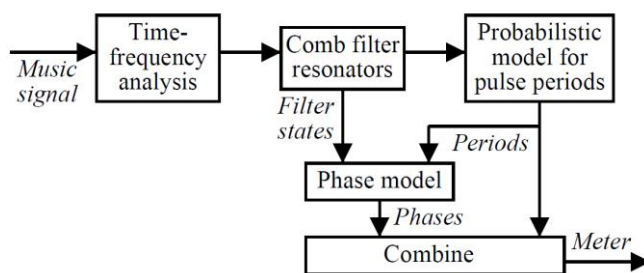


Рис. 1 – Схема построения ритмической картины по алгоритму А.Клапури

Список использованных источников:

1. F. Gouyon, S. Dixon, "A review of automatic rhythm description systems," *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.
2. M. Alonso, B. David, and G. Richard, "Tempo and beat estimation of musical signals," in Proc. International Conference on Music Information Retrieval. Barcelona: Audiovisual Institute, Pompeu Fabra University, 2004, pp. 158–163.
3. S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.
4. E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Am.*, vol. 103, no. 1, pp. 588–601, 1998.
5. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Speech and Audio Processing*, 2005.
6. G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
7. C. Uhle, J. Rohden, M. Cremer, and J. Herre, "Low complexity musical meter estimation from polyphonic music," in Proc. AES 25th International Conference. New York: Audio Engineering Society, 2004, pp. 63–68.
8. F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, P. Cano, "An experimental comparison of audio tempo induction algorithms", *IEEE Transactions on Speech and Audio Processing*, 14(5), 2006.

## ФАКТОРИЗАЦИЯ ЧИСЛА ПОСРЕДСТВОМ МОДИФИЦИРОВАННОГО АЛГОРИТМА ШОРА

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Кухарчук И. В.

Самаль Д. И. – к-т техн. наук, доцент

Процедура факторизации широко используется в современной криптографии, однако, на сегодняшний день нет эффективных высоких нижних оценок сложности этих алгоритмов. Вопрос о существовании алгоритма факторизации с полиномиальной сложностью на классическом компьютере является одной из важных открытых проблем современной теории чисел. В то же время факторизация с полиномиальной сложностью возможна с использованием квантовых компьютеров, однако существуют определённые технические сложности построения систем с достаточным количеством кубитов для обеспечения приемлемой точности вычислений.

Для решения поставленной задачи нет необходимости в реализации всего алгоритма факторизации при помощи квантовых вычислений, так как это приведёт к несоизмеримому увеличению сложности алгоритма в сравнении с приобретённой скоростью вычисления. Поэтому в нашем случае весь алгоритм разбит на две части: подготовительная часть выполняется на классическом компьютере, а ядро – с учётом специфики квантовых вычислений.

Суть алгоритма факторизации Шора заключается в сведении задачи факторизации к задаче поиска периода функции, так как в случае, если известен период функции, то факторизация осуществляется при помощи алгоритма Евклида за полиномиальное время на классическом компьютере.

Необходимо учитывать то обстоятельство, что алгоритм нахождения периода функции вероятностный, а это в свою очередь требует ввода дополнительных операций по верификации полученного периода на соответствие требованиям достаточности. В нашем случае этими требованиями является чётность полученного периода и проверка на корректность решения путём произведения полученных множителей и сравнение результата с начальным значением. В случае если найденное значение не удовлетворяет требованию, существует необходимость повторного запуска алгоритма поиска периода.

Функция, для которой производится поиск периода, как правило, имеет следующий вид:

$$f(x) = a^x \pmod{M}$$