

СЖАТИЕ АУДИОФАЙЛОВ БЕЗ ПОТЕРЬ С ПОМОЩЬЮ КОМБИНАЦИИ МЕТОДОВ КОМПРЕССИИ БАЙТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Ануфриев Д.И., Максименков В.А.

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

Калугина М.А. – канд. физ.-мат. наук, доцент

В докладе приведены результаты исследования возможности дополнительного уменьшения объема аудиофайлов, упакованных разными программными аудиокодеками, с помощью алгоритмов сжатия байтовых последовательностей без потерь (алгоритмы LZ77 и Хаффмана). Оценка эффективности разработанного алгоритма производится с помощью сравнения значений коэффициента сжатия файла.

Современные технологии хранения и передачи аудиоданных предъявляют высокие требования к эффективности методов сжатия, направленных на минимизацию объема информации при сохранении исходного качества. Особую значимость в данном контексте приобретают алгоритмы сжатия без потерь (lossless compression), обеспечивающие возможность полного восстановления первичных данных после процедуры декодирования.

Актуальность исследования обусловлена существенными объемами, занимаемыми аудиофайлами высокого качества, особенно в несжатых форматах (WAV) или при использовании специализированных lossless-кодеков (FLAC). Хотя современные аудиокодеки демонстрируют значительную степень компрессии [1], они не всегда в полной мере учитывают структурные особенности музыкальных композиций. В частности, большинство музыкальных произведений содержат повторяющиеся фрагменты (например, припевные секции или повторяющиеся мелодические паттерны), которые потенциально могут быть подвергнуты более эффективному сжатию. Поэтому представляется перспективным направление исследования возможности применения специализированных алгоритмов сжатия, учитывающих циклическую природу музыкальных данных. Подобные алгоритмы, оптимизированные для выявления и компактного представления повторяющихся последовательностей, могут обеспечить дополнительное уменьшение объема хранимой информации без ухудшения качества звучания. Данный подход, кроме того, открывает новые возможности для оптимизации процессов хранения и передачи аудиоконтента в профессиональной и потребительской сферах.

Сжатие без потерь происходит за счет представления данных в более коротком формате. Это позволяет сократить количество битов, необходимых для хранения или передачи данных, гарантируя при этом, что исходные данные могут быть полностью восстановлены [2]. Поэтому целью исследования стал сравнительный анализ результатов использования классических методов сжатия информации (алгоритмы LZ77 и Хаффмана) над аудиофайлами разных форматов (.wav, .mp3, .flac и тому подобные).

Алгоритм LZ77 [3] назван по инициалам своих авторов и году публикации, что заложило традицию именования словарных методов сжатия (алгоритмы LZ-группы). Его работа основана на использовании скользящего окна фиксированного размера (N последних символов) в качестве динамического словаря. Каждый закодированный блок включает три параметра: смещение до начала совпадающей подстроки в словаре, длину этой подстроки и первый символ входного потока, следующий за совпадением [4].

Кодирование Хаффмана представляет собой базовый метод генерации кодов переменной длины, обеспечивающих минимальную среднюю длину кодовой последовательности. Данный алгоритм получил широкое распространение и лежит в основе множества приложений для компрессии текстовых и графических данных. В одних случаях он применяется непосредственно, в других – интегрируется как этап в комбинированные схемы сжатия, сочетающие несколько алгоритмов [5].

В рамках проведенного исследования была разработана программная реализация алгоритмов LZ77 и Хаффмана на языке C++, обеспечивающая пошаговое сжатие данных. На первом этапе данные обрабатываются алгоритмом LZ77, который заменяет повторяющиеся последовательности ссылками на предыдущие вхождения. Результат передается на вход алгоритму Хаффмана, строящему оптимальные префиксные коды для дальнейшего сжатия.

Для эксперимента были отобраны аудиозаписи в форматах WAV (несжатый), MP3, FLAC и Opus (сжатые с потерями и без). Каждый файл последовательно обрабатывался связкой LZ77 + Хаффман, после чего вычислялся коэффициент сжатия (отношение размера файла после сжатия к исходному). Тестирование проводилось на файлах с разной структурой: музыкальными композициями, записями с тишиной и короткими звуковыми фрагментами.

Ниже представлена таблица с результатами работы программы (таблица 1), последовательно сжимающей аудиофайлы алгоритмами LZ77 и Хаффмана.

Таблица 1 – Результат работы алгоритмов с разными видами данных

Аудиофайл	Изначальный размер файла, Кбайт	Размер файла после сжатия алгоритмом LZ77, Кбайт	Размер файла после сжатия алгоритмом Хаффмана, Кбайт	Коэффициент сжатия (размер после сжатия / размер до сжатия)
Файлы в формате mp3				
Audio1.mp3	6119.296	6094.407	6080.980	0.993738
Audio2.mp3	9674.624	9537.532	9526.253	0.984663
Audio3.mp3	7012.224	6878.023	6810.527	0.971236
Audio4.mp3	2713.610	2628.650	2620.357	0.965635
Audio5.mp3	8183.680	8180.181	8162.737	0.997441
Audio6.mp3	1815.931	1540.833	1538.546	0.847249
Audio7.mp3	8.507	4.458	4.841	0.569884
Audio8.mp3	364.004	236.184	235.641	0.647358
Audio9.mp3	261.234	134.656	133.759	0.512028
Файлы в формате opus				
Audio1.opus	3177.346	3177.233	3177.747	1.000126
Audio2.opus	5168.365	5164.476	5164.990	0.999346
Audio3.opus	3118.924	3083.450	3083.927	0.988779
Audio4.opus	4068.916	4068.818	4069.332	1.000102
Файлы в формате wav				
Audio1.wav	26975.486	26777.578	24336.842	0.902183
Audio2.wav	41886.828	41263.110	40419.607	0.964971
Audio3.wav	30618.392	30451.876	30155.543	0.984883
Audio4.wav	36075.356	36065.183	34851.260	0.966068
Файлы в формате flac				
Audio2.flac	29339.997	29337.100	29337.614	0.999918
Audio3.flac	17134.946	17126.685	17127.199	0.999547
Audio4.flac	25966.801	25958.112	25967.315	1.000019
Audio10.flac	21501.049	21492.102	21491.229	0.999543

На основе проведенного эксперимента были сделаны следующие выводы: на файлах большого объема эффективность данного подхода достаточно мала, однако для аудиофайлов, размер которых соизмерим с размером смещения и скользящего окна в алгоритме LZ77 (порядка 300 кБ), данный подход показал хорошие результаты (сжатие примерно в 2 раза). Это означает, что аудиокодек mp3, который в среднем сжимает в 10 раз, может быть улучшен до сжатия примерно в 20 раз. Выяснено, что с файлами большого объема проблема заключается в следующем: при увеличении размера окна до размера, соизмеримого с размером среднестатистической песни (мелодии), время работы данного алгоритма на доступном на сегодня аппаратном обеспечении превышает время проигрывания мелодии. Следовательно, не получится проигрывать данный файл в формате реального времени.

Проведённое исследование показало, что дополнительное сжатие аудиофайлов алгоритмами LZ77 и Хаффмана эффективно только для несжатых форматов (WAV) и небольших MP3-файлов с локальными повторами (тишина, монотонные участки) благодаря способности LZ77 работать с короткими паттернами при ограниченном размере окна (1023 байта). Однако стоит заметить, что в условиях неограниченного времени выполнения сжатия данный подход имеет место быть, так как он обеспечивает весомую дополнительную компрессию уже сжатого аудиофайла.

В качестве дополнительного исследования представляется перспективным разработать модификацию алгоритма LZ77, которая будет делать обратные ссылки не только на полностью совпадающие фрагменты последовательности байт, но и на части, незначительно отличающиеся друг от друга. Данный подход может обеспечить более эффективное сжатие аудиофайла, так как зачастую два коротких фрагмента мелодии, которые человек считает одинаковыми, могут немного различаться в закодированном виде.

Список использованных источников:

- Muin, F. A. A review of lossless audio compression standards and algorithms / F. A. Muin, T. S. Gunawan, M. Kartiwi, M. A. Elsheikh // AIP Conference Proceedings, Vladivostok, 18–22 Sep. 2017. – AIP Publishing, 2017. – Т. 1883, № 1.
- Unit 4 Lab 4: Data Representation and Compression, Page 6 [Electronic resource] // Beauty and Joy of Computing. – Mode of access: <https://bjc.edc.org/bjc-r/cur/programming/4-internet/4-representation-compression/6-compression.html>. – Date of access: 25.03.2025.
- Ziv, J. A Universal Algorithms for Sequential Data Compression / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – 1977. – Т. 23, № 3. – С. 337–343.

61-я научная конференция аспирантов, магистрантов и студентов БГУИР

4. Семенюк, В. В. Экономное кодирование дискретной информации : учеб. пособие / В. В. Семенюк. – СПб. : СПбГИТМО (ТУ), 2001. – 115 с.
5. Сэломон, Д. Сжатие данных, изображения и звука : учеб. пособие / Д. Сэломон. – Москва : Техносфера, 2004. – 368 с.