МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

СЖАТИЕ С ПОТЕРЯМИ НА ПРИМЕРЕ КОМПРЕССИИ АУДИОФАЙЛОВ

АВТОРЕФЕРАТ ДИПЛОМНОЙ РАБОТЫ

Студентки 4 курса 451 группы
специальности 09.03.04 — Программная инженерия
факультета КНиИТ
Вьюрковой Светланы Игоревны

Научный руководитель
к. ф.-м. н.,доцент ______________ Г. Г. Наркайтис

Заведующий кафедрой
д. ф.-м. н., доцент ______________ С. В. Миронов

Саратов 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Популярность Интернета значительно выросла за последние несколько лет, он стал основным средством обмена файлами. Услуги с высокой пропускной способностью доступны сегодня всем, но факт остается фактом: время - самой дорогой ресурс, который есть у человека. Экономия времени и места для хранения означает более низкие затраты. Это проложила путь для требований улучшения сжатия данных.

Сжатие данных подразделяется на две основные категории: без потерь и с потерями. Сжатие без потерь создает точную копию оригинала после распаковки, в то время как его аналог с потерями - нет. Типичным примером сжатия без потерь является ZIP формат. Эта форма сжатия данных эффективна для ряда файлов. Сжатие изображения и аудио в этом формате не так эффективно, поскольку информация в этих типах данных менее избыточна. Аудио компакт-диски используют популярный формат WAV. Формат WAV несжатый и содержит звуковой файл без потерь качества. В связи с этим он имеет множество недостатков. Размер файла WAV зависит от его частоты дискретизации. 8-битный моно WAV с частотой дискретизации 22050 Гц (герц) займет 22050 байтов за секунду. 16-битный стерео WAV с частотой дискретизации 44,1 кГц (килогерц) занимает 176 400 байт за секунду (44 100 байт в секунду * 2 байта * 2 канала). Одна минута аудио с качеством компакт-диска занимает примерно 10 МБ. Эти огромные требования к хранению данных способствовали популярности MP3 как одного из наиболее широко используемых стандартов в мире сжатие звука. В данной работе будет оценена эффективность сжатия аудио-информации различными алгоритмами и проанализировано влияние потери точности на качественное восприятие звукового сигнала. Мы сопоставим результаты компрессии с потерями на основе быстрого преобразования Фурье и вейвлет-преобразования.

Цель:

Оценить эффективность сжатия с потерями на примере компрессии звуковых файлов с помощью различных алгоритмов и проследить, как потеря точности влияет на качественное восприятие звукового сигнала.

Задачи:

1. реализовать спектральное разложение сигнала с помощью быстрого преобразования Фурье;

2. изучить и реализовать вейвлет-преобразование;
3. разработать и реализовать адаптивную квантизацию спектральных коэффициентов;
4. реализовать алгоритм Хаффмана для сжатия данных без потерь;
5. сравнить эффективность и качество компрессии для БПФ и разных вариантов вейвлетов;
6. сравнить эффективность и качество компрессии для разной битности квантизации спектральных коэффициентов.

Данная тема является актуальной в связи с тем, что количество информации растёт, и необходимо повышать эффективность хранения и передачи информации, допускающей потерю качества.

1 Общие сведения о работе

Реализуем сжатие и распаковку звукового wav-файла, с применением алгоритмов быстрого преобразования Фурье, вейвлет-преобразования и Хаффмана. В общем процесс сжатия имеет следующий вид:

1. Вычитка информации из файла/запись восстановленных данных в новый файл;
   а) БПФ/ОБПФ;
   б) Вейвлет-преобразование/обратное вейвлет-преобразование;
2. Нормализация/денормализация коэффициентов;
3. Сжатие алгоритмом Хаффмана/распаковка алгоритмом Хаффмана;

1.1 Структура wav-файла

Для начала необходимо изучить, из чего состоит wav-файл, в его структуре данные начинаются с 44 байта, а до этого хранится заголовок. В программе заголовок представлен следующей структурой.

Проводим вычитку файла и получаем необходимые данные. Запись восстановленных коэффициентов заключается в том, чтобы занести в новый wav-файл заголовок изначального, но уже в область данных заполнить новыми числами, полученными после ОБПФ и обратного вейвлет-преобразования.

1.2 Быстрое преобразование Фурье

Быстрое преобразование Фурье является одним из важнейших алгоритмов для обработки сигналов и их анализа. БПФ позволяет перевести сигнал из рассмотрения во временной области в частотную, т.е. раскладывает сигнал на синусоиду и косинусоиду. Представление функции в частотной области называют спектром функции. На выходе имеется сигнал с такой же размерностью, что и у начального [1, 2].

Прямое БПФ:

$$X_n = \sum_{k=0}^{N-1} x_k W_N^{kn},$$

где $W_N^{kn} = e^{-i\frac{2\pi}{N}kn}$ – дискретные экспоненциальные функции, $n = 1, 2, \ldots, N-1$.

БПФ выполняется при помощи дополнительной функции $Butterfly$, которая позволяет производить обработку непрерывного потока комплексных отсчетов с АЦП, частота дискретизации которого в 2 раза выше тактовой частоты обработки. То есть «бабочка» БПФ за один такт производит вычисление сразу для двух комплексных отсчетов.

Разбиение N-точечного ДПФ на два N/2-точечных.

Пусть $x(k)$ – действительная последовательность длиной в N отсчётов и пусть $x(k) \Leftrightarrow X(n)$, $n \in N$. Разобьём последовательность $x(k)$ на пару $\frac{N}{2}$-точечных последовательности $x_1(k)$ и $x_2(k)$, где $x_1(k) = x(2k)$ – последовательность для чётных отсчётов, $x_2(k) = x(2k+1)$ – последовательность для чётных отсчётов. Пусть $x_1(k) \Leftrightarrow X_1(n)$ и $x_2(k) \Leftrightarrow X_2(n)$ – $\frac{N}{2}$-точечные ДПФ этих подпоследовательностей. Установим связь $X(n)$ с $X_1(n_1)_{N/2}$ и $X_2(n_2)_{N/2}$. Для первых $\frac{N}{2}$ коэффициентов ДПФ можно записать:

$$X(n) = \sum_{k=0}^{N-1} x(k) W_N^{-nk} = \sum_{k=0}^{N/2-1} x(2k) W_N^{-2nk} + \sum_{k=0}^{N/2-1} x(2k+1) W_N^{-n(2k+1)} =$$
$$= \sum_{k=0}^{N/2-1} x_1(k) W_{N/2}^{-nk} + W_N^{-n} \sum_{k=0}^{N-1} x_2(k) W_{N/2}^{-nk} = X_1(n)_{N/2} + W_N^{-n} X_2(n)_{N/2}.$$

Здесь учтено, что $W_N^{-2nk} = W_{N/2}^{-nk}$.Для $n = N/2, \ldots, N-1$ с учётом свойство симметрии будет иметь: $X(n) = X^*(N-n)$.

1.3 Вейвлеты

Вейвлеты - это целое семейство математических функций особого вида, которые позволяют анализировать частотные компоненты данных, и получаются при помощи сжатия (растяжения) и сдвигов по временной оси исходного вейвлета. Их график имеет вид волнообразных колебаний с амплитудой, за что они и заслужили такое название [3, 4]. Общий вид вейвлет-функции:

$$\psi_{ab}(t) = \frac{1}{\sqrt{a}} \phi \left(\frac{t-b}{a} \right),$$

где $\phi(t)$ – исходный(материнский) вейвлет, $\frac{1}{\sqrt{a}}$ – множитель, обеспечивающий нормализацию, t – время, b – параметр, характеризующий сдвиг по времени, a – параметр масштаба, $\psi(t) \in L$.

Большинство семейств дискретных вейвлетов строится на основе нескольких аксиом кратно-масштабного анализа. Существует три фундаментальных свойства:

- Ортогональность
- Компактность носителя
- Симметричность формы

В общем случае вейвлеты обладают из них только двумя. Семейств, которые обладают всеми тремя, не существует.

Существует множество видов вейвлетов: грубые, бесконечные регулярные, ортогональные с компактным носителем, биортогональные с компактным носителем, комплексные. Каждый вид имеет свои преимущества и недостатки. В данной работе будут рассматриваться ортогональные вейвлеты с компактным носителем, а именно вейвлеты Добеши, Симлета и Койфлета. Чтобы не высчитывать их коэффициенты самостоятельно, можно найти их на просторах интернета. В программе они реализованы как статические массивы.

Преобразование Фурье хорошо для стационарных сигналов, потому что оно даёт полную информацию о спектральных характеристиках сигнала в исследуемой частотной области и усредняет значение сигнала во времени, в то время как вейвлет-преобразование исследует изменения характеристик во времени и хорошо для нестационарного сигнала.

Свойства вейвлет-преобразования:

- Ограниченность. Квадрат нормы функции должен быть конечным.
- Локализация. Вейвлет-преобразование в отличие от преобразования Фурье использует локализованную исходную функцию и во времени, и по частоте.
- Нулевое среднее. График исходной функции должен осциллировать (быть знакопеременным) вокруг нуля на оси времени и иметь нулевую площадь.

  Равенство нулю площади функции $\psi(t)$, т.е. нулевого момента, приводит к тому, что фурье-преобразование $S_\psi(\omega)$ этой функции равно нулю при $\omega = 0$ и имеет вид полосового фильтра. При различных значениях a это будет набор полосовых фильтров.

  Часто для приложений бывает необходимо, чтобы не только нулевой, но и все первые n моментов были равны нулю.
- Автомодельность. Характерным признаком вейвлет-преобразования является его самоподобие. Все вейвлеты конкретного семейства $\psi_{ab}(t)$ име-

ют то же число осцилляций, что и материнский вейвлет $\psi(t)$, поскольку получены из него посредством масштабных преобразований (a) и сдвига (b).

За счет изменения масштаба (увеличение а приводит к сужению фурье-спектра функции), вейвлеты способны выявлять различие в характеристиках на разных шкалах (частотах), а за счет сдвига- проанализировать свойства сигнала в разных точках на всем исследуемом интервале. Поэтому, при анализе нестационарных сигналов, за счет свойства локальности вейвлетов, получают существенное преимущество перед преобразованием Фурье, которое дает только глобальные сведения о частотах (масштабах) анализируемого сигнала, так как используемая при этом система функций (комплексная экспонента или синусы и косинусы) определена на бесконечном интервале.

Осуществляться данное преобразование будет при помощи алгоритма Малла(другими словами, алгоритма быстрого преобразования). Одновременно сигнал раскладывается с помощью высокочастотного и низкочастотного фильтров. В результате получаются детализирующие коэффициенты (после ВЧ-фильтра) и коэффициенты аппроксимации (после НЧ-фильтра).

Так как половина частотного диапазона сигнала была отфильтрована, то, согласно теореме Котельникова, отсчёты сигналов можно проредить в 2 раза.

метод одномерного дискретного вейвлет-преобразования (ДВП) N-го порядка последовательности x_n определяется следующими рекуррентными соотношениями:

$$a_n^{(i)} = \sum_{k=0}^{N-1} g_k a_{2n-k}^{i-1} i = 1, 2, \ldots, J$$

$$d_n^{(i)} = \sum_{k=0}^{N-1} h_k a_{2n-k}^{i-1} a_n^{(0)} \equiv x_n,$$

где $a_n^{(i)}$ и $d_n^{(i)}$ являются аппроксимирующими и детализирующими коэффициентами i-го уровня, а g_k и $h_k (k = 0, 1, \ldots, N-1)$ – коэффициенты низкочастотного и высокочастотного анализирующих фильтров соответственно.

С другой стороны, сигнал x_n может быть восстановлен по коэффициентам $a_n^{(J)}, d_n^{(J)}, d_n^{(J-1)}, \ldots, d_n^{(1)}$ путём последовательной итерации по формулам:

$$a_n^{*(i-1)} = \begin{cases} \sum_{k=0}^{N/2-1} g'_{2k} a^{*(i)}_{\frac{m}{2}-k} + \sum_{k=0}^{N/2-1} h'_{2k} d^{*(i)}_{\frac{m}{2}-k},\ m \text{ – чётное} \\ \sum_{k=0}^{N/2-1} g'_{2k+1} a^{*(i)}_{\frac{m-1}{2}-k} + \sum_{k=0}^{N/2-1} h'_{2k+1} d^{*(i)}_{\frac{m-1}{2}-k},\ m \text{ – нечётное} \end{cases}$$

где g'_k и h'_k являются коэффициентами низкочастотного и высокочастотного синтезирующих фильтров, соответственно.

Для того, чтобы восстановленный сигнал соответствовал исходному, должны быть соответствующим образом подобраны анализирующий (раскладывающий) и синтезирующий (собирающий) фильтры.

Для вейвлет-преобразования функции $f(x)$ необходимо вычислить серию коэффициентов $a_n, d_n, d_{n-1}, \ldots, d1$, где a_n - аппроксимация функции, d_i - детализирующие коэффициенты функции, $i = 1, \ldots, n$. Каждый коэффициент находится интегрированием (11, 12):

$$a_{J-N,k} = (f, \varphi_{J-N,k}) = \int_R f(x)\varphi_{J-N,k}(x)dx;$$

$$d_{J-m,k} = (f, \psi_{J-m,k}) = \int_R f(x)\psi_{J-m,k}(x)dx, m = 1, 2, \ldots, N.$$

Возникает проблема вычисления большого количества интегралов с необходимой точностью. Следует также учитывать, что при высоком уровне разрешения J носители функций $\varphi_{J,k}()$ и $\psi J, k()$ становятся малыми порядка $\frac{1}{2^J}$.

1.3.1 Алгоритм Малла

Быстрое вейвлет-преобразование, предложенное Малла позволяет решить эту проблему. Алгоритм Малла даёт возможность вычислять коэффициенты вейвлет-разложения без интегрирования, используя алгебраические операции на основе свёртки:

$$a_n^{(i)} = \sum_{k=0}^{N-1} g_k a_{2n-k}^{i-1} i = 1, 2, \ldots, J$$

$$d_n^{(i)} = \sum_{k=0}^{N-1} h_k a_{2n-k}^{i-1} a_n^{(0)} \equiv x_n,$$

где $a_n^{(i)}$ и $d_n^{(i)}$ являются аппроксимирующими и детализирующими коэффициентами i-го уровня, а g_k и $h_k (k = 0, 1, \ldots, N - 1)$ – коэффициенты низкочастотного и высокочастотного анализирующих фильтров соответственно; N – порядок фильтра.

Эти равенства обеспечивают быстрые алгоритмы вычисления вейвлет-коэффициентов (каскадные алгоритмы, алгоритмы Малла). Термин «быстрые» означает не только, что в (13) используются более быстрые алгебраические процедуры, но и то, что при каждом преобразовании общее число новых коэффициентов не увеличивается в два раза, а остаётся прежним.

Разложение можно повторять несколько раз для дальнейшего увеличения частотного разрешения с дальнейшим прореживанием коэффициентов после НЧ и ВЧ-фильтрации. Это можно представить в виде двоичного дерева, где листья и узлы соответствуют пространствам с различной частотно-временной локализацией.

Единственное отличие фильтрации в алгоритме Малла от классического КИХ-фильтра, задаваемого уравнением $y(k) = \sum_{i=0}^{m} b_i x(k - i)$, заключается в том, что значения фильтруемого ряда выбираются через один - индекс $2n - k$ в a_{2n-k}^{i-1}. Это и есть децимация 2↓ – исключение из обработки каждого второго элемента.

1.4 Нормализация и денормализация коэффициентов

Затем, полученный из БПФ сигнал нормализуем, т.е. из полученного диапазона значений $[K1, K2]$ сначала получаем значения вида $[0, 1]$, а затем приводим к виду $[0, n]$, где n - 2^b, а b – битность, которую мы хотим получить. Значения действительной и мнимой частей вычисляются по формуле: $x = (y - min)/(max - min) * bits$, где x – нормализованное значение, y – исходное значение, min – минимальное значение из диапазона $[K1, K2]$, max – максимальное значение из диапазона $[K1, K2]$, $bits = 2^b$. Денормализация является обратной операцией для описанной ранее.

1.5 Алгоритм Хаффмана

Алгоритм Хаффмана является одним из первых эффективных алгоритмов для кодирования. Он был предложен в 1952 году. Основная идея алгоритма основывается на том, что обладая данными о частоте повторения каждого из символов, встречающегося в тексте, можно для каждого символа построить код переменной длины, который будет состоять из целого количества битов. Символ с большей вероятностью появления будет ставится в соответствии код меньшей длины. Под символами в работе подразумеваются некие повторяющиеся элементы строки, такие как буквы, цифры, знаки препинания, пробелы и т.д., а под кодом будем понимать последовательный набор битов [5].

Рассмотрим каждый из шагов реализации алгоритма Хаффмана на примере словосочетания "по шоссе Саша шел с саше". Для начала нужно совершить предподготовку данного словосочетания путем разбиения его на символы, а затем посчитать, сколько раз каждый символ встречается в тексте. На этом основании составим таблицу с частотами:

Затем на основании этой таблицы строится дерево Хаффмана по следующему алгоритму:

1. Каждый символ входного алфавита является свободным узлом, то есть листом, который имеет свой вес равный количеству вхождений символа в текст
2. Выбираются два узла с наименьшим весом
3. Создаётся третий узел, вес которого равен сумме весов взятых узлов. Узлы, которые уже образовали своё дерево, удаляются из списка свободных узлов, а их родитель добавляется в этот список
4. Дуга до одного листа будет соответствовать биту 0, а до другой биту 1
5. Вернуться к шагу 2, если количество свободных узлов не равно одному

Массив с нормализованными значениями сжимаем алгоритмом Хаффмана. В результате данные записываются в файл с разрешением DHF.

Функция $Main$ для сжатия принимает на вход файл, в котором содержатся нормализованные значения, полученные в преобразовании Фурье. Они хранятся в виде последовательных чисел сначала всех действительных частей, а затем мнимых. Для реализации алгоритма сжатия необходимо создать из входных данных дерево Хаффмана, закодировать все значения, а затем записать полученное в файл output.dhf. Полный листинг для алгоритма Хаффмана приведён во вложениях.

В результате было получено 6 разных выходных файлов.

Алгоритм Хаффмана является одним из первых эффективных алгоритмов для кодирования. Он был предложен в 1952 году. Основная идея алгоритма основывается на том, что обладая данными о частоте повторения каждого из символов, встречающегося в тексте, можно для каждого символа построить код переменной длины, который будет состоять из целого количества битов. Символ с большей вероятностью появления будет ставится в соответствии код меньшей длины. Под символами в работе подразумеваются некие повторяющиеся элементы строки, такие как буквы, цифры, знаки препинания, пробелы и т.д., а под кодом будем понимать последовательный набор битов .

Рассмотрим каждый из шагов реализации алгоритма Хаффмана на примере словосочетания "по шоссе Саша шел с саше". Для начала нужно совершить предподготовку данного словосочетания путем разбиения его на символы, а затем посчитать, сколько раз каждый символ встречается в тексте. На этом основании составим таблицу с частотами:

Затем на основании этой таблицы строится дерево Хаффмана по следующему алгоритму:

1. Каждый символ входного алфавита является свободным узлом, то есть листом, который имеет свой вес равный количеству вхождений символа в текст
2. Выбираются два узла с наименьшим весом
3. Создаётся третий узел, вес которого равен сумме весов взятых узлов. Узлы, которые уже образовали своё дерево, удаляются из списка свободных узлов, а их родитель добавляется в этот список
4. Дуга до одного листа будет соответствовать биту 0, а до другой биту 1
5. Вернуться к шагу 2, если количество свободных узлов не равно одному

Функция $Main$ описывает процесс, обратный для сжатия алгоритмом Хаффмана. Тут по считанным значениям восстанавливается дерево, по которому идет восстановление исходных данных. Весь листинг для алгоритма Хаффмана описан в приложении.

2 Результаты сравнительного анализа

Исходный файл имел размер 1025 КБ. Проведя по 7 тестовых запусков для каждого вида вейвлетов и БПФ с разными значениями b – битности.

Для оценки полученных результатов необходимо было просмотреть полученные коэффициенты сжатия, полученные при разной нормализации и разных алгоритмах, а также оценить потерю информации, т.е. замерить относительную погрешность потерь и провести сравнение полученных результатов.

Выводы:

- программой с применением БПФ удается добиться более лучшего сжатия, чем с вейвлет-преобразованием;
- преобразование вейвлетами имеет меньшие потери, чем преобразование Фурье;
- все виды ортогональных вейвлетов имеют примерно одинаковые коэффициенты сжатия и потери;
- наиболее подходящей является нормализация значений к 10-12–битным.

ЗАКЛЮЧЕНИЕ

Целью данной работы было оценить эффективность сжатия аудио-информации различными алгоритмами и проанализировать влияние потери точности на качественное восприятие звукового сигнала. Поставленные цели были достигнуты и все задачи выполнены. На основании полученных результатов можно сделать вывод, что наиболее эффективным алгоритмом для компрессии аудиофайлов является подход, основанный на БПФ и сжатии алгоритмом Хафмана. В ходе работы была оценена зависимость коэффициента сжатия и процента потерь от битности квантизации нормализованных значений спектра. Наилучшими показателями обладают программы с приведением к 10-12–битным значениям. Для них коэффициенты сжатия оказались самыми большими при наименьших потерях. Полученный результат, т.е. восстановленный сигнал можно оценить, включив его в любом музыкальном проигрывателе. Теория вейвлет-преобразований требует дальнейшего изучения для улучшения результатов работы программы.

Выполнены следующие задачи:

1. реализовано спектральное разложение сигнала с помощью быстрого преобразования Фурье;
2. изучино и реализовано вейвлет-преобразование;
3. разработана и реализована адаптивную квантизацию спектральных коэффициентов;
4. реализован алгоритм Хаффмана для сжатия данных без потерь;
5. проведено сравнение эффективности и качества компрессии для БПФ и разных вариантов вейвлетов;
6. проведено сравнение эффективности и качества компрессии для разной битности квантизации спектральных коэффициентов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Нуссбаумер, Г.* Быстрое преобразование Фурье и алгоритмы вычисления сверток / Г. Нуссбаумер. — Москва: Радио и связь, 1985.

2 Принцип построения алгоритмов быстрого преобразования Фурье [Электронный ресурс]. — URL: https://ru.dsplib.org/content/fft_introduction/fft_introduction.html (Дата обращения 15.05.2021). Загл. с экр. Яз. рус.

3 Сжатие аудиоданных с применением вейвлет-преобразований [Электронный ресурс]. — URL: https://bstudy.net/858151/tehnika/szhatie_audiodannyh_primeneniem_veyvlet_preobrazovaniy (Дата обращения 10.05.2021). Загл. с экр. Яз. рус.

4 Разработка и исследование алгоритма сжатия голосовых данных с использованием вейвлет-преобразований [Электронный ресурс]. — URL: http://masters.donntu.org/2005/kita/stoyanovsky/diss.htm (Дата обращения 10.05.2021). Загл. с экр. Яз. рус.

5 Сжатие информации с потерями [Электронный ресурс]. — URL: https://spravochnick.ru/informatika/kodirovanie_informacii/szhatie_informacii_s_poteryami/ (Дата обращения 10.05.2021). Загл. с экр. Яз. рус.