**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики**

Кафедра: «Компьютерных технологий»

Магистерская работа.

Адаптивное арифметическое кодирование коэффициентов вейвлетного разложения

Студент: Кирильчук В.Е.

Научный руководитель: Кудряшов Б.Д.

Санкт-Петербург

-2012-

Оглавление

[ВВЕДЕНИЕ 8](#_Toc326652276)

[Постановка задачи 9](#_Toc326652277)

[Подходы к сжатию аналоговой информации 10](#_Toc326652278)

[1. Измерение информации 10](#_Toc326652279)

[2. Сжатие данных 11](#_Toc326652280)

[3. Линейное предсказание 13](#_Toc326652281)

[4. Вейвлетное преобразование 15](#_Toc326652282)

[5. Арифметическое кодирование 18](#_Toc326652283)

[Обзор существующих алгоритмов сжатия изображений 21](#_Toc326652284)

[1. JPEG 21](#_Toc326652285)

[2. JPEG2000 24](#_Toc326652286)

[3. JPEG-LS 28](#_Toc326652287)

[4. WebP 30](#_Toc326652288)

[Разработка и обоснование алгоритма сжатия 32](#_Toc326652289)

[1. Введение 32](#_Toc326652290)

[2. Общая схема разработанного алгоритма 33](#_Toc326652291)

[3. Вейвлетное преобразование изображения 35](#_Toc326652292)

[4. Линейное предсказание 38](#_Toc326652293)

[5. Контекстно-зависимое арифметическое кодирование 47](#_Toc326652294)

[Практическая реализация 50](#_Toc326652295)

[1. Выбранные инструменты 50](#_Toc326652296)

[2. Общая структура классов и модулей 51](#_Toc326652297)

[3. Вейвлетное преобразование 53](#_Toc326652298)

[4. Линейное предсказание 54](#_Toc326652299)

[5. Кодирование (ввод\вывод) 55](#_Toc326652300)

[6. Разработанный алгоритм кодирования изображения 56](#_Toc326652301)

[7. Интерфейс пользователя 57](#_Toc326652302)

[Результаты работы 58](#_Toc326652303)

[Заключение 61](#_Toc326652304)

[Список литературы 62](#_Toc326652305)

# ВВЕДЕНИЕ

Цифровые изображения занимают всё большую часть информационного мира. Развитие интернета, наряду с нарастающей доступностью мобильных устройств, имеющих встроенные цифровые камеры и полноценный доступ к интернету, приводит к тому, что количество цифровых изображений хранящихся и передающихся по каналам связи в единицу времени постоянно увеличивается. Отсюда постоянный интерес к улучшению алгоритмов сжатия данных, представляющих изображения.

Сжатие данных важно как для скорости передачи, так и эффективности хранения. Кроме того, решение проблемы сжатия изображений, или в более общем смысле, кодирования изображений стимулировало развитие многих областей техники и математики.

В настоящее время существует большое количество различных алгоритмов сжатия изображений. Одним из наиболее известных алгоритмов сжатия изображений является алгоритм JPEG, использующий в своей основе дискретное косинусное преобразование. В 2000 году экспертной группой, создавшей JPEG, был разработан новый алгоритм с названием JPEG2000, который обладал массой преимуществ по сравнению с JPEG. Основой нового стандарта сжатия изображений стало дискретное вейвлетное преобразование и, хотя, новый алгоритм так окончательно и не заместил собой предыдущий, он породил большое количество исследований, связанных с использованием вейвлетного преобразования в качестве основы для алгоритма сжатия изображений.

Ещё одной особенностью стандарта JPEG2000 в сравнении с JPEG стало использование более эффективного энтропийного кодирования, а именно – адаптивного контекстно-зависимого арифметического кодирования.

В то же время, достаточно эффективными оказались алгоритмы сжатия изображений, основанные на так называемом методе линейных предсказаний. К таким алгоритмам относятся, например JPEG-LS и WebP.

Однако очень малое количество исследовательских работ посвящено анализу возможностей такого алгоритма, который использовал бы одновременно все три подхода: метод линейных предсказаний, вейвлетное преобразование и арифметическое кодирование. Именно этому и посвящена данная работа.

# Постановка задачи

Задачей работы является разработка и анализ алгоритма сжатия цифровых изображений, который сочетает в себе использование метода линейных предсказаний, вейвлетное преобразование, а также арифметическое кодирование. При этом под адаптивностью арифметического кодирования, фигурирующего в названии работы, подразумевается не только адаптивная модель кодера, но и некоторая адаптация алгоритма в зависимости от входного изображения.

Стоит отметить, что, так как работа носит чисто исследовательский характер, под разработкой следует понимать не написание готового промышленного алгоритма, а лишь тестовую реализацию с целью провести анализ возможностей предложенного в работе алгоритма.

# Подходы к сжатию аналоговой информации

## Измерение информации

Информационная энтропия — мера неопределённости или непредсказуемости информации, неопределённость появления какого-либо символа первичного алфавита. При отсутствии информационных потерь энтропия численно равна среднему количеству информации на символ передаваемого сообщения. Т.е. энтропия - это среднее количество информации, приходящейся на одно элементарное сообщение источника.

Информационная двоичная энтропия для независимых случайных событий с возможными состояниями рассчитывается по формуле:

, где p – функция вероятности

Энтропия даёт теоретическую оценку максимально возможного сжатия без потерь.

Если символы алфавита зависимы, то используют понятие условной энтропии.

называется средней условной энтропией источника B при известном сообщении из множества A.

Для условной энтропии справедливо следующее неравенство:

обращается в равенство, только если сообщения X и Y статистически независимы.

## Сжатие данных

Сжатие данных подразделяют на сжатие с потерями качества и сжатие без потерь.

Сжатие данных с потерями — метод сжатия (компрессии) данных, при использовании которого распакованные данные отличаются от исходных.

Сжатие без потерь — метод сжатия данных: видео, аудио, графики, документов представленных в цифровом виде, при использовании которого закодированные данные могут быть восстановлены с точностью до бита.

Сжатие данных основано на устранении избыточности, содержащейся в исходных данных. Простейшим примером избыточности является повторение в тексте фрагментов (например, слов естественного или машинного языка). Подобная избыточность обычно устраняется заменой повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины. Другой вид избыточности связан с тем, что некоторые значения в сжимаемых данных встречаются чаще других. Сокращение объёма данных достигается за счёт замены часто встречающихся данных короткими кодовыми словами, а редких — длинными (энтропийное кодирование). Сжатие данных, не обладающих свойством избыточности (например, случайный сигнал или белый шум), принципиально невозможно без потерь.

В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности. Иными словами, для сжатия данных используются некоторые априорные сведения о том, какого рода данные сжимаются. Не обладая такими сведениями об источнике, невозможно сделать никаких предположений о преобразовании, которое позволило бы уменьшить объём сообщения. Модель избыточности может быть статической, неизменной для всего сжимаемого сообщения, либо варьироваться на этапе сжатия (и восстановления). Методы, позволяющие на основе входных данных изменять модель избыточности информации, называются адаптивными. Неадаптивными являются обычно узкоспециализированные алгоритмы, применяемые для работы с данными, обладающими хорошо определёнными и неизменными характеристиками. Подавляющая часть достаточно универсальных алгоритмов являются в той или иной мере адаптивными.

Коэффициент сжатия — основная характеристика алгоритма сжатия. Она определяется как отношение объёма исходных несжатых данных к объёму сжатых, то есть:

,

где k – коэффициент сжатия, - объём исходных данных, - объём сжатых данных.

## Линейное предсказание

Формально, линейное предсказание – это вычислительная процедура, позволяющая по некоторому набору предшествующих отсчётов цифрового сигнала предсказать текущий отсчёт.

Пусть  — анализируемый цифровой сигнал. При линейном предсказании оценка текущего отсчёта сигнала   формируется как линейная комбинация предшествующих отсчётов:

Задача линейного предсказания заключается в том, чтобы найти параметры модели прогноза - набор коэффициентов , при котором средний квадрат ошибки предсказания минимален.

Величина  называется порядком линейного предсказания.

Для изображений можно рассмотреть следующую модель:

|  |  |
| --- | --- |
|  |  |
|  |  |

Пусть - это значение яркости изображения в точке с соответствующими координатами. Тогда по значениям яркости непосредственных соседей данной точки (лежащих в окрестности точки с координатами ) можно сформировать оценку .

Например, для третьего порядка линейного предсказания:

Разность между оценкой и действительным значением, которую можно ожидать относительно небольшой (по абсолютной величине), далее квантуется, кодируется и передается совместно с параметрами модели прогноза. Декодер восстанавливает яркость в положении по значениям оценки, вычисленной на основании уже декодированных отсчётов, параметров модели и переданной разности.

Для подавляющего большинства изображений энтропия разности  обычно существенно меньше энтропии исходного сигнала . Это делает возможным сжатие данных.

## Вейвлетное преобразование

Слово вейвлет (английское "wavelet" - "волночка") означает "маленькая волна". Вейвлеты - это обобщённое название семейств математических функций определенной формы, в которых все функции получаются из одной базовой (порождающей) функции (материнского вейвлета) посредством ее сдвигов и растяжений по оси времени. Вейвлет представляет собой оконную функцию, которая осциллирует и тем самым напоминает волну.

Вейвлеты позволяют анализировать различные частотные компоненты данных и при этом обладают существенными преимуществами по сравнению с преобразованием Фурье, потому что вейвлет-преобразование позволяет судить не только о частотном спектре сигнала, но также о том, в какой момент времени появилась та или иная гармоника. С их помощью можно легко анализировать прерывистые сигналы, либо сигналы с острыми всплесками. Кроме того вейвлеты позволяют анализировать данные согласно масштабу, на одном из заданных уровней (мелком или крупном). Они нашли широкое применение в цифровой обработке изображений, цифровых сигналов и анализе данных. Обычно вейвлет преобразования подразделяют на два класса: непрерывные и дискретные.

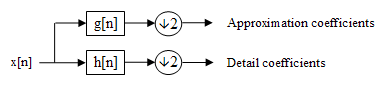
Дискретные вейвлет-преобразования относятся к вейвлет-преобразованиям, в которых вейвлеты представлены дискретными сигналами. Может быть показано, что дискретное вейвлетное преобразование эквивалентно фильтрации коэффициентов сигнала фильтрами с конечным импульсным откликом. Сначала сигнал пропускается через низкочастотный (low-pass) фильтр с импульсным откликом g:

где g это импульсный отклик фильтра

Одновременно сигнал раскладывается с помощью высокочастотного (high-pass) фильтра с импульсным откликом h. В результате получаются детализирующие коэффициенты (после ВЧ-фильтра) и коэффициенты аппроксимации (после НЧ-фильтра). Эти два фильтра связаны между собой и называются квадратурными зеркальными фильтрами (QMF).

Так как половина частотного диапазона сигнала была отфильтрована, то, согласно теореме Котельникова (теорема Найквиста — Шеннона), отсчёты сигналов можно проредить в 2 раза:

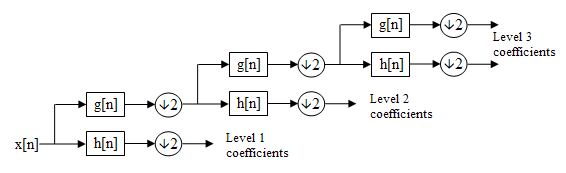
Такое разложение вдвое уменьшило разрешение по времени в силу прореживания сигнала. Однако каждый из получившихся сигналов представляет половину частотной полосы исходного сигнала, так что частотное разрешение удвоилось.

[](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Wavelets_-_DWT.png)

С помощью оператора прореживания  вышеупомянутые суммы можно записать короче:

Вычисление полной свёртки  с последующим прореживанием — это излишняя трата вычислительных ресурсов. Схема лифтинга является оптимизацией, основанной на чередовании этих двух вычислений.

Это разложение можно повторить 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.

[](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Wavelets_-_Filter_Bank.png)

При сжатии изображений используется двумерное дискретное вейвлетное преобразование. Если изображение представлено двумерной матрицей, содержащей значения отсчётов изображения, то двумерное дискретное вейвлетное преобразование является поочерёдным применением преобразования к строкам и столбцам матрицы.

Само вейвлет-преобразование не сжимает данные, но позволяет таким образом преобразовать входное изображение, что без заметного ухудшения качества изображения можно сократить его избыточность.

## Арифметическое кодирование

Арифметическое кодирование — один из алгоритмов энтропийного сжатия. Он обеспечивает почти оптимальную степень сжатия с точки зрения энтропийной оценки кодирования Шеннона. На каждый символ требуется почти бит, где — информационная энтропия источника. В отличие от алгоритма Хаффмана, метод арифметического кодирования показывает высокую эффективность для дробных неравномерных интервалов распределения вероятностей кодируемых символов. Однако в случае равновероятного распределения символов, данный метод приближается к префиксному коду Хаффмана и даже может занимать на один бит больше.

Метод арифметического кодирования основан на идее преобразования входного потока в число с плавающей точкой из интервала . Каждый поступающий на обработку символ уменьшает этот интервал пропорционально вероятности своего появления. По мере поступления очередного символа интервал уменьшается, а число бит, представляющие этот интервал - увеличивается. Заметим, что изначально длина интервала равна 1. Известно, что сумма всех вероятностей появления символов алфавита также равна 1. Пусть - количество символов алфавита, - вероятность появления в тексте символа с порядковым номером . Пусть вероятности появления символов известны. Тогда символу с номером ставится в соответствие интервал:

Если первым на вход поступает символ с порядковым номером в алфавите, то вместо исходного берётся k-ый интервал. Он снова делится на частей в тех же пропорциях, что и исходный. Аналогично для второго символа и так далее.

Затем для построенного интервала находится число, принадлежащее его внутренней части и равное целому числу, деленному на минимально возможную положительную целую степень двойки. Это число и будет кодом для рассматриваемой последовательности. Из него можно однозначно восстановить исходную последовательность символов. Т.к. все возможные конкретные коды — это числа из интервала [0, 1), то из соображений экономии можно отбрасывать лидирующий ноль и десятичную точку. Нужен еще один специальный код-маркер, сигнализирующий о конце сообщения. Когда декодер встречает этот символ, он прекращает свой процесс.

Если вероятности появления символов в тексте изначально не известны, то можно воспользоваться их оценкой, которая может меняться в процессе кодирования. Такой метод арифметического кодирования называется адаптивным, т.е. при каждом сопоставлении символу кода, алгоритм изменяет внутренний ход вычислений так, что в следующий раз этому же символу может быть сопоставлен другой код, т.е. происходит “адаптация” алгоритма к поступающим для кодирования символам. При декодировании происходит аналогичный процесс. Необходимость применения адаптивного алгоритма возникает в том случае, если вероятностные оценки для символов сообщения неизвестны до начала работы алгоритма.

Построение адаптивного арифметического кода для последовательности символов из заданного множества можно реализовать следующим алгоритмом. Каждому символу сопоставляется его вес, вначале вес для всех равен единице. Все символы располагаются в естественном порядке, например по возрастанию. Вероятность каждого символа устанавливается равной его весу, деленному на суммарный вес всех символов. После получения очередного символа и постройки интервала для него, вес этого символа увеличивается на один.

В отличие от префиксного кодирования, арифметическое кодирование позволяет генерировать коды как целой, так и нецелой длины. Являясь теоретически оптимальным методом, арифметическое кодирование превосходит в эффективности префиксное кодирование. Оно также опережает префиксное кодирование в скорости построения системы кодов, однако из-за повышенной сложности нередко заметно уступает в скорости самого кодирования (имеется в виду процесс генерации кодовой последовательности).

Несмотря на некоторые существенные преимущества арифметического кодирования, данный метод до последнего времени был недостаточно распространен на практике. На сегодняшний день в большинстве коммерческих приложений для построения системы кодов переменной длины арифметическое кодирование практически вытеснило кодирование Хаффмана. Арифметическое кодирование применяется в тех случаях, когда требуется добиться максимально возможного качества информационного представления и когда скорость работы не имеет решающего значения.

# Обзор существующих алгоритмов сжатия изображений

## JPEG

JPEG - один из популярных графических форматов, применяемый для хранения фотоизображений и подобных им изображений. Алгоритм JPEG в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета. Наибольшее распространение JPEG получил в цифровой фотографии и для хранения и передачи изображений с использованием сети Интернет.

При сжатии изображение преобразуется из цветового пространства RGB в YUV. После этого преобразования для каналов изображения U и V, отвечающих за цвет, может выполняться "прореживание", которое заключается в том, что каждому блоку из 4 пикселей (2х2) яркостного канала Y ставятся в соответствие усреднённые значения блоков из U и V. Если к качеству восстановленного после сжатия изображения предъявляются повышенные требования, прореживание может выполняться в каком-то одном направлении — по вертикали или по горизонтали, или не выполняться вовсе. Допускаются и некоторые другие типы прореживания.

Далее яркостной компонент Y и отвечающие за цвет компоненты U и V разбиваются на блоки 8х8 пикселей. Каждый такой блок подвергается дискретному косинусному преобразованию (ДКП). Полученные коэффициенты ДКП квантуются (для Y, U и V в общем случае используются разные матрицы квантования) и пакуются с использованием кодов Хаффмана. Стандарт JPEG допускает также использование арифметического кодирования, однако из-за патентных ограничений (патент на описанный в стандарте JPEG арифметический QM-кодер принадлежит IBM) на практике оно используется редко. В популярную библиотеку libjpeg последних версий включена поддержка арифметического кодирования, но с просмотром сжатых с использованием этого метода изображений могут возникнуть проблемы, поскольку многие программы просмотра не поддерживают их декодирование.

Матрицы, используемые для квантования коэффициентов ДКП, хранятся в заголовочной части JPEG-файла. Обычно они строятся так, что высокочастотные коэффициенты подвергаются более сильному квантованию, чем низкочастотные. Это приводит к огрублению мелких деталей на изображении. Чем выше степень сжатия, тем более сильному квантованию подвергаются все коэффициенты.

При сохранении изображения в JPEG-файле указывается параметр качества, задаваемый в некоторых условных единицах, например, от 1 до 100 или от 1 до 10. Большее число обычно соответствует лучшему качеству (и большему размеру сжатого файла). Однако даже при использовании наивысшего качества (соответствующего матрице квантования, состоящей из одних только единиц) восстановленное изображение не будет в точности совпадать с исходным, что связано как с конечной точностью выполнения ДКП, так и с необходимостью округления значений Y, U, V и коэффициентов ДКП до ближайшего целого. Режим сжатия Lossless JPEG, не использующий ДКП, обеспечивает точное совпадение восстановленного и исходного изображений, однако его малая эффективность (коэффициент сжатия редко превышает 2) и отсутствие поддержки со стороны разработчиков программного обеспечения не способствовали популярности Lossless JPEG.

К недостаткам сжатия по стандарту JPEG следует отнести появление на восстановленных изображениях при высоких степенях сжатия характерных артефактов: изображение рассыпается на блоки размером 8x8 пикселей (этот эффект особенно заметен на областях изображения с плавными изменениями яркости), в областях с высокой пространственной частотой (например, на контрастных контурах и границах изображения) возникают артефакты в виде шумовых ореолов (рис. 1).

Риc. 1 Артефакты JPEG возникающие при большой степени сжатия

Следует отметить, что стандарт предусматривает использование специальных фильтров для подавления блоковых артефактов, но на практике подобные фильтры, несмотря на их высокую эффективность, практически не используются. Однако, несмотря на недостатки, JPEG получил очень широкое распространение из-за достаточно высокой (относительно существовавших во время его появления альтернатив) степени сжатия, поддержке сжатия полно цветных изображений и относительно невысокой вычислительной сложности.

## JPEG2000

JPEG 2000 (или jp2) — графический формат, который вместо дискретного косинусного преобразования, применяемого в формате JPEG, использует технологию вейвлет-преобразования, основывающуюся на представлении сигнала в виде суперпозиции базовых функций — волновых пакетов.

В результате такой компрессии изображение получается более гладким и чётким, а размер файла по сравнению с JPEG при одинаковом качестве оказывается меньшим. JPEG 2000 полностью свободен от главного недостатка JPEG: благодаря использованию вейвлетов, изображения, сохранённые в этом формате, при высоких степенях сжатия не содержат артефактов в виде «решётки» из блоков размером 8х8 пикселей.

Артефакты, возникающие при сжатии по алгоритму JPEG 2000 с высокой степенью компрессии (потерь), отличаются от артефактов, возникающих при сжатии компрессором JPEG — в тех местах, где оригинальное изображение имело плавные переходы цветов, — они становятся ещё более плавными (размытыми); в тех же местах где были резкие переходы яркости или цвета (участки изображения с высокой контрастностью) — возникает характерный артефакт в виде яркого контура, обрамляющего границу перехода, незначительно (на пару пикселей) отступающего от более тёмного участка перехода. Различия в артефактах — нет сетки в 8 на 8 пикселей; не искажаются цвета мелких деталей, сильно отличающихся по цвету от фона; артефакты, характерные для JPEG 2000, становятся заметными при больших, чем в случае JPEG, степенях сжатия.

Схема кодирования, используемая в стандарте JPEG 2000, кратко может быть описана следующим образом. Исходное изображение разбивается на прямоугольные сегменты (tile), которые кодируются независимо. После преобразования цветового пространства каждый сегмент изображения подвергается процедуре многоуровневого дискретного вейвлетного преобразования. Для этого используются низкочастотный и высокочастотный вейвлетные фильтры.

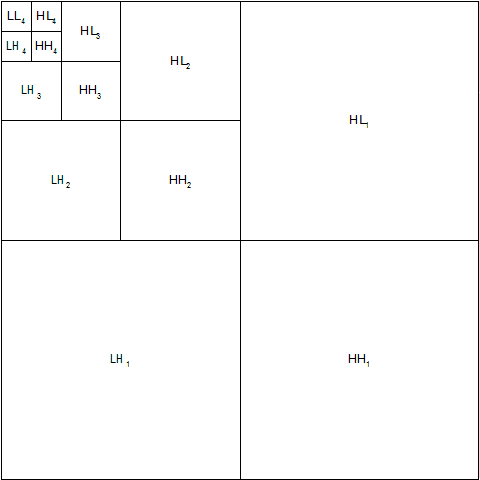
При разложении сигнала сегмента изображения вначале выполняется разложение по строкам, а затем по столбцам. Результатом разложения являются 4 матрицы: HH1, HL1, LH1, LL1, соответствующие фильтрации фильтром по строкам и по столбцам, фильтром по строкам и по столбцам, фильтром по строкам и по столбцам, фильтром по строкам и столбцам. Далее производится децимация (прореживание) полученных матриц по строками столбцам с коэффициентом 2. Затем матрица LL1 подвергается дальнейшему вейвлетному разложению [4][5]. Его результатом являются матрицы HH2, HL2, LH2, LL2 (picture). Такое разложение повторяется v раз. Результатом разложения является набор из 3v + 1 матриц уменьшающейся размерности (рис. 2).

Рис.2 Результат двумерного вейвлетного преобразования

Каждая из матриц разложения делится на блоки, которые кодируются независимо друг от друга при помощи контекстно-зависимого адаптивного двоичного арифметического кодера.

Коэффициенты матриц вейвлетного разложения представляются в двоичном виде. Все биты коэффициентов, относящихся к одному разряду, образуют битовую плоскость. Нумерация битовых плоскостей начинается с самого старшего разряда. Самая старшая битовая плоскость, в которой есть хотя бы один ненулевой коэффициент, является наиболее значимой. Кодирование коэффициентов матриц разложения происходит на уровне битовых плоскостей, при этом каждая битовая плоскость кодируется с помощью арифметического кодера за три прохода: первый называется significance propagation, второй — magnitude refinement и третий — cleanup.

Кодирование плоскостей начинается с наиболее значимой битовой плоскости, для которой применяется только третий проход (cleanup), для всех остальных битовых плоскостей применяются все три прохода.

Каждому коэффициенту в блоке на этапе кодирования соответствует степень значимости (significance state). Степень значимости может принимать два значения: 1, т. е. коэффициент считается значимым, и 0, что говорит о том, что коэффициент незначимый. Первоначально все коэффициенты считаются незначимыми. После кодирования первого значимого бита (т. е. единицы) коэффициент становится значимым. Каждому коэффициенту сопоставляется 8 разрядный «контекстный вектор», состоящий из степеней значимости восьми соседних коэффициентов (picture). По сумме компонент вектора определяется номер контекста.

|  |  |  |
| --- | --- | --- |
| a1 | a2 | a3 |
| a4 | **X** | a5 |
| a6 | a7 | a8 |

Первый проход (significance propagation). Данный проход включает в себя кодирование бит незначимых коэффициентов, которые имеют ненулевой контекст. Если на данном проходе кодируется единичный бит, то степень значимости коэффициента, к которому этот бит относится, становится равной 1, после чего кодируется знак коэффициента.

Кодирование знака. Номер контекста при кодировании знака зависит от знака и значимости соседних коэффициентов по вертикали (a4, a5) и по горизонтали (a2, a7). После вычисления контекста кодируемый знак представляется как сумма по модулю 2 между действительным значением и предсказанным.

Второй проход (magnitude refinement). На данном проходе кодируются биты значимых коэффициентов. Номер контекста вычисляется по всем соседним коэффициентам. При этом делается различие между битом, стоящим сразу после первого значимого бита, и оставшимися битами этого коэффициента.

Третий проход (cleanup). На этом проходе кодируется наиболее значимая битовая плоскость, а также биты незначимых коэффициентов, для которых значение контекста во время первого прохода было равно 0. На этом проходе используются контексты первого прохода, а также еще два специальных контекста.

Если в столбце четыре идущих подряд коэффициента кодируются на этом проходе и значения контекстов для них равны 0, тогда используется специальный контекст «Runlength». Если все коэффициенты остаются незначимыми (все 4 бита в столбце в текущей битовой плоскости равны 0), то вместе с контекстом «Runlength» на арифметический кодер подается 0. Если хотя бы один из коэффициентов становится значимым на этом проходе (значение бита на текущей битовой плоскости равно 1), то на арифметический кодер подается 1, а далее кодируются еще 2 бита, которые сообщают о местоположении первой единицы в колонке. Так как вероятности этих двух бит равномерно распределены, то для их кодирования используется еще один специальный контекст «Uniform». После того как позиция первого ненулевого бита определена, оставшиеся значения в колонке кодируются так же, как на первом проходе с использованием тех же 9 контекстов.

## JPEG-LS

Группа экспертов в области фотографии (Joint Photographic Experts Group) в дополнение к известным форматам сжатия изображений JPEG и JPEG 2000, ориентированным, прежде всего на сжатие с потерями, предложила также стандарт на сжатие без потерь — JPEG-LS (в котором, однако, предусмотрен также режим сжатия с ограниченными потерями).

Алгоритм сжатия, лежащий в основе JPEG-LS, использует адаптивное предсказание значения текущего пикселя по окружению, включающему уже закодированные пиксели (метод Median Edge Detection), классификацию контекста, контекстное моделирование ошибки предсказания и её коррекцию, а также энтропийное кодирование скорректированной ошибки предсказания (используется кодирование Голомба-Райса). Для повышения эффективности кодирования низко энтропийных изображений (или фрагментов изображений) алгоритм предусматривает автоматический переход в режим кодирования длин серий, что позволяет использовать его для сжатия без потерь (или с ограниченными потерями) не только фотореалистичных изображений, но и компьютерной графики.

Предсказание в алгоритме JPEG-LS выполняется согласно следующей модели:

|  |  |  |
| --- | --- | --- |
| С | B | D |
| A | **X** |  |

Контексты для кодирования составляются согласно локальному градиенту, который вычисляется следующим образом:

Для JPEG-LS согласно полученным для локального градиента разностям выделяют по 9 регионов, что даёт в общей сложности 729 контекстов, однако, учитывая симметрию, общее число контекстов составляет 365.

Локальный градиент отражает степень гладкости или «ребристости» соседних с X отсчётов. Данные разности тесно связаны со статистическим поведением ошибок предсказания.

Формат JPEG-LS разрабатывался, прежде всего, для хранения изображений в медицинских целях, то есть для тех случаев, когда важно иметь большое изображение без малейших потерь качества. За основу был взят формат LOCO-I, разработанный в стенах «HP Labs». Затем он был доработан совместными усилиями «Hewlett-Packard» и «Mitsubishi».

## WebP

WebP — формат сжатия изображений с потерями, предложенный компанией Google в 2010 году. Алгоритм основан на алгоритме сжатия неподвижных изображений (ключевых кадров) из видео кодека VP8.

Сжатие состоит из двух этапов. На первом делается попытка «предсказать» содержимое одних блоков по уже декодированным (три блока над текущим и один блок слева от него), на втором кодируется ошибка предсказания. Блоки рассматриваются в порядке слева направо и сверху вниз. Режимы предсказания работают с 3 размерами макроблоков:

* 4x4 в канале яркости
* 16x16 в канале яркости
* 8x8 в канале цветности

В VP8 и WebP реализованы режимы предсказания:

* Горизонтальное, H\_PRED. Каждый столбец текущего блока есть копия столбца слева от текущего блока.
* Вертикальное, V\_PRED. Каждая строка текущего блока есть копия строки над текущим блоком.
* DC предсказание, DC\_PRED. Заполняет весь блок одинаковыми значениями, полученными за счет усреднения значений пикселей вышележащей строки и столбца слева от текущего блока.
* Предсказание True Motion, TM\_PRED. Разработано в On2 Technologies. Кроме строки над блоком и столбца слева от него, используется пиксель, расположенный сверху-слева от блока. Разница между угловым пикселем и строкой сверху записывается в строки блока, при этом к значениям добавляется значение соответствующего пикселя из столбца.

Для блоков 4x4 реализовано 6 дополнительных режимов, сходных с V\_PRED и H\_PRED, но с диагональными направлениями.

Для сжатия ошибок предсказания и подблоков, которые не были предсказаны, используется дискретное косинусное преобразование DCT (и, изредка, преобразование Уолша—Адамара, WHT). Оба преобразования работают с подблоками размером 4х4 пикселя. Реализация преобразований выполнена на представлении чисел с фиксированной точностью, чтобы уменьшить ошибки округления. Коэффициенты DCT и WHT пакуются энтропийным кодеком.

# Разработка и обоснование алгоритма сжатия

## Введение

Арифметическое кодирование обеспечивает почти оптимальную степень сжатия с точки зрения энтропийной оценки кодирования Шеннона и показывает высокую эффективность для дробных неравномерных интервалов распределения вероятностей кодируемых символов. Однако в случае равновероятного распределения символов метод арифметического кодирования приближается к префиксному коду Хаффмана и даже может занимать на один бит больше. [6]

Кроме того, характерной особенностью энтропийных методов кодирования и, в частности, арифметического кодирования является тот факт, что более вероятные символы представляются более короткими кодовыми словами, а менее вероятные – более длинными. Таким образом, для эффективного сжатия при использовании арифметического кодирования очень важным оказывается характер входного распределения.

Под адаптивным арифметическим кодированием в работе следует понимать не только одноимённый метод кодирования с адаптивной моделью, но и предварительную адаптацию разработанного алгоритма сжатия изображений под конкретное изображение с целью улучшить параметры входного распределения. Данная адаптация выполняется при помощи специального метода линейного предсказания, который используется в данной работе.

Адаптация и последующее арифметическое кодирование применяются к результату вейвлетного преобразования. Вейвлетное преобразование позволяет разбить изображение на области высоких и низких частот. Важной частью работы является анализ характерных особенностей кодирования этих областей c целью достижения наилучших результатов.

## Общая схема разработанного алгоритма

Данная работа не предполагает написания готового промышленного алгоритма сжатия изображений. Поэтому был принят ряд ограничений, которые позволили сосредоточиться на наиболее важных частях алгоритма. В работе рассматривается обработка и сжатие полутоновых изображений, высота и ширина которых равны между собой и являются степенью двойки. В работе так же предполагается, что размеры изображений не превышают 1024х1024 отсчётов, если же изображение больше, то можно воспользоваться техникой, которая применяется в JPEG2000 – изображение разбивается на прямоугольные области (tile), которые кодируются независимо. Кроме того в работе рассматривается только алгоритм сжатия без потерь.

Предложенный в работе алгоритм состоит из трёх основных частей:

1. Двумерное дискретное вейвлетное преобразование оригинального изображения
2. Обработка низкочастотной области изображения методом линейного предсказания
3. Арифметическое кодирование полученных данных

На вход алгоритму подаётся полутоновое изображение, к которому применяется вейвлетное преобразование. Количество уровней вейвлетного преобразования является варьируемым параметром, в случае, если количество преобразований равно нулю первая часть алгоритма пропускается. Если вейвлетное преобразование не выполняется, то выполняется линейное предсказание для всего оригинального изображения (т.е. всё оригинальное изображение принимается за низкочастотную область) и последующее кодирование результатов.

В работе используется многопроходный адаптивный метод линейного предсказания. Именно он позволяет преобразовать и адаптировать низкочастотную область таким образом, чтобы улучшить распределение, подаваемое на вход арифметическому кодеру за счёт использования избыточности данных в низкочастотной области. В начале работы алгоритма низкочастотная область изображения разбивается на активные и пассивные зоны: рассматриваются блоки изображения размером 4x4. Выбирается некоторый порог дисперсии, если дисперсия блока превышает заданный порог дисперсии, то данный блок считается активным, иначе – пассивным.

Затем для активной и пассивной области составляются системы уравнений в соответствии с формулой для линейного предсказания 4 порядка и при помощи метода наименьших квадратов находятся параметры моделей предсказания. Таким образом, для гладких областей изображения используется одна модель предсказания, а для областей с выраженным перепадом значений – другая. В общем случае, алгоритм позволяет использовать и большее количество областей. Области так же используются при сжатии высокочастотных областей.

После этого составляется разность между предсказанием и оригинальной низкочастотной областью. В первую очередь кодируются параметры модели предсказания и распределение зон.

Далее выполняется контекстно-зависимое арифметическое кодирование - так для низкочастотной области (разности предсказания и оригинала) в качестве контекста выступают уже переданные значения соседних точек и область (пассивная или активная), а для остальных областей – непредсказанная низкочастотная область, область и соседние точки.

Процесс декодирования выполняется в обратном порядке начиная с низкочастотной области. Зная параметры модели предсказания и распределение зон, по разностям восстанавливаются оригинальные значения низкочастотной области. Затем, восстанавливаются высокочастотные области соответствующего уровня вейвлетного преобразования. Если вейвлетное преобразование было многоуровневым, то по восстановленным низкочастотным и высокочастотным областям декодированного уровня восстанавливается низкочастотная область предыдущего уровня. Процесс восстановления происходит до получения изображения.

## Вейвлетное преобразование изображения

Одним из наиболее важных параметров вейвлетного преобразования является выбор базиса преобразования или фильтров, которые используются при преобразовании. Вообще говоря, вейвлетное преобразование выполняется над множеством действительных чисел, что порождает проблемы при сжатии без потерь. Так же при использовании вейвлетного преобразования может возникать так называемый граничный эффект – область, в которой коэффициенты вейвлетного преобразования искажаются разрывной природой границы сигнала. Однако существуют базисы и схемы фильтрации, которые позволяют восстанавливать исходный сигнал без потерь [10]. Именно такая схема применяется в JPEG-2000 для сжатия изображений без потерь.

Так как в работе также рассматривается сжатие без потерь, то был выбран тот же базис и способ фильтрации, что и в JPEG-2000, а именно LeGall 5/3 биортогональный вейвлет в качестве базиса и так называемая лифтинг схема [10].

Коэффициенты фильтров данного базиса:

|  |  |  |
| --- | --- | --- |
| k | Lowpass filter | Highpass filter |
|  |  |  |
|  |  |  |
|  |  |  |

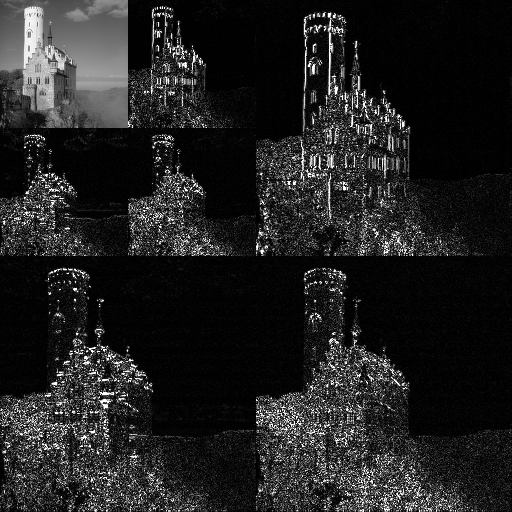
Рассмотрим результат после двух уровней преобразования (рис. 3):

Рис. 3 результат двух уровней двумерного дискретного вейвлетного преобразования

На изображении видно, что низкочастотная область напоминает исходное изображение. На самом деле, энтропия низкочастотной области также остаётся почти как у исходного изображения. А вот остальные области обладают меньшей энтропией по сравнению с оригинальным изображением, теоретически это означает, что преобразованное изображение может быть сжато больше, чем исходное, так как средняя энтропия уменьшилась.

Ещё одна особенность, которую видно на изображении – там, где оригинальное изображение было однородно и гладко, на высокочастотных областях почти не возникает коэффициентов с высокими значениями. Это означает, что по оригинальной картинке или по низкочастотной области (так как она сильно похожа на оригинальное изображение) мы можем попытаться выделить такие зоны. Кроме того очевидно, что стоит воспользоваться этой информацией при контекстно-зависимом кодировании.

Несмотря на то, что в работе рассматривается только сжатие без потерь, стоит так же упомянуть, что основная особенность вейвлетного преобразования состоит в том, что квантование коэффициентов в высокочастотных областях значительно усиливает сжатие, не приводя при этом к сильному снижению качества восстановленного изображения.

## Линейное предсказание

Достаточно важную часть работы составляет метод линейных предсказаний, так как именно с его помощью выполняется адаптация низкочастотной области с целью улучшить параметры входного распределения для арифметического кодера. Делается это за счёт избыточности содержащейся в низкочастотной области.

Попытка применить линейное предсказание для высокочастотных областей ни по соседним отсчетам, ни по отчётам низкочастотной области не дала ощутимого результата. Можно предположить, что данные внутри высокочастотных областей оказываются сильно декоррелированы между собой благодаря вейвлетному преобразованию, а связь с низкочастотной областью не линейна.

Обобщённая схема разработанного алгоритма линейного предсказания такова:

1. Вся низкочастотная область разбивается на n зон по какому-либо критерию (например, по дисперсии блоков размером 4х4)
2. Для каждой зоны выбирается k начальных фильтров (или k наборов различных начальных параметров модели предсказания). При этом размерность фильтров зависит от порядка линейного предсказания
3. Для каждой зоны выполняется локальная оптимизация наборов параметров по следующему алгоритму:
   1. Высчитывается средняя квадратичная ошибка предсказания каждого набора для каждого отсчёта.
   2. Каждому набору приписываются отсчёты, для которых данный набор дал наименьшую ошибку предсказания. Получаем подмножества отсчётов для каждого из наборов
   3. Для каждого полученного подмножества вычисляем новые параметры модели предсказания при помощи метода наименьших квадратов.
   4. Повторяем процедуру с новыми наборами параметров модели предсказания. Количество итераций является варьируемым параметром.
4. Для каждой зоны имеется k локально-оптимальных фильтров с точки зрения средней квадратичной ошибки.
5. Выполняется предсказание.

Опишем процедуру локальной оптимизации подробней. Пусть используется модель предсказания третьего порядка. Тогда для каждой точки изображения её предсказание вычисляется по следующей формуле:

где a1, a2, a3 – параметры модели предсказания

Вычислим для всех наборов параметров модели предсказания среднее квадратичное отклонение предсказания от значения точки. Найдём для каждой точки параметры модели, которые дают минимальное отклонение. Для каждого набора параметров модели получим множество точек, которые данная модель предсказывает лучше остальных (рис. 4).

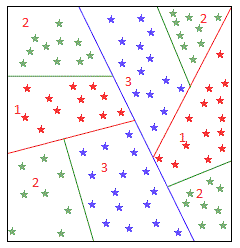


Рис. 4 иллюстрация подмножеств. Номер обозначает набор параметров модели.

Теперь для каждого такого подмножества выполним процедуру оптимизации: решим систему исходных уравнений каждого подмножества относительно параметров модели при помощи метода наименьших квадратов:

1. Пусть вектор неизвестных параметров
2. Пусть A матрица, составленная из отсчётов изображения, по которым идёт предсказание. Матрица A не квадратная, а прямоугольная – такая, что число уравнений больше числа неизвестных.
3. Пусть b вектор предсказываемых значений точек (оригинальных значений)
4. Получаем систему уравнений . Такая система уравнений, в общем случае не имеет решения (если ранг на самом деле больше числа переменных). Поэтому эту систему можно "решить" только в смысле выбора такого вектора, чтобы минимизировать "расстояние" между векторами и . Для этого можно применить критерий минимизации суммы квадратов разностей левой и правой частей уравнений системы, т.е.
5. Решение этой задачи минимизации приводит к решению следующей системы уравнений

Решение последней системы даёт новые коэффициенты параметров модели, оптимальные с точки зрения средней квадратичной ошибки на подмножестве данной модели. Таким образом, общая средняя квадратичная ошибка уменьшается. Оказывается, что вместе с уменьшением общей средней квадратичной ошибки уменьшается и энтропия, но только до определённого порога.

Однако обобщённая схема довольно сложна как с вычислительной точки зрения, так и с точки зрения последующего кодирования. Она нуждается в дополнительных исследованиях, которые достойны отдельной исследовательской работы.

В данной работе используется вырожденная версия обобщённого алгоритма:

1. Вся низкочастотная область разбивается на 2 зоны по дисперсии блоков размером 4х4 (порог дисперсии является варьируемым параметром)
2. Для каждой зоны выбирается только по одному фильтру. Оптимальные фильтры получаются методом наименьших квадратов. Количество итераций равно единице. Порядок предсказания равен четырём.
3. Выполняется предсказание.

E:\Dropbox\Dropbox\Учёба\Magistr Work\Пояснительная записка\Изображения\Castle512.tifРассмотрим пример работы алгоритма применяемого в работе:

Рис. оригинальное изображение

Применим алгоритм к изображению (рис. 5) (считаем, что всё изображение - низкочастотная область)

Сначала получим распределение зон (рис. 6) (в данном случае порог дисперсии равен 50):

E:\Dropbox\Dropbox\Учёба\Magistr Work\Пояснительная записка\Изображения\zones(Castle512).tif

Рис. 6 Распределение зон. Тёмные участки – пассивные области, светлые – активные.

Видно, что однородные области, такие как небо, вода и облака – отнеслись к пассивной области, в то время, как контрастная кирпичная кладка замка и деревья – оказались в активной зоне из-за контрастных перепадов яркости.

Таким образом, удаётся отделить участки изображения с плавными переходами от участков изображения с резкими границами. Это очень важно, потому что линейное предсказание хорошо справляется именно с гладкими участками, максимально устраняя избыточность в таких областях.

E:\Dropbox\Dropbox\Учёба\Magistr Work\Пояснительная записка\Изображения\Diff(Castle512).tifТеперь рассмотрим разность между предсказанием и оригинальной низкочастотной областью (рис. 7):

Для пассивной и активной области были найдены следующие параметры моделей: [-0.2668, 0.6271, 0.1172, 0.5224] и [-0.4427, 0.7110, 0.0865, 0.6429] соответственно.

Рис. 6 Разность между предсказанием и оригинальным изображением  
 (сдвинутая относительно нуля для наглядности)

Исходная энтропия изображения при этом снизилась с 7.447 до 4.201 в то, время как стандартные методы вроде горизонтального или вертикального предсказания дают энтропию порядка 5.5 бит. В то же время обобщённая версия алгоритма даёт оценку порядка 3.9 бит.

Однако в качестве дополнительной информации теперь следует также закодировать выбранные фильтры и распределение зон, чтобы декодер смог восстановить изображение.

Распределение коэффициентов фильтров носит случайный характер, поэтому они кодируются без какого-либо сжатия. А для зон хорошо подходит алгоритм типа deflate. Разность между предсказанием и оригинальным изображением кодируется арифметическим кодером.

Адаптивность данного метода заключается именно в подборе и передаче фильтров для каждого изображения. То есть алгоритм не использует известных заранее моделей предсказаний, как это делает большинство подобных алгоритмов. С другой стороны, данный алгоритм ещё и значительно улучшает статистическое распределение отсчётов изображения.

Рассмотрим распределение отсчётов для исходных (рис. 8) и кодируемых данных (рис. 9):

E:\Dropbox\Dropbox\Учёба\Magistr Work\Пояснительная записка\Изображения\hist(Castle512-orig).tif

Рис. 7 распределение отсчётов на оригинальном изображении

E:\Dropbox\Dropbox\Учёба\Magistr Work\Пояснительная записка\Изображения\hist(Castle512-pred).tif

Рис. 8 распределение отсчётов на разности между предсказанием и оригиналом

Видно, что после предсказания статистическое распределение улучшилось с точки зрения арифметического кодирования: огромное количество нулевых отсчётов с большой вероятностью. Отсчёты, вероятность которых мала, можно попробовать кодировать другим способом, выдавая в арифметическом кодере так называемый эскейп код.

Более того, как оказалось, распределение зон, полученное на этапе линейного предсказания, может быть использовано при кодировании не только низкочастотной области, но и при кодировании остальных областей полученных после вейвлетного преобразования, так как большинство ненулевых коэффициентов вейвлетного преобразования находятся именно в активных областях.

Ещё одной особенностью данного метода является то, что он может быть использован в качестве предобработки изображения для других кодеров при их небольшой модификации (кодирование и считывание параметров модели и распределения зон).

## Контекстно-зависимое арифметическое кодирование

Для арифметического кодирования в работе использовался адаптивный арифметический кодер, работающий по следующей модели: вначале символам алфавита назначался вес равный единице, а вероятности символов оценивались как отношение веса к общему числу символов алфавита. При кодировании символа, его вес увеличивается на единицу. По достижении границы точности имплементации арифметического кодера, веса символов делятся пополам. С одной стороны это должно ухудшать степень сжатия, с другой стороны вносит некоторый эффект локальности, то есть со временем старая информация теряет своё значение, что положительно сказывается на сжатии.

Алфавит кодера является ограниченным и фиксированным, но в результате вейвлетного преобразования нам могут встретиться символы, выходящие за границы этого алфавита. Однако вероятность таких символов крайне мала, поэтому для них арифметически кодер выдаёт эскейп символ, а само значение кодируется другим кодером в отдельный поток. Другой кодер кодирует такие значения следующим образом: зная максимальное и минимальное значение для текущей кодируемой области, для любого символа из этого интервала в поток записывается бит (range – размер диапазона от минимального значения до максимального). Так как вейвлетное преобразование нормировано и значения находятся в разумных пределах, такой подход даёт хорошие результаты.

Важной частью арифметического кодирования в работе является составление контекстов для кодирования. Контекстно-зависимое кодирование позволяет воспользоваться тем фактом, что условная энтропия не превосходит безусловной энтропии. В качестве условия выступает текущий контекст, причём, если есть зависимость распределения от контекста, то условная энтропия оказывается меньше безусловной, что позволяет кодировать данные более эффективно.

Для назначения контекста могут использоваться только уже переданные данные, иначе декодер не сможет определить контекст. Для низкочастотной области при кодировании очередного символа для назначения контекста имеет смысл использовать соседние отсчёты, которые были уже закодированы и активность (активная/пассивная) зоны, в которой находится данный отсчёт.

В работе используется три соседних отсчёта:

|  |  |
| --- | --- |
|  |  |
|  |  |

Исходя из анализа того, как работает линейное предсказание, было сделано предположение, что для назначения контекста надо использовать сумму соседних отсчётов и разность отсчётов на рёбрах ( и ).

Рассмотрим график зависимости ошибки предсказания в пассивной области некоторого изображения от суммы соседних отсчётов (рис. 9):

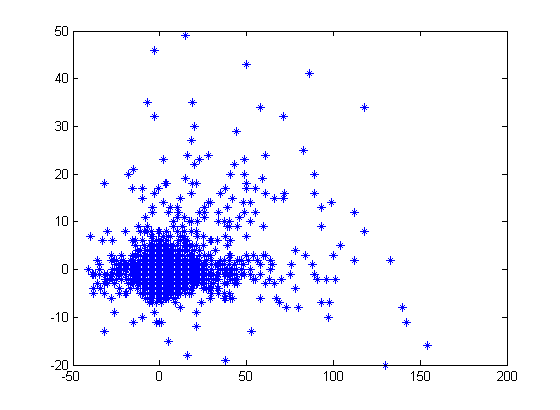
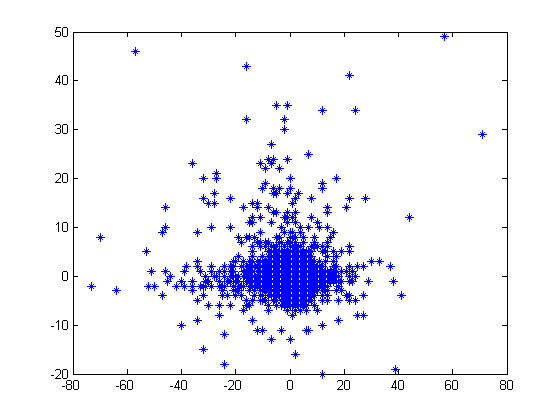


Рис. 9 распределение ошибки предсказания в пассивной области.   
По оси OX – сумма, OY – значение ошибки предсказания

Большинство элементов сконцентрировано в области, соответствующей сумме от -15 до 15. Чем больше сумма, тем меньше там элементов, а их распределение всё случайней. Интерес так же представляют собой элементы, которые находятся в верхней области графика. Оказывается, что эти элементы соответствуют большим разностям на рёбрах.

Рассмотрим график зависимости ошибки предсказания в пассивной области от разности (рис. 10)



Ри. 10 распределение ошибки предсказания в пассивной области.  
По оси OX – разность B-A, OY – значение ошибки предсказания

На графике (рис. 10) видно, что два максимальных элемента соответствуют большим по модулю значениям разности. Это говорит о том, что, несмотря на то, что для данных элементов сумма оказалась близка к нулю, разность на одном из рёбер была высока.

Так как графики достаточно симметричны, то можно рассматривать модули суммы и разностей. Исходя из этого, для модуля суммы и модулей разностей на рёбрах выделяется по 4 региона, что даёт 64 контекста для пассивной области. Для активной области также выделяется 64 контекста, но границы регионов смещены, так как в активных зонах распределение отличается от пассивных зон.

# Практическая реализация

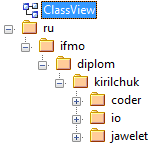
## Выбранные инструменты

Для разработки необходимых программных решений и написания алгоритма сжатия изображений был выбран объектно-ориентированный язык программирования Java. Такой выбор обусловлен относительной простотой данного языка программирования и наличием большого количества свободных библиотек с открытым исходным кодом под данный язык программирования. В работе были использованы несколько таких библиотек. Наиболее важным критерием выбора данного языка являлось наличие опыта программирования на нём автора работы.

Для анализа полученных результатов использовался пакет прикладных программ для решения задач технических вычислений MATLAB. В составе пакета MATLAB имеется большое количество функций для построения графиков, в том числе трёхмерных, а также визуального анализа данных.

При получении результатов кодирования для JPEG2000 использовался пакет программ для неинтерактивной (пакетной) обработки графических файлов imagemagick.

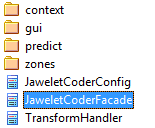
## Общая структура классов и модулей



Приложение содержит три модуля:

* coder – содержит классы, связанные непосредственно с алгоритмом кодирования
* io – содержит классы, связанные с преобразованием данных при вводе\выводе через потоки данных
* jawelet – содержит классы, связанные с вейвлетным преобразованием, а также основные утилитные классы.

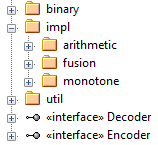
#### Модуль coder



Основным классом данного модуля является класс JaweletCoderFacade, который содержит главный метод для кодирования исходного изображения. Входящие параметры: исходное изображение, конфигурация кодера и путь до целевого файла (файл, куда будет сохранён результат сжатия).

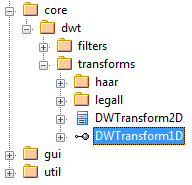
Подмодули, содержащиеся в данном модуле, отвечают за определение контекстов для контекстно-зависимого кодирования, графический интерфейс для пакетной обработки изображений, линейное предсказание и определение пассивных и активных зон изображения.

#### Модуль io



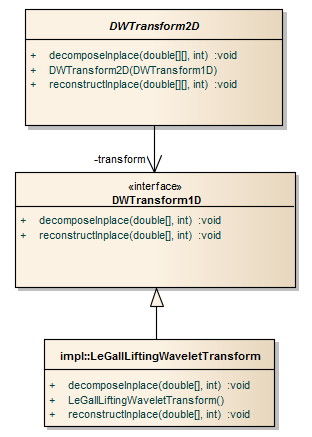
В данном модуле содержаться интерфейсы для кодера и декодера (кодер принимает на вход целочисленное значение, кодирует его и записывает в исходящий поток, декодер – декодирует следующее значение из потока) и их реализации.

#### Модуль jawelet



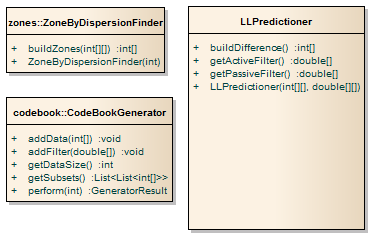
Основным интерфейсом данного модуля является интерфейс DWTransform1D, он содержит основные методы, связанные с одномерным дискретным вейвлетным преобразованием. Модуль также содержит две имплементации данного интерфейса для разных типов вейвлетов. Класс DWTransform2D содержит основные методы, связанные с двумерным дискретным вейвлетным преобразованием. Также модуль содержит подмодуль графического интерфейса для отображения результата вейвлетного преобразования.

## Вейвлетное преобразование



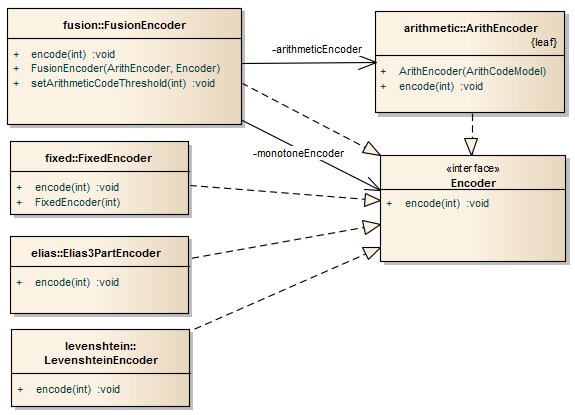
Двумерное вейвлетное преобразование содержит методы для преобразования исходного изображения с задаваемым количеством уровней преобразования и восстановления из заданного уровня. Конструктор двумерного преобразования принимает реализацию одномерного преобразования в качестве параметра. Реализация представлена в работе классом LeGallLiftingWaveletTransform.

## Линейное предсказание



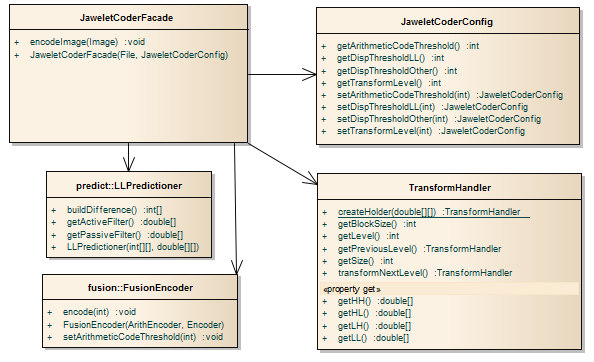
Сначала по заданному порогу дисперсии находится распределение зон, затем экземпляр класса LLPredictioner на основании зон, используя экземпляры класса CodeBookGenerator, выполняет поиск оптимальных фильтров. Затем составляется разность между оригиналом и предсказанием.

## Кодирование (ввод\вывод)



Для кодирования каждого из контекстов используется по экземпляру класса FusionEncoder. FusionEncoder агрегирует в себе два кодера: арифметический и обычный (одна из трёх представленных реализаций). Суть заключается в том, что если кодируемое значение превышает arithmeticCodeThreshold, то в арифметическом кодере выдаётся эскейп код и кодирование выполняет обычный кодер.

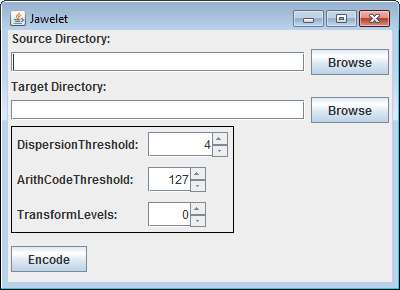
## Разработанный алгоритм кодирования изображения



Экземпляр класса JaweletCoderFacade использует все выше перечисленное классы для того, чтобы закодировать изображение. Сначала при помощи объекта класса TransformHandler выполняется вейвлетное преобразование, затем его низкочастотная область предсказывается и затем выполняется контекстно-зависимое кодирование.

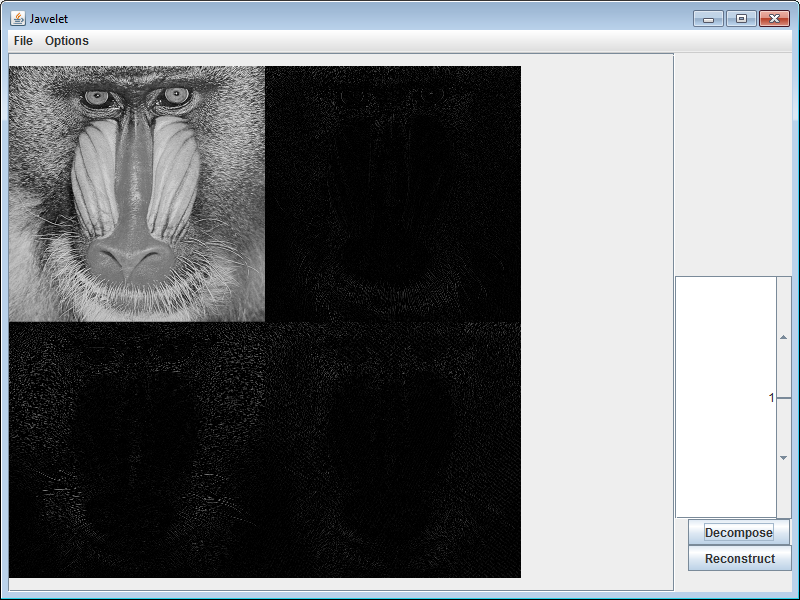
## Интерфейс пользователя

Основным интерфейсом программы является интерфейс для пакетной обработки изображений:



В интерфейсе задаются параметры кодирования, директория с исходными изображениями и директория для хранения результата кодирования.

Также в программе есть вспомогательный интерфейс для визуального просмотра результата вейвлетного преобразования:



# Результаты работы

В результате работы был разработан алгоритм сжатия цифровых изображений, который сочетает в себе использование метода линейных предсказаний, вейвлетное преобразование, а также контекстно-зависимое арифметическое кодирование.

В процессе разработки алгоритма был применён нестандартный подход к линейному предсказанию. Данный подход описан в работе и представляет интерес для дальнейших исследований.

В качестве результатов работы алгоритма, приведём сравнение с результатами JPEG2K на некотором наборе изображений. В набор входят различные изображения от портретов до пейзажей размером от 128х128 до 1024х1024 пикселей:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Изображение | Несжатый размер (byte) | JPEG2K (byte) | Разработанный алгоритм (byte) | Коэффициент сжатия относительно JPEG2K |
| 4.1.04.tiff | 65934 | 32167 | 32590 | 1,30% |
| 4.1.05.tiff | 65934 | 32916 | 33915 | 2,95% |
| 4.1.06.tiff | 65934 | 42865 | 41859 | -2,40% |
| 4.1.07.tiff | 65934 | 20574 | 19558 | -5,19% |
| 4.1.08.tiff | 65934 | 24070 | 23573 | -2,11% |
| 4.2.01.tiff | 262734 | 123564 | 123177 | -0,31% |
| 4.2.02.tiff | 262734 | 138454 | 138737 | 0,20% |
| 4.2.03.tiff | 262734 | 200247 | 200839 | 0,29% |
| 4.2.04.tiff | 262734 | 141147 | 140719 | -0,30% |
| 4.2.05.tiff | 262734 | 130880 | 128819 | -1,60% |
| 4.2.06.tiff | 262734 | 168905 | 165862 | -1,83% |
| 4.2.07.tiff | 262734 | 151546 | 150814 | -0,49% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Изображение | Несжатый размер (byte) | JPEG2K (byte) | Разработанный алгоритм (byte) | Коэффициент сжатия относительно JPEG2K |
| 5.1.09.tiff | 65934 | 43064 | 41447 | -3,90% |
| 5.1.10.tiff | 65934 | 48756 | 47723 | -2,16% |
| 5.1.11.tiff | 65934 | 28051 | 27797 | -0,91% |
| 5.1.12.tiff | 65934 | 30801 | 31675 | 2,76% |
| 5.1.13.tiff | 65934 | 18441 | 15162 | -21,63% |
| 5.1.14.tiff | 65934 | 43192 | 43448 | 0,59% |
| 5.2.09.tiff | 262734 | 170201 | 168864 | -0,79% |
| 5.2.10.tiff | 262734 | 188116 | 186572 | -0,83% |
| 5.3.01.tiff | 1049934 | 632143 | 623368 | -1,41% |
| 5.3.02.tiff | 1049934 | 717829 | 703073 | -2,10% |
| 7.1.01.tiff | 262734 | 155414 | 149521 | -3,94% |
| 7.1.02.tiff | 262734 | 122209 | 115630 | -5,69% |
| 7.1.03.tiff | 262734 | 162433 | 157687 | -3,01% |
| 7.1.07.tiff | 262734 | 175913 | 169619 | -3,71% |
| 7.1.09.tiff | 262734 | 169747 | 164423 | -3,24% |
| elaine.512.tiff | 262734 | 162084 | 159646 | -1,53% |
| gray21.512.tiff | 262734 | 2194 | 1515 | -44,82% |
| house.tiff | 262734 | 138769 | 142279 | 2,47% |
|  |  |  | Среднее (с 5.1.13.tiff и gray21.512.tiff) | -3,45% |
|  |  |  | Среднее (без 5.1.13.tiff и gray21.512.tiff) | -1,32% |

Как видно, несмотря на то, что алгоритм работает по совершенно иному принципу, результаты близки к эталонному алгоритму, а на некоторых изображениях превосходят его на один-два процента.

Два изображения, которые показали преимущество алгоритма более чем на 20 процентов – это просто удачные с точки зрения разработанного алгоритма изображения. Простые однотонные изображения с примитивными фигурами. В них линейное предсказание работает очень хорошо.

Преимущество разработанного алгоритма состоит в чуть большей степени сжатия, причем, если на фотографических изображениях, преимущество незначительно, то для простых геометрических фигур с однотонной или градиентной заливкой алгоритм значительно опережает JPEG2k. Кроме того разработанный алгоритм имеет некоторый потенциал по дальнейшему усовершенствованию, что в итоге может дать большее преимущество.

Недостатком представленного в работе алгоритма является его незавершённость, в частности, многие варьируемые параметры, такие как порог дисперсии для нахождения пассивных и активных зон, количество преобразований и другие задаются пользователем, а не подбираются алгоритмом автоматически. Кроме того, количество контекстов для арифметического кодирования может быть увеличено, так же имеет смысл использовать более специфичный арифметический кодер (например, MQ-кодер), так как используемый в работе кодер излишне универсален.

Ещё одним недостатком является вычислительная сложность алгоритма, однако, так как целью работы было исследование, а не готовый оптимальный алгоритм, эти недостатки нельзя считать существенным.

# Заключение

Данная работа относится к типу поисковых научно исследовательских работ, для таких работ важен не только и не столько конечный результат, сколько идеи, возникающие в процессе исследования.

В рамках работы был получен не только неплохой конечный результат, но и был разработан интересный подход к методу линейной фильтрации. Полное исследование данного подхода осталось за рамками работы, так как он не относится непосредственно к теме данной работы и не является её целью. Однако данный подход может стать темой для других исследовательских работ.

Подход с выделением пассивных и активных зон, которые влияют как на линейное предсказание, так и на контекстно-зависимое арифметическое кодирование тоже оказался довольно интересен сам по себе. Он не только позволяет выполнять линейное предсказание лучше, но и позволяет выделить зоны в высокочастотных областях вейвлетного преобразования, содержащие наибольшее количество близких к нулю элементов.

Кроме того, несмотря на то, что в работе рассмотрен только алгоритм сжатия без потерь, он может с лёгкостью быть дополнен до алгоритма сжатия с потерями, достаточно лишь добавить квантование коэффициентов вейвлетного преобразования в высокочастотных областях.

С другой стороны результаты работы таковы, что для алгоритма сжатия изображений без потерь, вейвлетное преобразование не даёт ощутимого выигрыша в сжатии и даже наоборот, ухудшает сжатие по сравнению с алгоритмом, использующим только линейное предсказание. Это может означать, что способ по кодированию коэффициентов вейвлетного преобразования предложенный в работе оказался не достаточно хорошим, следовательно, надо искать другой подход.

# Список литературы

[1] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. 2-е издание. Пер. с англ. 2005г., 863с.  
[2] Википедия. Свободная энциклопедия. [Электронный ресурс] – Режим доступа : <http://ru.wikipedia.org>  
[3] Wikipedia. The free encyclopedia. [Электронный ресурс] – Режим доступа : <http://en.wikipedia.org>  
[4] Strang G., Nguyen T., Wavelets and filter banks. Wellesley-Cambridge Press. 1996г., 485с.   
[5] Bocharova I. E., Compression for multimedia. Cambridge University Press, 2009г., 265с.  
[6] Кудряшов Б.Д, Теория информации. Издательство «СПб Питер», 2009г., 322с.  
[7] Paul G. Howard, Jefrey Scott Vitter, Practical Implementation of Arithmetic Coding, Brown University Department of Computer Science, 1992г, 30c.  
[8] Уэлстид С., Фракталы и вейвлеты для сжатия изображений в действии, Издательство «Триумф», 2003г.,320с.  
[9] Pascal G., Filter coefficients for popular wavelets, 2006г., 11c.  
[10] Daubechies I., Sweldens W., Wavelet transforms that map integers to integers, Academic Press, 1997г., 31c.  
[11] M. D. Adams and R. Ward, Wavelet Transforms in the JPEG-2000 Standard, in Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, Aug. 2001, vol. 1, pp. 160-163.