Оглавление

[Введение 2](#_Toc105530544)

[Актуальность 2](#_Toc105530545)

[Цели и задачи выпускной квалификационной работы 2](#_Toc105530546)

[Содержание работы 3](#_Toc105530547)

[Глава 1. Основные понятия 4](#_Toc105530548)

[1.1 Полутоновые изображения 4](#_Toc105530549)

[1.2 Дискретный сигнал 4](#_Toc105530550)

[1.3 Кратно-масштабный анализ 5](#_Toc105530551)

[1.4 Масштабирующие функции 6](#_Toc105530552)

[1.4 Вейвлет-преобразование 7](#_Toc105530553)

[1.5 Квантование 7](#_Toc105530554)

[1.6 Арифметическое кодирование 9](#_Toc105530555)

[1.7 Машинное обучение 10](#_Toc105530556)

[1.8 Квадратичная регрессия 11](#_Toc105530557)

[1.9 Случайный лес 12](#_Toc105530558)

[Глава 2. Обзор существующих методов сжатия изображений и разбор предлагаемого алгоритма 13](#_Toc105530559)

[2.1 Обзор существующих методов сжатия изображений 13](#_Toc105530560)

[2.2 JPEG 14](#_Toc105530561)

[2.3 JPEG-2000 17](#_Toc105530562)

[1.8 Контекстное кодирование 20](#_Toc105530563)

[Список использованной литературы 21](#_Toc105530564)

# **Введение**

## **Актуальность**

В современном мире интернет-технологии всё глубже проникают в нашу жизнь, возрастает значимость проблем хранения и передачи информации. В частности, одной из наиболее актуальных тем исследований является эффективное представление изображений. Развитие данной предметной области происходит в направлении сокращения объема, занимаемого данными при хранении информации. Иногда затраты на хранение и передачу данных становятся основным пунктом расходов, которые были бы сильно выше, если бы не существовало алгоритмов сжатия. И требования к оптимальности хранения данных только растут, появляются новые форматы высокой четкости изображений и видео, а развитие пропускной способности каналов связи не всегда соответствуют требованиям к скорости передачи информации. Например, на современных web сайтах от 30% до 70% размера страницы занимают изображения.

Одним из вариантов решения этих проблем является применение методов компрессии с целью уменьшения занимаемого интересующими данными объема. Чаще всего основной идеей сжатия является уменьшение качества изображения с помощью уменьшения его детализации.

Объектом исследования данной работы являются системы хранений изображений. Предметом исследования являются методы сжатия изображений.

## **Цели и задачи выпускной квалификационной работы**

Целью работы является исследование и анализ методов сжатий изображений с использованием дискретных вейвлет-преобразований (ДВП). В рамках работы изображения разбиваются на независимые компоненты и кодируются независимо арифметическим кодером с учетом связей между соседними элементами (контекстный кодер). Далее предпринимаются попытки повысить качество учета статистических связей соседей.

Основные задачи, решение которых необходимо для достижения поставленной цели:

* проанализировать уже существующие методы компрессии изображений, использующих вейвлет-преобразования;
* проверить эффективность методов сжатия изображений с использованием контекстного кодирования;
* повысить эффективность контекстного кодирования коэффициентов ДВП за счет использования технологий машинного обучения;
* реализовать описанные алгоритмы и разработать программу для их тестирования.

## **Содержание работы**

В первой главе данной работы находится вводная информация, содержащая необходимые теоретические основы, подкреплённые математическими выкладками.

Во второй главе описаны идеи и особенности уже существующих алгоритмов сжатия изображений, выдвигаются гипотезы об их улучшении. Раскрывается реализация упомянутых методов сжатия.

В третьей главе продемонстрированы результаты сравнения эффективности работы рассматриваемых методов, представлены примеры конкретных изображений, восстановленных описанными алгоритмами.

# **Глава 1. Основные понятия**

## **1.1 Полутоновые изображения**

Статические растровые изображения представляют собой двумерный массив чисел (матрицу), описывающую визуальное представление предметов. Элементы этого массива задают цвет точки на изображении и носят название пиксели (от английского pixel ≈ picture element). Пиксель – это наименьшая составляющая изображения. Размеры матрицы пикселей задают разрешение изображения

Все изображения можно подразделить на две группы: с палитрой и без нее. У изображений с палитрой в пикселе хранится число (индекс) в некотором одномерном векторе цветов, называемом палитрой. Чаще всего встречаются палитры из 16 и 256 цветов.

Изображения без палитры бывают в какой-либо системе цветопредставления и *в градациях серого* (полутоновые изображения) [1]. Для последних значение каждого пиксела интерпретируется как яркость соответствующей точки. В данной работе рассматриваются изображения такого вида.

## **1.2 Дискретный сигнал**

Слово *сигнал* означает физический процесс, который отображает некоторую информацию (или сообщение). Математически сигнал, как правило, имеет определенный вид и описывается некоторой функцией . Сигналы бывают аналоговыми, дискретными и цифровыми.

*Аналоговый* сигнал задается кусочно-непрерывной или непрерывной функцией .

*Дискретный* же сигнал задается функцией дискретного аргумента , область определения которой [2]. Отсчетами сигнала принято называть набор значений . Величина , расстояние между соседними отсчётами по оси абсцисс – шаг дискретизации. [2]

Если набор значений, которые принимает дискретный сигнал конечен, то такой сигнал называется *цифровым*. А значения, которые принимает сигнал, называются *уровнями сигнала*.

В реальном мире почти все явления описываются аналоговым сигналом, который не поддается цифровой обработке, так как набор возможных значений сигнала бесконечен (хотя сами значения обычно лежат в конечном интервале). Для исследования процессов сигнал преобразуется в дискретный (дискретизация сигнала). Обычно процесс дискретизации представляет под собой выбор значений сигналов с заданным шагом:

где *, - - период дискретизации*.

Причем период дискретизации обычно выбирается исходят из теоремы Котельникова, которая утверждает, что дискретный сигнал можно восстановить без потерь, если выполняется условие:

где – максимальная частота исходного сигнала в спектральной области [2].

## **1.3 Кратно-масштабный анализ**

Работать с функцией сигнала в исходном виде, как правило, сложно, поэтому достаточно часто используется представление функции в виде линейной комбинации функций из выбранной системы, которую мы обозначим за :

где – *коэффициенты разложения*, – *функции разложения*. Отметим, что сумма может быть как конечной, так и бесконечной.

Если для любой заданной функции существует единственный способ представления в виде линейной комбинации функций разложения, то система функций называются *базисом,* а функции - *базисные функции*.

Замыкание линейной оболочки функции является множеством функций, которые можно представить в виде линейной комбинации базисных функции, и обозначается:

Последовательность подпространств {𝑉𝑚} ⊂ , 𝑚 ∈ 𝑍, образует кратно-масштабный анализ, если обладает следующими свойствами [2]:

1. Подпространства вложены, ∀𝑚 ∈ 𝑍: 𝑉𝑚 ⊂ 𝑉𝑚+1.
2. Если функция 𝑓(𝑥) ∈ 𝑉𝑚, то 𝑓(2𝑥) ∈ 𝑉𝑚+1 и наоборот.
3. Существует некоторая функция 𝜙(𝑥) ∈ 𝑉0, целочисленные сдвиги которой {𝜙(𝑥 − 𝑛)}𝑛∈𝑍 ⊂ 𝑉0 образуют ортонормированный базис подпространства 𝑉0 . Такая функция 𝜙(𝑥) называется масштабирующей.
4. Для всех подпространств есть единственный общий элемент – нулевой, ∩ 𝑉𝑚 = {0}, 𝑚 ∈ 𝑍.
5. Замыкание множества всех подпространств является пространством 𝐿2 (𝑅): ⋃ = 𝐿2 (𝑅), 𝑚 ∈ 𝑍.

## **1.4 Масштабирующие функции**

Пусть задана функция – квадратично-интегрируемая, вещественная. Тогда рассмотрим систему вида:

*(1)*

где – целые числа.

Система задает систему функций разложения, которая включает в себя двоичные сжатия и растяжения, а также смещений вдоль аргумента. носит название *масштабирующей функции* [12]*.*

Если функция хорошо подобрана, то пространство квадратично-интегрируемых функций и замыкание линейной оболочки системы функций разложения совпадают.

Рассмотрим подпространства кратно масштабного анализа {𝑉𝑚}. Будем обозначать ортогональное дополнение подпространства до как . Тогда, расширив наше обозначение для любого :

Если , то:

С учетом пятого свойства кратно-масштабного анализа получим:

Основное утверждение кратно-масштабного анализа гласит: ортонормированный базис в подпространстве 𝑊m задается множество функций , причем функция ѱ(𝑥) ∈ 𝑊0 существует для любой масштабирующей функции. Функцию, для которой выполняется данное утверждение, принято называть материнским *вейвлетом*.

Для любой функции 𝑓(𝑥)) ∈ 𝐿2 (𝑅) существует разложение по ортогональному базису вейвлетов [2]:

Где 𝑓M = 𝐴M(𝑓), 𝑐m.n (𝑓) = 〈𝑓, ѱm,n 〉. Таким образом, материнский вейвлет можно представить с помощью базиса :

где . Уравнение называется *масштабирующим* для вейвлетов.

## **1.4 Вейвлет-преобразование**

Вейвлет-преобразование разбивает данные или функции на составляющее с различными частотами, каждая из которых позже рассматривается отдельно с разрешением, подходящим по масштабу [3].

Для одномерного сигнала вейвлет-преобразование представляет собой обобщенный ряд Фурье в системе базисных функций, полученных из материнского вейвлета. Тогда вещественную функцию, описывающую одномерный сигнал можно записать в виде:

Здесь j0 – произвольный начальный масштаб, – коэффициенты приближений (получили такое название благодаря тому, что задают приближение исходной функции в масштабе ), – коэффициент деталей (носит такое название, так как увеличивает детализацию, добавляя приближенную сумму вейвлетов).

В случае, когда – дискретная функция, набор коэффициентов называется дискретным вейвлет-преобрзованием. [2]

## **1.5 Квантование**

Квантованием называется разбиение возможных значений сигнала на конечное число элементов (уровней) и округление значений сигнала до этих значений. В обработке изображений квантование необходимо для сокращения объема информации, необходимой для передачи сигнала. Так как во время квантования происходит округление значений исходного сигнала, этот процесс связан с потерей информации.

Проквантованный сигнал получается из исходного сигнала по формуле:

где –шаг квантования, а оператор означает взятие целой части числа.

Возможные значения сигнала, полученные после его квантования, называются *уровнями квантования.*

Из формулы видно, что сигнал, который попадает в интервал приравнивается к 0. Этот интервал называется *мертвой зоной*.

Частный случай квантования – *равномерное квантование*, характерной особенностью которого является постоянство шага квантования .

Если значения сигнала распределены неравномерно, например, значения сигнала звука, то шаг квантования в области, где значения сигнала имеют более высокую плотность распределения, выбирается более мелким. Таким образом повышается точность восстановленного сигнала, а квантование носит название *неравномерного квантования*.

В данной работе будет рассматриваться равномерное квантование, так как оно хорошо зарекомендовало себя в области обработки изображений.

Чтобы получить исходный сигнал необходимо выполнить *деквантование*. Деквантование происходит по формуле:

Графически соотношение проквантованного сигнала и исходного выглядит следующим образом:



Рисунок 1. Квантование сигнала

## **1.6 Арифметическое кодирование**

Арифметическим кодированием называется отображение слов алфавита с заданным распределение вероятностей на множество двоичных слов [4].

Основанная идея арифметического кодирования заключается в следующем: слова алфавита сортируются по возрастанию вероятности их появления, пропорционально этим вероятностям делится интервал от 0 до 1. Каждая часть такого интервала соответствует двоичному представлению слова исходного алфавита. Таким образом, для наиболее часто встречающихся слов будут использоваться наиболее компактные представления.

Для того, чтобы идея арифметического кодирования стала более наглядной, покажем её на примере.

Допустим, что алфавит нашего источника сигнала состоит из четырех символов: . Причем вероятности появления символов известны и записаны в таблицу:

|  |  |  |
| --- | --- | --- |
| Символ | Вероятность появления | Начальный подынтервал |
| a | 0.2 | [0, 0.2) |
| b | 0.4 | [0.2, 0.6) |
| c | 0.2 | [0.6, 0.8) |
| d | 0.2 | [0.8, 1) |

Таблица 1. Распределение символов по исходному интервалу

Мы будем кодировать последовательность символов: .

В начале работы алгоритма сообщение распределяется по всему интервалу . Интервал распределяется между символами в соответствии с частотой их появления. В таблице 1 видно, что символ , частота появления которого в два раза выше частоты появления остальных символов, занял в два раза больший подынтервал, чем остальные символы. Причем остальные символы заняли подынтревалы одинакового размера, так как имеют одинаковую частоту появления. Далее мы начинаем кодировать сообщение. Первым символом сообщения является символ . Поэтому далее мы будем рассматривать соответствующий ему интервал вместо исходного . Так как рассматриваемый интервал изменился, необходимо снова перераспределить подынтервалы между символами:

|  |  |  |
| --- | --- | --- |
| Символ | Вероятность появления | Начальный подынтервал |
| a | 0.2 | [0, 0.04) |
| b | 0.4 | [0.04, 0.12) |
| c | 0.2 | [0.12, 0.16) |
| d | 0.2 | [0.16, 0.2) |

Таблица 2. Распределение символов по интервалу

Далее процесс повторяется для каждого символа сообщения. В конечном итоге мы получим подынтервал . Любое произвольно взятое число из полученного интервала будет кодировать наше сообщение. Допустим, мы выбрали число 0.0627. Тогда последовательность из пяти символов нам удалось закодировать с помощью трех цифр.

## **1.7 Энтропия**

Рассмотрим модель дискретного источника сообщений , количество состояний которого конечно. Состояния источника изменяются в заданные моменты времени. Положим, что вероятность перехода источника в состояние задана и равна .

Физический смысл состояния можно понимать как случайный выбор символа из алфавита . Для того, чтобы измерить количество информации, которое приходится на один символ сообщения, введем меру количества информации.

В конце 1940-х годов Клод Шеннон предложил свою величину в качестве меры количества информации:

где – количество символов в алфавите источника, – вероятность появления k-ого символа, а основание логарифма не играет важной роли. Данная величина называется *энтропией*.

Физический смысл энтропии можно описать, как среднее количество информации, которое содержит один символ источника информации. [2] При чем в цифровой обработке изображений основание логарифма принято брать двоичным, это связано с тем, что информация измеряется в битах.

## **1.7 Машинное обучение**

Машинное обучение представляет собой науку (и искусство) программирования компьютеров для того, чтобы они могли обучаться на основе данных. Теория машинного обучения изучает способы решения задач на основе схожих задач, решаемых в процессе обучения.

Методы машинного обучения используют математический аппарат линейной алгебры, теории вероятности и математической статистики, численных методов и других разделов математики. [13]

Алгоритмы машинного обучения делятся на четыре вида: обучение с учителем, обучение без учителя, частичное обучение, обучение с подкреплением. [8]

При обучении с учителем данные, которые планируется использовать в системе, поставляются с метками (желательные решения, ответы). Числовые характеристики, используемые для нахождения метки, называются признаками. Матрицу, строки которой состоят из набора признаков, соответствующих одной целевой метке, будем обозначать за . Вектор-столбец желательных решений - . Набор данных с признаками и метками называется обучающей выборкой.

Как пример обучения с учителем можно рассмотреть задачу регрессии. Допускается, что между признаками и ответами есть зависимость, которая нам неизвестна:

Далее ставится задача о нахождении приближения функции с помощью аппроксимации её другими функциями.

Как пример применения решения задачи регрессии может служить задача предсказания цены автомобиля: необходимо спрогнозировать цену автомобиля в зависимости от его пробега, модели, возраста. Признаками будут являться величина пробега, номер модель и возраст автомобиля (например, в годах), а ответами в данной задаче будут значения стоимости автомобиля.

Еще одной типичной задачей обучения с учителем является задача классификации. Классификация отличается от регрессии конечностью множества возможных значений метки. Примером такой задачи может служить распознавание рукописных букв, в случае которого ответами будет конечное множество .

Обучение без учителя отличается от обучения с учителем тем, что известны признаки, но значения целевой метки, соответствующие им, не заданы. В таком случае ставится задача о нахождении зависимостей между объектами. Одним из наиболее часто встречающихся примеров обучения без учителя является *кластеризация*. Цель кластеризации - распределить объекты из обучающей выборки на множества (кластеры) таким образом, чтобы эти множества не пересекались, а объекты, попавшие в один кластер, были схожи по некоторому признаку.

Частичное обучение представляет собой объединение обучения с учителем и обучения без учителя. В случае частичного обучения используются как данные с ответами, так и данные с неизвестными значениям целевой метки.

Последним видом алгоритмов машинного обучения является обучение с подкреплением. Данный метод основан на взаимодействии алгоритма с окружающей средой.

В данной работе используется два алгоритма обучения с учителем, а именно методы решения задачи регрессии. Рассмотрим их более подробно.

## **1.8 Квадратичная регрессия**

Пусть задана обучающая выборка , где , при .

Задача квадратичной регрессии состоит в нахождении функции, которая лучше всего приближает значение [9]:

где – матрицы весовых коэффициентов, – свободный член.

Для того, чтобы найти матрицы мы составим функцию вида

которую назовем функцией потерь. Можно взять функции другого вида, но мы остановимся на этой.

Теперь задача регрессии может быть записана в виде задачи минимизации функции :

Таким образом, минимизируя ведённую нами функцию потерь, мы сможем вычислить значения такие, при которых выражение будет «предсказывать» величину .

## **1.9 Случайный лес**

Для начала введем понятие дерева решений, для этого опишем алгоритм его построения.



Рисунок 2. Решающее дерево

Обучающая выборка разбивается на две части и помещается в узлы дерева таким образом, чтобы попавшие в разные сегменты выборки данные максимально различались по зависимой переменной (какому-нибудь признаку). Далее для каждого узла вычисляется среднее значение признака и выносится решение – прогноз среднего значения целевой переменной. Полученные в результате узлы аналогично разбиваются на две части. В итоге формируется иерархическая структура, содержащая в себе зависимость между признаками и метками, которая называется деревом. Прогнозом целевой метки будут листья полученного древа. Приближение предсказания к целевой метке достигается за счёт регулирования порогов, по которым узлы разбиваются на две части. [11]

Идея случайного леса заключается в том, чтобы из исходного набора данных извлечь случайные выборки (с возвращением), на каждой такой подвыборке обучить решающее дерево. Позже, чтобы получить прогноз леса, собираются ответы всех деревьев и усредняются.

## **1.10 Заключение к главе**

Подведем итоги. Целью данной выпускной квалификационной работы является анализ эффективности существующих алгоритмов компрессии изображений на основе дискретных вейвлет-преобразований и исследование их модификаций. В рамках работы четырнадцать тренировочных изображений будут подвергнуты дискретному четырёхуровнему вейвлет-преобразованию, затем полученные вейвлет-коэффициенты будут проквантованы и после чего полученный набор данных будет кодироваться арифметическим кодером, имеющим три реализации. В каждой из реализации будут учитываться статистические связи между проквантованными вейвлет-коэффициентами (контекстное кодирование). В первой версии кодера зависимости между элементами будут аппроксимироваться простым линейным многочленом. Во второй версии – задачу определения связей между вейвлет-коэффициентами мы будем решать с помощью квадратичной регрессии. А в третей реализации для учета статистических зависимостей между элементами мы идем использовать случайный лес. После того, как тренировочные изображения будут закодированы и по ним будут обучены алгоритмы, учитывающие статистические зависимости между элементами, эффективность работы алгоритмов будет анализироваться на трех тестовых изображениях.

Для достижения поставленной цели необходимо решить следующие задачи:

1. проанализировать существующие методы компрессии изображений с применением дискретных вейвлет преобразований
2. исследовать эффективность работы алгоритмов сжатия изображений с использованием контекстного кодирования и методов машинного обучения
3. разработать программу для тестирования исследуемых алгоритмов, подбора оптимальных параметров, оценки эффективности их работы
4. проанализировать эффективность работы предлагаемых методов компрессии на примере тестовых изображений

# **Глава 2. Обзор существующих методов сжатия изображений и разбор предлагаемого алгоритма**

## **2.1 Обзор существующих методов сжатия изображений**

Все методы сжатия изображений можно разделить на два вида: сжатие без потерь и сжатие с потерями. Сжатие с потерями отличается от сжатия без потерь тем, что в первом случае при восстановлении теряется часть исходной информации.

В данной работе рассматриваются методы сжатия с потерями.

Одним из источников уменьшения объема, занимаемого информацией, является более оптимальное представление повторяющихся последовательностей. Основная идея сжатия данных состоит в том, чтобы вместо дублирования последовательности сохранить её один раз, а далее вместо повторяющихся наборов символов сохранять ссылки на обработанные ранее аналогичные фрагменты последовательности.

Вторым источником сокращения размера изображения является потеря части данных. Несмотря на то, что качество восстановленной графической информации падает, методы сжатия с потерями широко распространены. Их популярность обусловлена тем, что при небольшой потере данных, которой можно пренебречь, алгоритмы позволяют получить значительный выигрыш в уменьшении размера изображений. Примером одним из самых популярных представителей данного класса алгоритмов является *JPEG*.

### **2.1.1 JPEG**

JPEG (Joint Photographic Experts Group) является одним из наиболее популярных форматов хранения фотографий и изображений фотографического качества. Метод особенно хорошо справляется с сжатием изображений, в котором отсутствуют резкие переходи цвета или яркости. Поэтому данный алгоритм хорошо показывает себя при работе с фотографиями, однако плохо пригоден для сжатия текстовой графики.

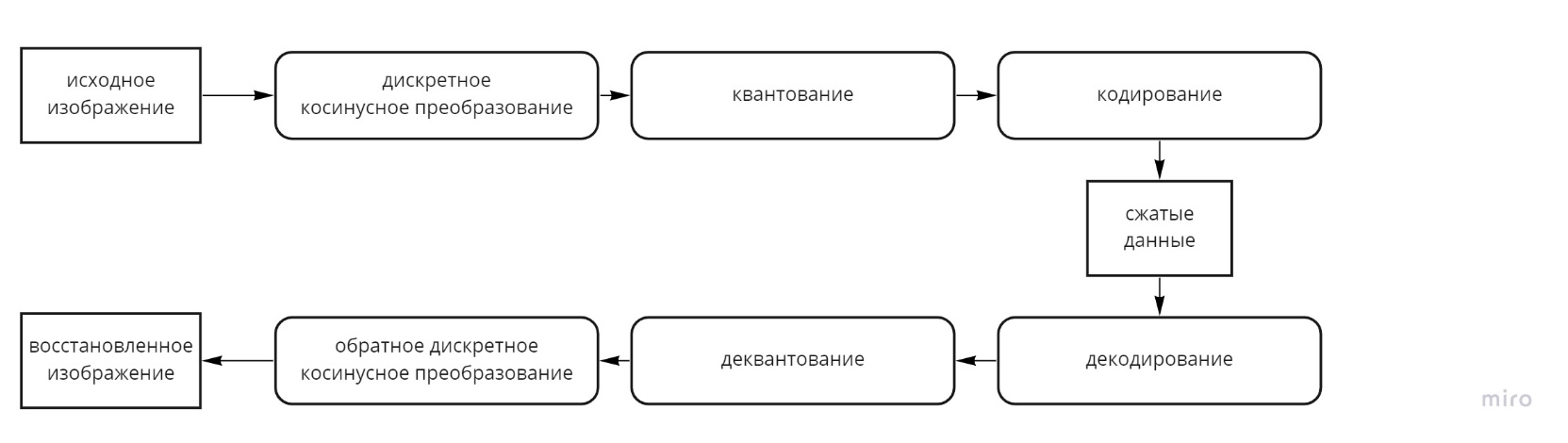
Алгоритм компрессии изображения JPEG делится на шесть этапов. Схематично эти этапы изображены на рисунке:

Рисунок 3. Алгоритм JPEG

Первый этап заключается в замене цветовой модели изображения на модель , которая подразумевает разделение цвета и яркости. Далее каждый канал будет обрабатываться отдельно, что позволит применить более оптимальное сжатие к каждому из них. Переход к новой цветовой модели осуществляется по формулам:

Второй этап включает в себяcубдискретизацию. Выбираются группы пикселей 2x2 в каналах и . Для каждого такого квадрата из 16 пикслей рассчитывается среднее значение, которым заменяется группа. Канал на данном этапе не обрабатывается, так как его изменения наиболее заметны человеческому глазу, из-за чего при субдискретизации компонента потеря качества изображения становится сильно заметна.

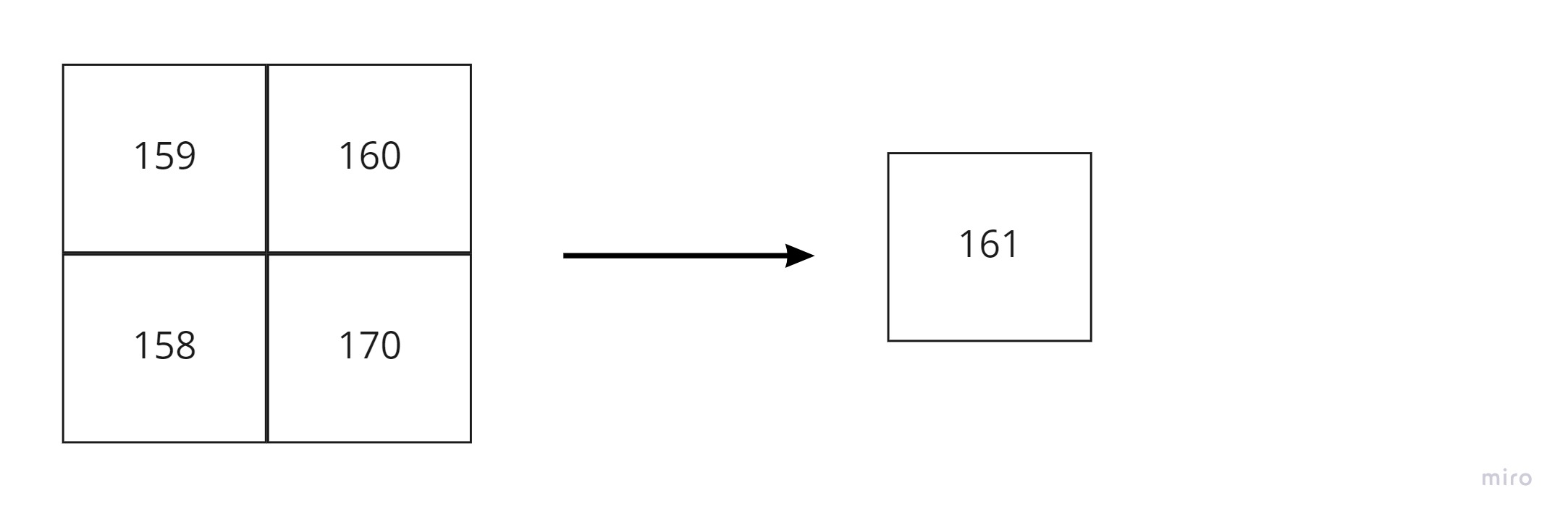


Рисунок 4. Пример субдискретизации

На третьем этапе происходит двумерное дискретное косинусное преобразование (ДКП), сгруппированных в квадраты 8x8 элементов. При таком выборе размера группировки пикселей с одной стороны уже хорошо показывают себя декоррелирующие свойства дискретного косинусного преобразования, которые позволяют избавиться от избыточности в данных, а с другой стороны – область достаточно мала, чтобы характеристики изображения можно считать локально стационарными. В результате ДКП мы переходим в частотную область. Данных переход осуществляется по формулам:

где .[2]

Следующий этап – скалярное квантование коэффициентов, полученных на предыдущем шаге. Этот этап необходим для уменьшения набора значений, элементов спектра. Сигнал квантуется по формуле:

где – шаг квантования, который определяется по матрице квантования, – операция взятия целой части. Стандарт JPEG не устанавливает матрицу квантования, но предоставляет рекомендуемый набор значний. От выбора матрицы квантования зависит степень сжатия изображения. Большая часть потери информации происходит на этапе квантования. С другой стороны, именно этот этап позволяет значительно сократить занимаемый данными объем. Изменение матрицы квантования позволяет добиться компромисса между потерей качества и степенью сжатия изображения. Важной особенностью данного этапа является зануление большого количества значений.

Пятый этап. После ДКП и квантования двумерный набор чисел, полученный на выходе, зигзагообразно обходится и считывается в линейную последовательность.

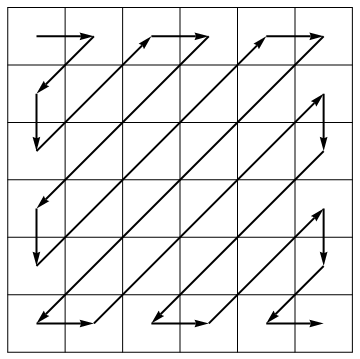


Рисунок 5. Порядок считывание ДКП-коэффициентов

Данный способ чтения обусловлен тем, что в полученная одномерная последовательность содержит в себе коэффициенты ДКП в порядке возрастания частот спектра, а нулевые компоненты оказываются сгруппированы в конце последовательности. Таким образом коэффициенты, отвечающие за более крупные детали, заметные глазу, располагаются раньше элементов, отвечающих за мелкие детали. После чего последовательность кодируется с помощью арифметического кодирования.

Для того, чтобы по сжатым данных получить исходное изображение, этапы алгоритма JPEG запускаются в обратном порядке: декодирование, деквантование, обратное дискретное косинусное преобразование. После того, как изображение восстановлено, к нему прибавляется небольшое количество шума для того, чтобы «сгладить» перепады между значениями граничных пикселей.

Методы сжатия изображений активно развиваются и с 2001 года принят расширенный стандарт алгоритма JPEG, который носит название JPEG-2000. [2]

### **2.1.2 JPEG-2000**

Несмотря на то, что JPEG широко используется в большинстве приложений, ориентированных на работу с изображениями, он имеет свои недостатки, а именно: трудности в обработке изображений с резкими перепадами (например, текст или изображения, полученные с помощью компьютерной графики), грубое квантование коэффициентов, отвечающих за низкие частоты.

Компрессия изображений алгоритма JPEG-2000 основано на использовании вейвле-преобразований и контекстом кодировании вейвлет-коэффициентов. Новый стандарт позволяет лучше сохранить качество изображения при высокой степени сжатия. Особенности представления данных позволили плавно восстанавливать изображения, не дожидаясь чтения потока данных до конца. Стандарт стал более устойчив к помехам, которые могут поваляться во время передачи данных. Появилась возможность сжатия без потерь. [14]

Можно выделить четыре основных этапов работы алгоритма JPEG-2000:

1. Переход в оптимальную цветовую модель
2. Дискретное вейвлет-преобразование (ДВП)
3. Квантование
4. Кодирование

Первый этап стандарта JPEG-2000 аналогичен первому этапу алгоритма JPEG.С помощью соответствующих преобразований осуществляется переход в цветовую модель .

На втором этапе происходит дискретное вейвлет-преобразование сигнала. Именно этот этап является отличительной особенностью стандарта JPEG-2000. В результате дискретного вейлвет-преобразования над каждым каналом , , получается отфильтрованный высокочастотными и низкочастотными фильтрами по строкам и столбцам двумерного сигнала. После первой итерации такой обработки на выходе получается изображение, разбитое на четыре компонента (*саббэнд*а), которые являются уменьшенными версиями исходного изображения. Каждый из саббэндов передает часть информации первоначального двумерного сигнала. Пример ДВП приведен на рисунке.



Рисунок 6. Вейвлет-преобразование

Исходное изображение (Рисунок 6. а) разбивается на две части, после чего подвергается фильтрации (Рисунок 6. в): низкочастотной (НЧ), высокочастотной (ВЧ). Левый верхний саббэнд полученный, после первой итерации преобразования и дважды прошедший низкочастотную фильтрацию, несет в себе наибольшее количество информации и является уменьшенной версией исходного изображения. Левый нижний квадрат более точно передает горизонтальные составляющие. А правый верхний – вертикальные. Саббэенды, полученные в результате высокочастотной фильтрации по строкам и столбцам, передают наименьшее количество информации. Описанные особенности хорошо заметны на изображении. Приведем пример двухуровневого вейлвет-преобразования.



Рисунок 7. Пример двухуровневого вейвлет-преобразования

На рисунке 7 хорошо видны описанные зависимости. Например, что саббэнд в левом верхнем углу более точно передает вертикальные составляющие изображения.

Третий этап – этап квантования. В JPEG-2000 применяется квантование «с мертвой зоной», описанное в главе 1.5. В отличие от стандарта JPEG – шаг квантования в алгоритме JPEG-2000 необязательно является целым числом.

На четвертом этапе происходит кодирование проквантованных коэффициентов ДВП. Арифметическое кодирование осуществляется с помощью многомодельного кодера с памятью (то есть выбор модели для кодирования очередного символа зависит от предыдущих символов).

Аналогично стандарту JPEG для того, чтобы восстановить исходное изображение по сжатым данных, необходимо пройти этапы работы алгоритма JPEG-2000 в обратном порядке.

В результате работы описанного алгоритма работы JPEG-200 восстановленные изображения получаются более качественными и «гладкими», чем в JPEG. При этом размер сжатого новым стандартом изображения значительно меньше, чем у его предшественника.

## **2.2 Предлагаемые методы сжатия изображений**

### **2.2.1 Общая схема метода компрессии изображение на основе дискретного вейвлет-преобразования**

Общую схему метода компрессии изображение на основе ДВП можно представить следующим образом:



Рисунок 8.Общая схема метода компрессии изображение на основе ДВП

На рисунке обозначены основные этапы, которые проходит изображения во время сжатия и восстановления. Исходной изображение будет подвергаться четырехуровневому дискретному вейвлет-преобразованию. После чего, полученные вейвелет-коэффициенты проходят через этап квантования с мертвой зоной. Проквантованные коэффициенты поступают на вход арифметического контекстного кодера, описанного в главе 2.2.2. В результате на выходе получаются сжатые данные, восстановления изображения по котором осуществляется с помощью обработки данных в обратном порядке: декодирование, деквантование, обратное четырехуровневое ДВП. Блоки, выделенные цветом, предлагается улучшить в рамках данной работы.

Общая метода компрессии реализована на языке программирования Python. На нулевом этапе обработки изображение считывается в память с помощью библиотеки для работы с изображениями OpenCV [15]. Далее изображение, загруженное в память, проходит через четырёхуровневое ДВП, реализованное на базе библиотеки PyWavelets [16]. После этого полученные вейвлет-коэффициенты квантуются с помощью кантователя с мертвой зоной, который был описан в главе 1.5. Последний этап работы метода – контекстное кодирование, алгоритм работы которого рассмотрен в следующей главе. Этап контекстного кодирования предполагается улучшить в рамках данной работы.

### **2.2.2 Контекстно кодирование**

Одной из модификаций стандартного арифметического кодирования, дающего больший выигрыш в сжатии данных, является контекстное кодирование, идея которого была заимствована в [5]. В контекстном арифметическом кодировании используется несколько моделей (гистограмм распределения частот появления символов), выбор которой для очередного кодируемого символа будет зависеть от его соседних элементов.

В данной работе используется четыре модели для кодирования диагональных саббэндов и четыре модели для кодирования оставшихся саббэндов. Для каждого вида саббэндов обход вейвлет-коэффициентов выбран таким образом, чтобы статические связи между элементами проявлялись как можно больше. Например, саббэнд полученный в результате двухуровневого вейлвет-преобразования и находящийся в правом верхнем углу, передает вертикальные составляющие изображения, поэтому данный саббэнд мы будем кодировать, проходя элементы по вертикали. На рисунке более наглядно продемонстрированы особенности кодирования компонентов.



Рисунок 9. Направления обхода саббэндов во время кодирования

На рисунке 9 желтыми отрезками обозначены направления обхода саббэндов во время их кодирования.

Но на этом учет статистических связей между элементами не заканчивается. Как уже говорилось, для каждого очередного символа модель, которой он кодируется, выбирается по его соседним элементам. Для этого по соседним элементам рассчитывается величина прогноза , в зависимости от которой выбирается модель.

Выбор соседних элементов для каждого саббэнда индивидуален. Обозначим коэффициенты соседних элементов для очередного символа за . Для саббэндов, передающих горизонтальные составляющие изображения, выбор соседних элементов будем осуществлять следующим образом:

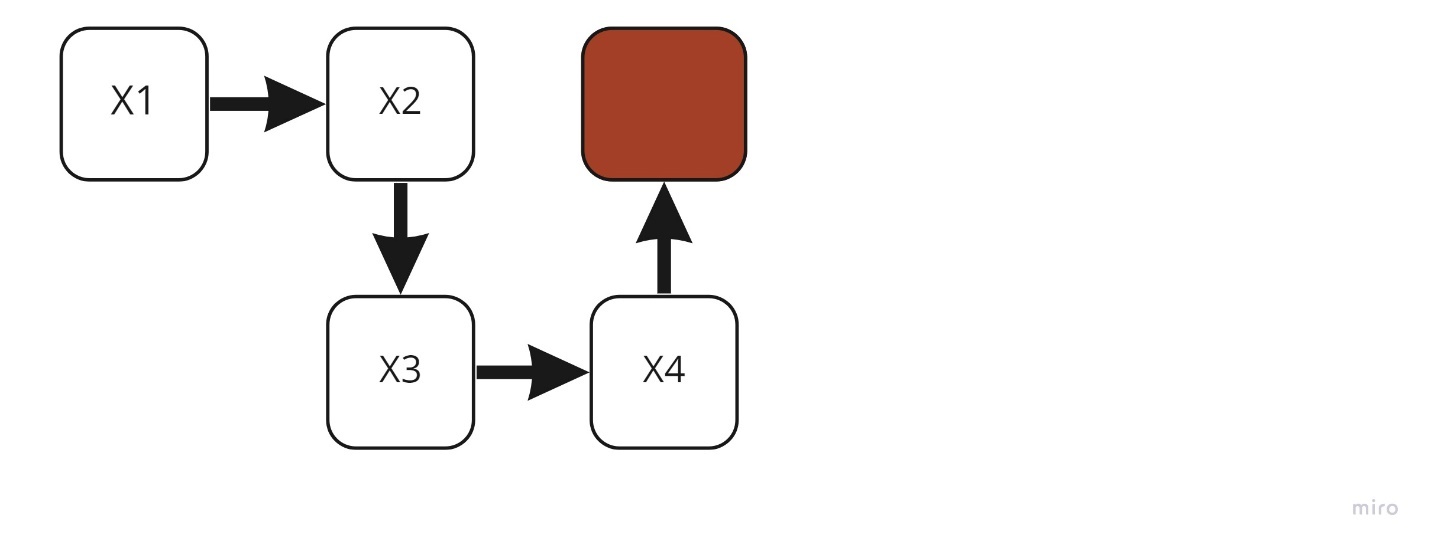


Рисунок 10. Выбор соседних элементов в горизонтальном направлении

На рисунке 10 стрелками изображено направление обхода саббэнда, а цветом выделен кодируемый символ.

Аналогично покажем способ выбора соседних элементов для саббэнда, передающего вертикальные составляющие изображения.

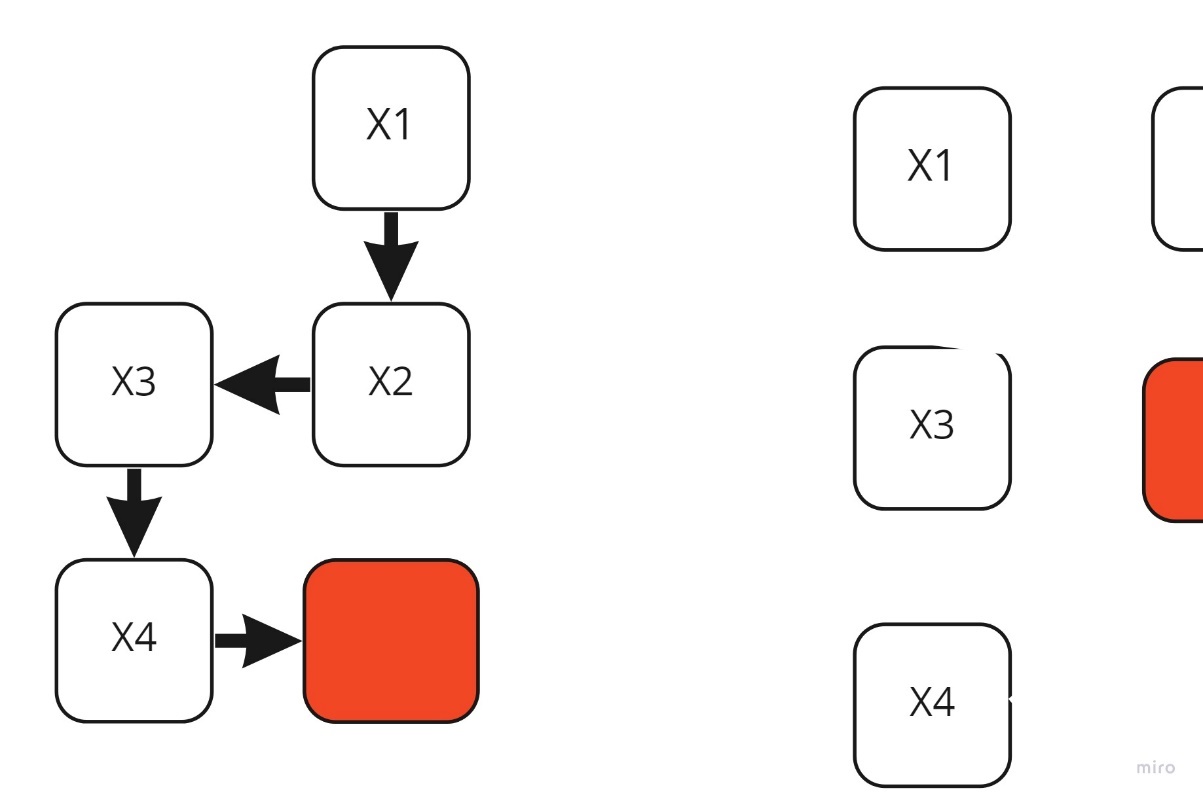


Рисунок 11. Выбор соседних элементов в вертикальном направлении

Для саббэндов, в которых статистические связи между элементами проявляются наиболее ярко при обходе «змейкой», соседние эелменты выбираются следующим образом:

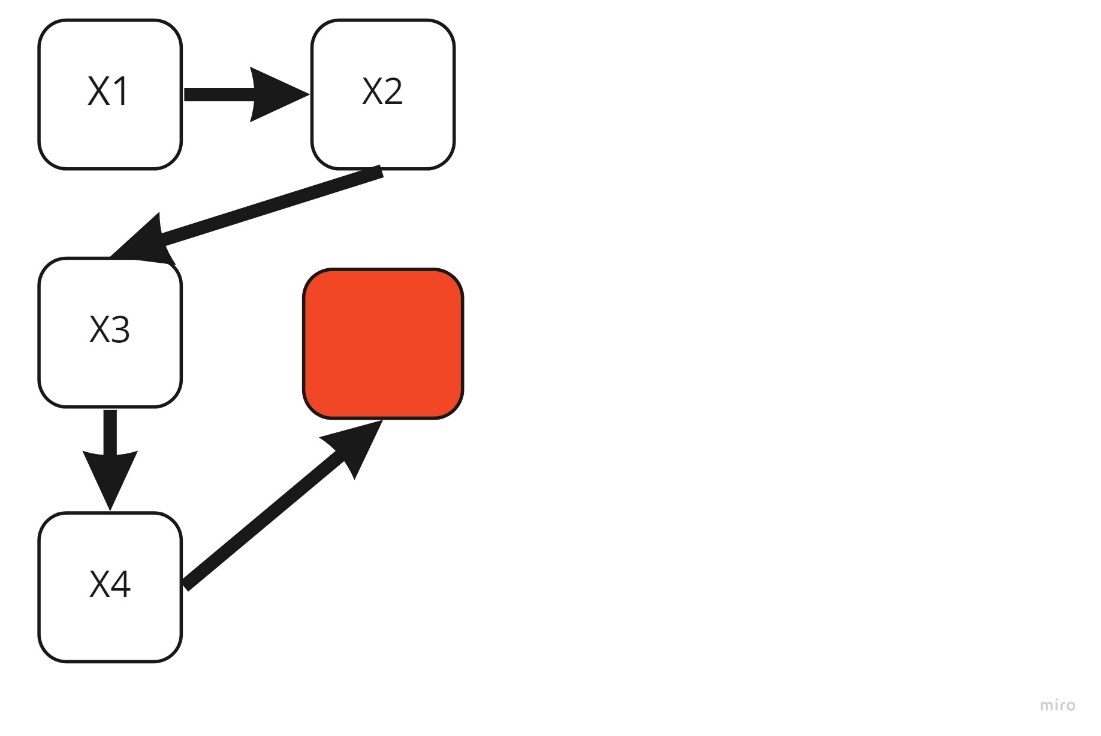


Рисунок 12. Выбор соседних элементов в диагональном направлении

Для большей наглядности покажем выбор соседних элементов и направления обходов на саббэндов на конкретном примере.

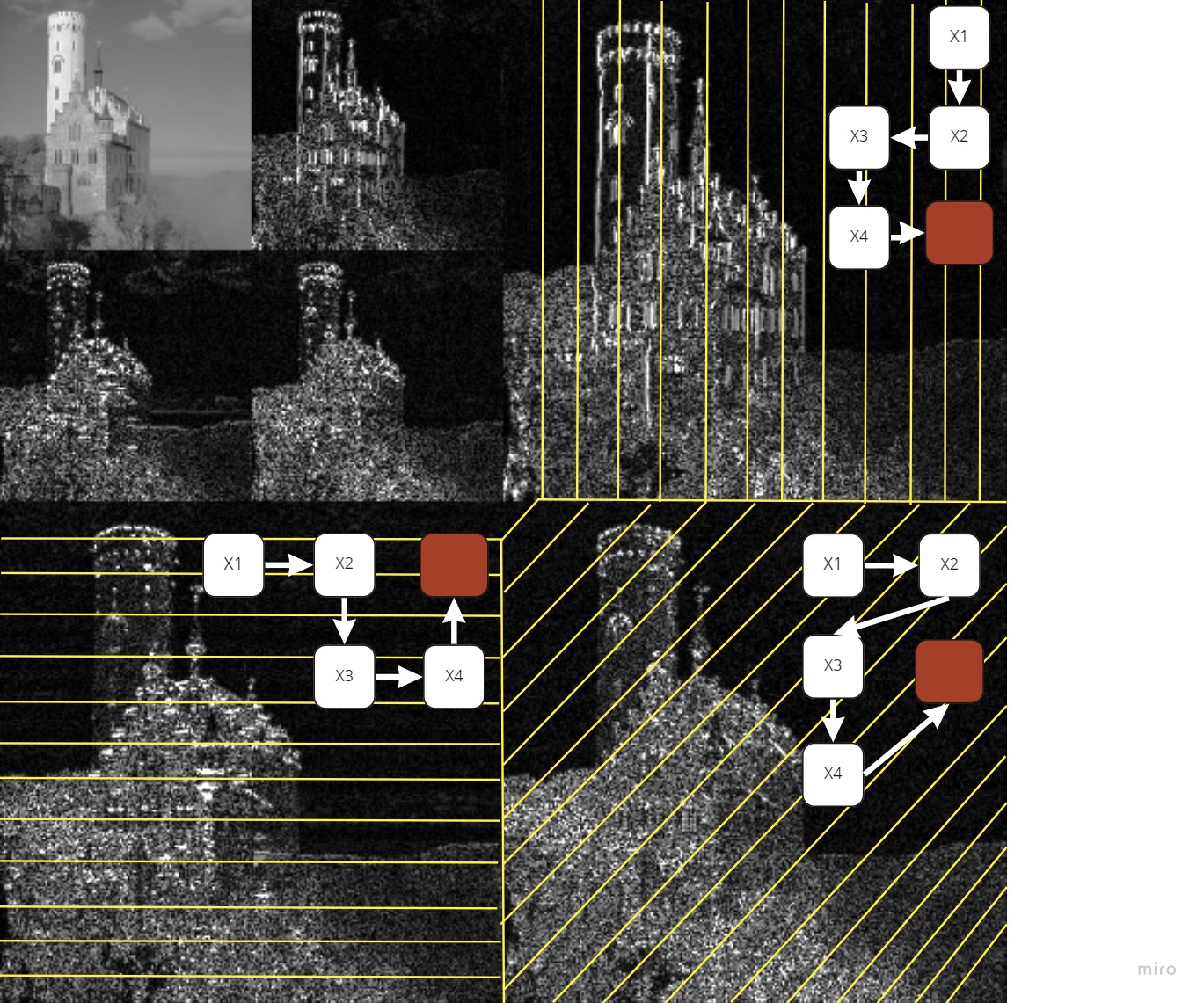


Рисунок 13. Пример обхода саббэндов с выбором соседних элементов

После того, как соседи выбраны, необходимо рассчитать величину прогноза . Для этого используется формула:

Далее по заранее эмпирически выбранным пороговым значениям выбирается одна из четырех моделей, используемых для кодирования саббэнда. Выбор осуществляется следующим образом:

1. Если , то для кодирования символа выбирается первая модель.
2. Если , то для кодирования символа выбирается вторая модель.
3. Если , то для кодирования символа выбирается третья модель.
4. Если , то для кодирования символа выбирается четвертая модель.

Таким образом, арифметическое кодирование учитывает статические связи между элементами с целью их более оптимальной обработки.

### **2.2.3 Предлагаемые методы улучшения контекстного кодирования**

В ходе работы было предположено, что развитие идеи контекстного кодирования способно улучшить характеристики алгоритма компрессии изображений. Контекстное кодирование показывало хорошие результаты относительно обычного арифметического кодирования и дальнейшее развитие алгоритма видится в более точном учете статистических связей между вейвлет-коэффициентами. Другими словами, встает задача о более точном предсказании модели для кодирования очередного символа по его соседним элементам, с помощью которой удастся понизить избыточность данных в сжатом изображении.

В базовом алгоритме прогноз вычисляется как линейная комбинация модулей значений вейвлет-коэффициентов. Было сделано предположение, что статистические связи между элементами носят более сложный характер и для их описания необходимы более сложные функции. В частности, были применены методы машинного обучения.

Кроме того, в базовом алгоритме использовался *адаптивный кодер*, модели-гистограммы частот появления символов которого обновлялись по мере кодирования данных. Этот подход был сохранен в улучшенной версии алгоритма.

В классическом алгоритме вероятностные модели на начальном этапе кодирования инициализируется нулями. В предлагаемом методе адаптивным кодером было закодировано четырнадцать полутоновых изображений, которые мы будем называть *тренировочными*. Качество работы предлагаемого алгоритма сжатия оценивалось на трех изображениях, которые не присутствовали в тренировочном наборе, эти три изображения мы будем называть *тестовыми*. Причем после кодирования очередного изображения вероятностные модели сохранялись и использовались для инициализации гистограмм частот появления символов перед обработкой следующего изображения. После того, как гистограммы обучены на всех 14 тренировочных изображениях, полученные вероятностные модели записываются в файл. Далее, во время кодирования тестовых изображений, гистограммы инициализируются значениями из файла. Таким образом, вероятностные модели кодера – предобучены.

Как же заставить алгоритм машинного обучения предсказывать вероятностную модель для кодирования очередного элемента таким образом, чтобы битовые затраты уменьшились?

### **2.2.4 Применение методов машинного обучения**

Во время кодирования очередного символа мы располагаем информацией о значениях его соседних элементов, именно эти значения и будут признаками в нашей выборке.

Пронумеруем вероятностные модели арифметического кодера, используемые для кодирования одного саббэнда, обозначив их за . Наша цель – использовать такие вероятностные модели для кодирования символов, чтобы битовые затраты были минимальны. Поэтому в качестве метки (целевой величины) для каждого символа будем использовать номер вероятностной модели, кодирующей символ с минимальными битовыми затратами.

Таким образом, матрица признаков будет иметь вид:

где – значение -го соседнего элемента для -го символа, N – количество кодируемых элементов. Схема выбора соседних элементов для каждого саббэнда приведена в главе 2.2.2.

Составим матрицу меток, в которой будут содержаться номера вероятностных моделей для кодирования советующего символа:

где – номер вероятностной модели, которая кодирует -ый символ с минимальными битовыми затратами.

После того, как тренировочная выборка составлена, осталось обучить модель, которая будет аппроксимировать зависимость таким образом, что

В качестве функции ошибки, которая будет штрафовать модель за отклонение значения прогноза от целевой функции, мы будем использовать бинарную энтропию, так как нашей задачей является уменьшение битовых затрат во время кодирования.

Для предсказания значений по величинам мы будем использовать три алгоритма:

1. Квадратичная регрессия
2. Случайный лес из 10 решающих деревьев глубины 4
3. Случайный лес из 100 решающих деревьев глубины 1

Обучающая выборка была составлена на четырнадцати тестовых изображениях.

## **2.2.5 Заключение к главе**

В разделах 2.1.1 и 2.1.2 были рассмотрены алгоритмы сжатия изображений, имеющие широкую известность: стандарты JPEG и JPEG-2000. В разделах 2.2.1-2.2.2 была пояснена общая схема компрессии изображений на основе дискретных вейвлет-преобразований. В разделах 2.2.3-2.2.4 были предложены методы улучшения общего алгоритма сжатия изображений, описанного ранее. В главе 3 будет проведено сравнение эффективности всех перечисленных алгоритмов.

# **Глава 3. Результаты экспериментов**

## **Метрики эффективности алгоритмов компрессии изображений**

Так как мы будем использовать методы сжатия с потерями, одной из метрик эффективности методов будет точность, с которой алгоритм восстанавливает исходное изображение. В нашем случае такой метрикой будет *PSNR* (*peak signal-to-noise ratio*) – пиковое отношение сигнала к шуму, величина которой вычисляется по формуле:

где ε2 (здесь – размеры изображения), – максимально значение, принимаемое пикселем изображения. PSNR измеряются в дБ. Значения PSNR будут откладываться по оси ординат.

По оси абсцисс будут расположены значения величины *bpp* (bits per pixel), которая представляет собой количество бит, используемых для хранения одного пикселя в сжатом изображении.

## **Результаты проведенных экспериментов**

Проанализируем описанные ранее методы сжатия на трёх изображениях:



Рисунок 14. Тестовые изображения: a – «Lena», б – «Barbara», в – «Goldhill»

Сначала сравним качество работы базового алгоритма с стандартами JPEG и JPEG-2000:

Видно, что базовый алгоритм уже превосходит JPEG, но все еще далёк от JPEG-2000.

Добавим на графики результаты работы алгоритмов с применением квадратичной регрессии и случайного леса, состоящего из 10 решающих деревьев глубины 4.

Заметим, что примирение методов машинного обучения уже показывает более лучшие результаты, чем аналогичный алгоритм без этих методов. Это логично, так как алгоритмы машинного обучения позволяет выбирать статистические модели, которые более эффективно кодируют элементы изображения. Попробуем улучшить результаты, варьируя параметры случайного леса.

Видно, что случайный лес из 100 решающих деревьев глубины 1 в среднем показывает себя лучше, чем аналогичная модель из 10 решающих деревьев глубины 4. Данная особенность часто проявляется в ансамблевых алгоритмах.

Рассмотрим качество восстановленных после компрессии изображений на примере изображения «Lena».

Зависимость между битовыми затратами и качеством восстановленного изображения видна невооруженным взглядом. Несмотря на то, что в обоих случаях изображение успешно восстанавливается, на рисунке … заметны…

* 1. **Заключение к главе**

Все предложенные в рамках работы методы сжатия изображений показывают

неплохие результаты. Контекстное адаптивное арифметическое кодирование позволяет добиться более высокой степени сжатия изображения по сравнению со стандартом JPEG. Использования методов машинного обучения позволило повысить эффективность контекстного адаптивного арифметического кодирования. Это объясняется тем, что статические связи между вейвлет-коэффициентами не линейны и аппроксимация их многочленом первой степени (как в базовой версии алгоритма) показывает себя хуже, чем методы машинного обучения, способные восстанавливать более сложные зависимости.

# **Заключение**

# **Список использованной литературы**

[1] Д.С.Ватолин «Алгоритмы cжатия изображений». (Методическое пособие).1999. 76 с.

[2] С.В. Умняшкин. «Основы теории цифровой обработки сигналов». М.: Техносфера, 2016. 528 с.

[3] Добеши И. «Десять лекций по вейвлетам». "РХД", 2001 г. 464 с.

[4] В. Н. Потапов, “Арифметическое кодирование сообщений с использованием случайных последовательностей”, ПДМ, 2008, № 2(2), 133 с.

[5] А. В. Григорьев «Компрессия изображений на основе пакетных вейвлет-преобразований». (Бакалаврская работа). 2016. 53 с.

[6] Научная библиотека [Электронный ресурс]. URL: <https://scask.ru/a_lect_cod.php?id=12>

[7] С. К. Абармов, Н. В. Бурцев, С. С. Кривенко, А. Н. Зеляченко, В. В. Лукин «Автоматическое сжатие в окрестности оптимальной рабочей точки изображений с шумом кодерами типа SPIHT и JPEG2000». Информационные технологии. 2016. 109с.

[8] Жерон Орельен «Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow» 2018, 688 с.

[9] В.В.Вьюгин «Математические основы машинного обучения и прогнозирования». 2013. 387с.

[10] C. П. Чистяков «Случайные леса: обзор». 2013. С. 117–136

[11] Груздев А.В. «Прогнозное моделирование в IBM SPSS Statistics, R и Python. Метод деревьев решений и случайный лес». 2017. C. 642 с.

[12] Р. Гонсалес, Р. Вудс. «Цифровая обработка изображений». 2012. 1104 с.

[13] Миронов А.М. «Машинное обучение учебное пособие». 2018. 84с.

[14] Семенюк В. В. «Обзор стандарта JPEG2000» 2002. 10с.

[15] OpenCV. <https://docs.opencv.org/4.x/index.html>

[16] PyWavelets. https://pywavelets.readthedocs.io/en/latest/