
Анализ цифровой системы передачи данных через зашумленный канал связи на основе метода сжатия Хаффмана и кодированием при помощи циклических кодов Боуза-Чоудхури-Хоквингема

В. А. Бренько

Тихоокеанский государственный университет, Хабаровск

Аннотация: В данной статье рассматривается эффективность цифровой системы передачи данных через зашумленный канал связи, использующей метод сжатия Хаффмана и циклическое кодирование БЧХ (Боуза-Чоудхури-Хоквингема). Сжатие Хаффмана уменьшает избыточность данных, что повышает эффективную скорость передачи, а коды БЧХ обеспечивают обнаружение и исправление ошибок, возникающих из-за шума в канале. Анализ, вероятно, включает оценку таких параметров, как степень сжатия, скорость передачи данных, вероятность ошибки после декодирования и вычислительная сложность алгоритмов. Результаты показывают, насколько эффективно данное сочетание методов повышает надежность передачи данных в условиях шума.

Ключевые слова: цифровая система передачи, циклическое кодирование, степень сжатия, декодирование, кодирование.

Введение

В современном мире цифровая передача данных является фундаментом информационных технологий, обеспечивая связь, доступ к информации и функционирование множества систем. От надежности и эффективности этих систем зависит огромное количество процессов в нашей жизни. Однако, несмотря на стремительное развитие технологий, физические каналы связи остаются подверженными воздействию различных помех и шумов, которые могут искажать передаваемые данные и приводить к ошибкам. Для успешной передачи данных через зашумленные каналы необходимо решать две ключевые задачи: максимально эффективно использовать пропускную способность канала и обеспечить целостность передаваемой информации. Первая задача решается с помощью методов сжатия данных, которые позволяют уменьшить объем передаваемой информации без существенной потери или с допустимой потерей качества. Вторая задача требует применения методов помехоустойчивого кодирования (кодирования с коррекцией ошибок), которые добавляют избыточность в передаваемые данные таким

образом, чтобы на приемной стороне можно было обнаружить и, при необходимости, исправить ошибки, возникшие в канале связи. Целью исследования является анализ эффективности совместного использования метода сжатия Хаффмана и циклического кодирования Боуза-Чоудхури-Хоквингема (далее БЧХ) для повышения надежности и скорости передачи данных через зашумленный канал связи. Это предполагает определение, насколько данная комбинация методов позволяет уменьшить вероятность ошибки на приеме при заданном уровне шума в канале, а также оценить влияние сжатия на общую пропускную способность системы.

Результаты и анализ исследования

Моделирование цифрового канала связи заключается в формировании битового массива ошибок с помощью заданного значения вероятности и наложении бинарного массива ошибок на бинарный массив. Структурная схема цифровой системы передачи показана на рис.1.

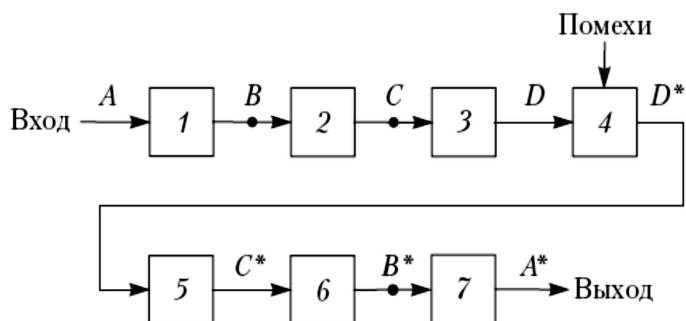


Рис. 1. – Структурная схема цифровой системы передачи

1 - кодер источника (блок эффективного кодирования); 2 - кодер канала (блок помехоустойчивого кодирования); 3 - модулятор; 4 - линия связи; 5 - демодулятор; 6 - декодер канала (блок помехоустойчивого декодирования); 7 - декодер источника (блок эффективного декодирования)

Для решения задачи был разработан программный модуль на языке C++ реализующий следующие блоки структурной модели цифровой системы передачи информации рис.1:

- а) кодер-декодер источника (1, 7);
- б) кодер-декодер канала (2, 6);
- в) канал связи (4).

В качестве источника информации используются цифровые данные статических изображений в градациях яркости. Для подготовки данных необходимо написать программу на языке C++ реализующую следующие функции:

1. Открытие и загрузка произвольного изображения Image из файла формат .bmp в цветовом изображении (далее RGB) формате (24 бит) с шириной W и высотой H, формирование трехбайтовых двумерных массивов это R, G и B. (Код скрыт), которой позволяет использовать графический интерфейс для удобного выбора изображения (файла в форматах .png или .jpg) с локального диска, начиная просмотр с диска C, и затем сохранить путь к выбранному изображению для дальнейшего использования в программе. Он также включает базовую обработку ошибок и отмены со стороны пользователя.

Задаем параметры стандартного окна просмотра файлов Windows (class OPARGB) который предоставляет удобный способ для:

- загрузки пиксельных данных изображения (в форматах RGB или ARGB) и их преобразования в удобное планарное представление (отдельные массивы для R, G, B). Это упрощает операции, которые манипулируют отдельными цветовыми каналами.

- обратной конвертации из планарного представления в стандартный интерливинг-массив RGB. Автоматического управления памятью для своих внутренних массивов через деструктор.

Этот тип структуры данных часто используется в задачах обработки изображений, где может быть необходимо работать с отдельными цветовыми каналами независимо друг от друга.

2. Формирование байтового массива яркости (далее Y).

3. Формирование битового массива яркости (далее Y_{bit}).

(Код (class YUV) скрыт) который предназначен для хранения изображения в цветовом пространстве (далее YUV) (или YCbCr, в

зависимости от точных коэффициентов) и выполнения преобразований между RGB и YUV.

Цветовое пространство YUV часто используется в видеосигналах и цифровых изображениях, так как оно отделяет информацию о яркости от информации о цвете. Это позволяет эффективно сжимать данные, поскольку человеческий глаз менее чувствителен к потерям в цветности, чем в яркости.

4. Проведем тестирование для нескольких изображений.

Это изображение успешно прошло через преобразование в YUV и обратно. Черно-белое изображение прошло через YUV без каких-либо изменений, всё прекрасно. Попробуем изображение большого разрешения. Изображения с большими разрешениями тоже прекрасно проходят через YUV.

5. Разработка кодера-декодера источника.

Программа на языке C++ реализует следующие функции:

- реализация алгоритма сжатия массива на основе дискретно косинусного преобразования (далее ДКП) и алгоритма Хаффмана. (Код скрыт) который реализует **стадию квантования коэффициентов яркости в алгоритме сжатия изображений**. Он берет преобразованные коэффициенты YТ, масштабирует их в зависимости от заданного качества Qua, делит на значения из матрицы квантования QY и округляет до целых чисел, сохраняя результат в YQ. Этот процесс уменьшает объем данных, но приводит к необратимой потере информации, что и является целью алгоритмов сжатием с потерями. После квантования исходные трансформированные данные YТ освобождаются.

Представленный код для ДКП массива изображения, ранее преобразованного в YUV и ДКП обратно в YUV [1,2].

Кодирование Хаффмана предназначено для выполнения сжатия данных по алгоритму Хаффмана. В данном контексте он применяется для сжатия квантованных ДКТ-коэффициентов, полученных из объекта класса ДКТ.

Класс Хаффмана предназначен для взятия дискретных коэффициентов изображения (после ДКТ и квантования) и их сжатия в бинарный поток, который может быть сохранен или передан. Он также упаковывает необходимые метаданные для последующей декомпрессии. Однако из-за упомянутых недостатков его реализация Хаффман-кодирования не является стандартной или оптимальной. Код выполняет кодирование массива при помощи метода Хаффмана [3].

Код для декодирования Хаффмана обратно в ДКТ предназначен для **декодирования** сжатых данных и восстановления объекта ДКТ, содержащего квантованные коэффициенты. Формирование битового сжатого массива Y_{bit} с размером $LenY$.

Оценка коэффициента сжатия рассчитывается по формуле $K_{сж} = 8 \cdot W \cdot \frac{N}{LenY}$. Декодирование массива Y_{bit} в байтовый массив Y . И вывод декодированного изображения на экран и визуальная оценка качества.

Проводим проверку с различными изображениями.

Изображение №1 – высокое разрешение сжимается при качестве 100%. Коэффициент сжатия получился равным 0,904. Тоже самое изображение, но качество уже 50%. Коэффициент сжатия теперь стал равен 0,568. Теперь качество изображения равно 10%. Коэффициент сжатия равен 0,309. Изображение визуально почти не потеряло своих качеств.

Изображение №2 – черно-белое изображение сжалось очень хорошо, при 100% качестве коэффициент сжатия стал равен 0,595. При 50% качестве коэффициент сжатия не стал сильно выше, всего 0,436. При 10% качестве черно-белое изображение малого разрешения визуально потеряло в качестве. Коэффициент сжатия стал целых 0.288.

Изображение №3 – ранее сжатое другой программой. При сжатии методом Хаффмана при качестве 100% наоборот увеличивает размер файла. Коэффициент сжатия равен 1,144. При качестве равном 50% коэффициент

сжатия равен 0,696. При 10% качестве, изображение визуально сильно потеряло свое качество. Коэффициент сжатия равен 0,332.

Разработка кодера-декодера канала

Для разработки выбраны помехоустойчивые коды БЧХ, порождающие полиномы которых приведены в таблице 1.

Таблица № 1

Помехоустойчивые коды БЧХ

№ п/п	Параметры БЧХ			Порождающий полином $g(x) - \text{mod } 8$	$d_{\min}$	Коррекция	
	n	m	k			r	S
1	31	16	15	107657	7	6	3

Рассчитываем избыточность: $r = n - k = 16$. Избыточность в процентах от общей длины кодового слова:

Избыточность: $\frac{r}{n} \cdot 100\% = \frac{16}{31} \cdot 100\% = 51,612\%$ Скорость кода отношением количества информационных символов к длине кодового слова:

Скорость: $\frac{k}{n} = \frac{15}{31} = 0,483$. В процентах: 48,387%.

И так, избыточность кода БЧХ (31,15) составляет 16 символов, что равняется 51,612% от длины кодового слова. Скорость кода равна 0,483, что в процентах равно 48,387%. На рис.2 показана схема функционального кодера.

Реализовать код программы на C++ реализующие операции кодирования и декодирования БЧХ бинарного массива Y_{bit} .

Код (class БЧХ), он предназначен для реализации **кодирования с исправлением ошибок** с использованием кодов БЧХ. Это позволяет добавлять избыточные биты к сообщению таким образом, чтобы при передаче по шумному каналу можно было обнаружить и исправить определенное количество ошибок. Это очень важно для надежной передачи данных, например, в памяти, на дисках или по беспроводным каналам [4].

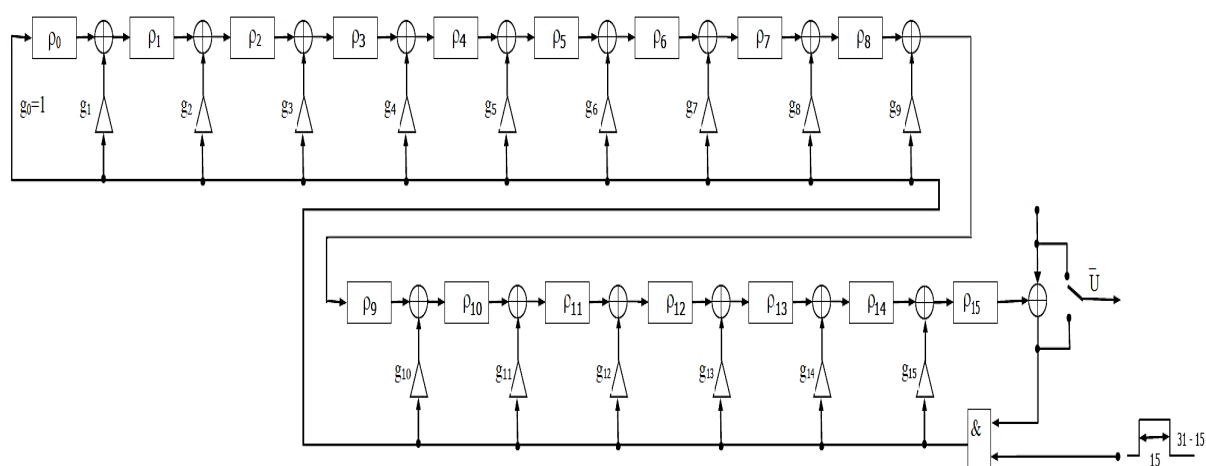


Рис. 2 – Схема функционального кодера на основании общей схемы

Код принимает на вход байтовый массив и ссылку на объект класса Хаффмана. И вызывает метод BCH-Builder с полученными аргументами. Функция build-BCH реализует процесс кодирования потока данных, где исходные данные разбиваются на m -битные блоки, каждый блок кодируется в n -битное кодовое слово BCH, а затем эти кодовые слова упаковываются бит за битом в выходной массив `encoded_data`. Это является ключевым шагом в использовании кодов с исправлением ошибок для повышения надежности данных. Эта функция предназначена для **кодирования потока данных** с использованием кода BCH, разбивая входные данные на блоки сообщений и кодируя каждый блок.

Считывается каждый бит по отдельности, и в зависимости от заданной длины составляет и кодирует сообщения. После чего, побитно записывает в массив.

Код описывает два метода: `TransformtoByteArray` (Преобразовать в массив байтов) и `TfTo-Хаффмана`, которые, являются частью класса BCH или другого класса, занимающегося декодированием и восстановлением данных после кодирования BCH [5]. Данный код осуществляет декодирование кода BCH с исправлением ошибок, записывая уже исправленное сообщение в новый массив.

А после, массив переводится в код Хаффмана, (функция decode) реализация декодера БЧХ-кода, но использует весьма нетрадиционный подход для обнаружения и исправления ошибок, основанный на циклическом сдвиге и прямом применении вычисленных промежуточных значений. Данный код получает на вход закодированное сообщение с ошибками декодирует сообщение, и возвращает его.

Разработка модели канала связи

Моделирование цифрового канала связи заключается в формировании битового массива ошибок с помощью заданного значения вероятности P и наложении бинарного массива ошибок на бинарный массив Y_{bit} . Значение вероятности P задается произвольно в интерфейсном окне разработанной системы передачи. (Код скрыт) с помощью которого вводим случайные побитовые ошибки в заданный массив данных. Основные особенности и недостатки:

- Назначение: Моделирование ошибок в бинарных данных, что является важным этапом при тестировании алгоритмов исправления ошибок (например, БЧХ-кодов).

- Инкапсуляция: Функция хорошо инкапсулирована и выполняет одну конкретную задачу.

- Гибкость: Позволяет задавать вероятность ошибки и исключать заголовок из процесса порчи данных.

- Критический баг в генераторе случайных чисел (далее ГСЧ): Самая большая проблема в коде – это многократный и неправильный вызов `std::srand`. `std::srand` должен быть вызван один раз в самом начале программы (`main()` или аналогично). Перезаполнение ГСЧ внутри цикла по битам делает генерируемые «случайные» числа крайне неслучайными (так как временная метка, используемая для заполнения, не меняется достаточно быстро), что приводит к некорректной симуляции ошибок.

- abs() избыточность: rand() возвращает неотрицательные числа, так что abs() не нужен.

Для корректной работы функции добавить шум (AddNoise) необходимо удалить все вызовы std::srand внутри циклов, оставив только один вызов в начале программы (например, в main() или при инициализации класса). Данный код создаёт массив шума, а после накладывает его на массив кодированных данных [6 - 8].

Проверка влияния шума

1. Не сжатые изображения до появления визуальных ошибок отображены в таблицах 2, 3, 4 и соответствующие им рисунки 3, 4, 5.

Таблица № 2

Изображение №1 высокого разрешение при качестве 100%

Шанс ошибки	0,0005	0,001	0,0015	0,002	0,0025	0,003
Ошибок не исправлено	2975	5660	8605	11743	15166	17347
Ошибок исправлено	221	384	568	760	970	1102

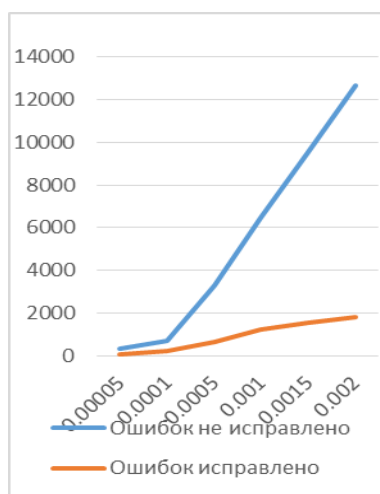


Рис.3

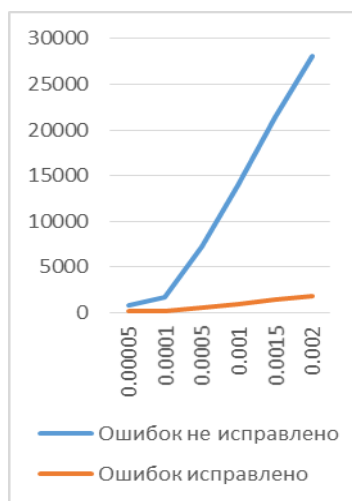


Рис.4

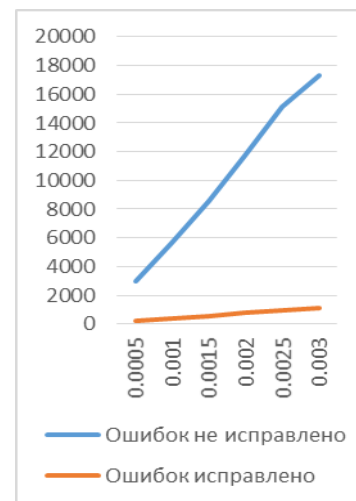


Рис.5

Таблица № 3

Изображение №2 черно-белое

Шанс ошибки	0,00005	0,0001	0,0005	0,001	0,0015	0,002
Ошибок не исправлено	796	1631	7187	13897	21396	28051
Ошибок исправлено	92	175	520	939	1405	1813

Таблица № 4

Изображение №3, ранее сжатое другой программой

Шанс ошибки	0,00005	0,0001	0,0005	0,001	0,0015	0,002
Ошибок не исправлено	338	713	3299	6445	9532	12664
Ошибок исправлено	79	225	654	1256	1564	1813

2. Сжатые изображения отражены в таблицах 5, 6, 7 и соответствующие им рисунки 6, 7, 8.

Таблица № 5

Изображение №1 высокого разрешение при качестве 100%

Шанс ошибки	0,00001	0,00002	0,00003	0,00004	0,00005
Ошибок не исправлено	193	200	251	320	377
Ошибок исправлено	42	48	64	75	101

Таблица № 6

Изображение №2 черно-белое

Шанс ошибки	0,00001	0,000025	0,00005	0,000075	0,0001
Ошибок не исправлено	224	255	535	775	1097
Ошибок исправлено	57	68	128	196	345

Таблица № 7

Изображение №3, ранее сжатое другой программой

Шанс ошибки	0,00001	0,00005	0,0001	0,00015	0,0002
Ошибок не исправлено	112	230	454	554	830
Ошибок исправлено	15	54	86	127	154

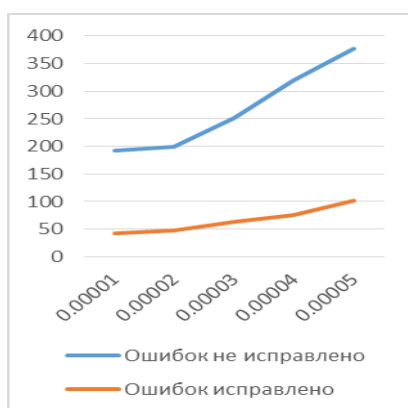


Рис.6



Рис.7



Рис.8

Заключение

Была разработана модель передачи информации на основе метода сжатия Хаффмана и кодированием при помощи циклических кодов БЧХ.

В ходе исследования было установлено, что сжатое изображение передается гораздо хуже, чем не сжатые. В результате сжатия изображения

шанс повреждения большего количества битов от даже малого количества ошибок очень высок. Одной из причин такого стал метод сжатия Хаффмана, который основан на частоте появления битов, а значит малейшее повреждение способно изменить порядок и повлиять на всё дерево.

Помехоустойчивое же кодирование очень хорошо исправляет некоторые недочеты метода сжатия Хаффмана, но не справляется при больших количествах ошибок.

Литература

1. Касами Т., Токура Н., Ивадари Е., Инагаки Я. Теория кодирования. / Пер. с японского. – М.: Мир, 1978. - 576 с.
2. Кларк Д.К. Кодирование с исправлением ошибок в системах цифровой связи /Пер. с англ. Гельфанда С. И.; Под ред. Цыбакова Б. С.. - М. : Радио и связь, 1987. - 391с. : ил.
3. Королёв А. И. Коды и устройства помехоустойчивого кодирования информации. - Минск, 2002. - 286 с.
4. Блейхут Р. Теория и практика кодов, контролирующих ошибки : Пер. с англ. – М. : Мир, 1986. 576 с.
5. Березюк Н. Т., Андрущенко А. Г., Мощицкий С. С. и др. Кодирование информации (двоичные коды). Харьков, издательское объединение «Вища школа», 1978. 252 с.
6. Циклические коды URL: scask.ru/a_book_p_net.php?id=129. Дата доступа 24.03.25
7. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки: Пер. с англ. - М.: Связь, 1979. - 744 с., ил.
8. Никитин Г. И., Поддубный С.С. Помехоустойчивые циклические коды: учеб.пособие; М-во общ. и проф. образования РФ. С.-петерб. ун-т аэрокосм. приборостроения. - Санкт-Петербург: 1998. - 71 с. : ил.
9. Теория прикладного кодирования: учебное пособие [доп. МО РБ] : в 2

т. Т. 2 / под ред. В. К. Конопелько. - Минск : БГУИР, 2004. - 398 с.

10. Костров Б.В. Основы цифровой передачи и кодирования информации. Рязань: РГРТУ, 2010. – 196 с.

References

1. Kasami T., Tokura N., Iwadari E., Inagaki Ya. Teoriya kodirovaniya [Coding Theory]. Perevod s yaponskogo. Moskva: Mir, 1978. 576 p.

2. Klark D. K. Kodirovaniye s ispravleniyem oshibok v sistemakh tsifrovoy svyazi [Error-correcting coding in digital communication systems]. Per. s angl. S. I. Gel'fanda; Pod red. B. S. Tsybakova. M.: Radio i svyaz', 1987. 391 p. : il.

3. Korolev A. I. Kody i ustroystva pomekhoustoychivogo kodirovaniya informatsii [Codes and devices for noise-resistant coding of information]. Minsk, 2002. 286 p.

4. Bleykhut R. Teoriya i praktika kodov, kontroliruyushchikh oshibki [Theory and Practice of Error-Controlling Codes]: Per. s angl. M.: Mir, 1986. 576 p.

5. Berezyuk N. T., Andrushchenko A. G., Moshchitskiy S. S. i dr. Kodirovaniye informatsii (dvoichnyye kody) [Information coding (binary codes)]. Khar'kov, izdatel'skoye ob'yedineniye «Vishcha shkola», 1978. 252 p.

6. Tsiklicheskiye kody [Cyclic codes]. URL: scask.ru/a_book_p_net.php?id=129. Data dostupa 24.03.25

7. Mak-Vil'yams F. Dzh., Sloen N. Dzh. A. Teoriya kodov, ispravlyayushchikh oshibki [Theory of error-correcting codes]: Per. s angl. M.: Svyaz', 1979. 744 p., il.

9. Teoriya prikladnogo kodirovaniya [Theory of applied coding]: uchebnoye posobiye [dop. MO RB]: v 2 t. Т. 2. Pod red. V. K. Konopel'ko. Minsk: BGUIR, 2004. 398 p.

10. Kostrov B.V. Osnovy tsifrovoy peredachi i kodirovaniya informatsii [Fundamentals of digital transmission and coding of information]. Ryazan: RGRTU, 2010. 196 p.

Дата поступления: 24.06.2025

Дата публикации: 25.08.2025