

МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

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

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №02

По дисциплине «Введение в цифровую культуру и программирование»

Эффективное кодирование

Вариант №1

Выполнил студент группы №М3109
Бабурин Тимур Сергеевич

Проверил
Хлопотов Максим Валерьевич

САНКТ-ПЕТЕРБУРГ

2019

ИСХОДНОЕ ИЗОБРАЖЕНИЕ И ФОТОГРАФИЯ В ЗАДАННОМ ФОРМАТЕ.



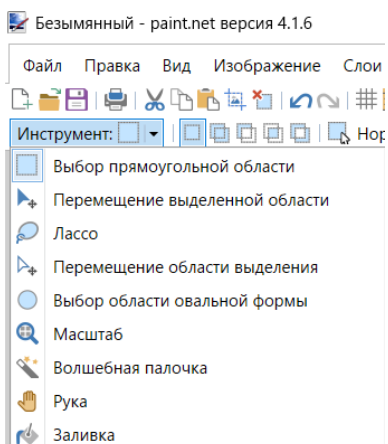
Исходное изображение.



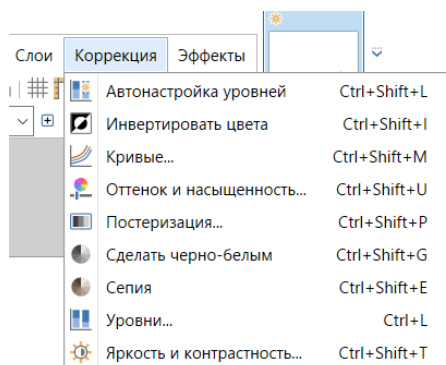
Фотография в заданном формате.

Изображение было обработано с помощью программы: "paint.net".

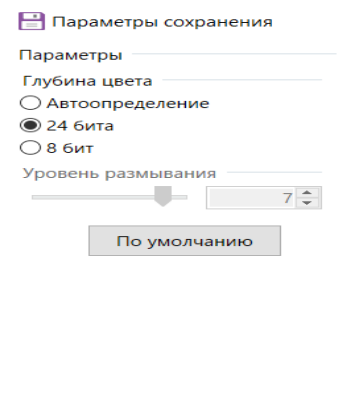
1. Обрезание происходило с помощью инструмента: "Выбор прямоугольной области".
2. Изменение цвета происходило с помощью функций: "Коррекция" и "Сделать чёрно-белым".
3. Глубина цвета была изменена при сохранении изображения (8 бит):



1.



2.



3.

ПОЛУЧЕННАЯ ЦИФРОВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ.

Программа.

```
#include <iostream>
#include <cmath>
#include "bitmap_image.hpp" //Библиотека растровых изображений C++
// для растрового изображения с 24-битным разрешением на пиксель.

using namespace std;

int main()
{
    bitmap_image image("image_ch_b.bmp"); //Объект, где хранится изображение.
    int picture[128][128]; //Массив для данных изображения.
    for (int i = 0; i < image.width(); i++)
    {
        for (int j = 0; j < image.height(); j++)
        {
            rgb_t colour; //Структура, где хранится значение характеристики RGB.
            image.get_pixel(i, j, colour); //Получаем данные.
            picture[i][j] = colour.red; //Значения RGB в нашем формате(ч/б) равны, поэтому записать
            //можем любое.
        }
    }

    cout << "Sequence:" << endl;

    int freq[261]; //Массив для частоты встречаемости, позже пригодится для вычисления энтропий.
    for (int i = 0; i < 128; i++)
    {
        picture[i][64] = round(double(picture[i][64]) / 20) * 20; //Квантование.
        freq[picture[i][64]]++;
        cout << picture[i][64] << ' '; //Выводим на консоль среднюю строку.
    }

    cout << endl << endl << "Count and frequency:" << endl;

    float entropy = 0;
    for (int i = 0; i < 261; i++)
    {
        if (freq[i] != 0)
        {
            cout << i << ' ' << freq[i] << endl; //Выводим на консоль частоту встречаемости.
            entropy -= (double(freq[i]) / 128) * log2(double(freq[i]) / 128); //Считаем энтропию
            //с помощью формулы.
        }
    }

    cout << endl;

    cout << "Entropy = " << entropy; //Выводим на консоль энтропию.
    return 0;
}
```

Последовательность.

180 200 200 200 220 220 200 200 180 180 180 200 220 220 220 220 220 200 180 200 240
240 260 240 240 240 240 240 220 220 220 220 220 200 180 160 160 140 140 160 160 140 140 120
80 60 80 100 100 80 60 80 60 40 60 60 60 60 40 40 40 60 60 60 40 40 40 80 100 100 100 100
100 120 100 100 100 120 120 120 120 140 140 140 140 120 120 140 140 120 100 100 80 80 80 80
60 60 100 120 120 140 140 160 180 200 200 200 200 180 180 180 200 200 200 220 220 240 240
260 260 260 260 260 240 240 240

**ВЕРОЯТНОСТЬ ПОЯВЛЕНИЯ.КОЛИЧЕСТВО СИМВОЛОВ
АЛФАВИТА.ЭНТРОПИЯ.РАСЧЁТНАЯ ДЛИНА ДВОИЧНОГО КОДА.**

<i>Символ</i>	<i>Вероятность</i>
160	$\frac{5}{128} = 0.039$
260	$\frac{6}{128} = 0.046$
40	$\frac{8}{128} = 0.062$
80	$\frac{9}{128} = 0.070$
180	$\frac{10}{128} = 0.078$
120	$\frac{11}{128} = 0.085$
60	$\frac{12}{128} = 0.093$
140	$\frac{12}{128} = 0.093$
240	$\frac{12}{128} = 0.093$
100	$\frac{13}{128} = 0.101$
220	$\frac{14}{128} = 0.109$
200	$\frac{16}{128} = 0.125$

Количество символов: 12.

Энтропия по формуле: 3.5204.

Расчётная длина двоичного кода: 4.

<i>Символ</i>	160	260	40	80	180	120	60	140	240	100	220	200
<i>Код</i>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011

Эффективный код.

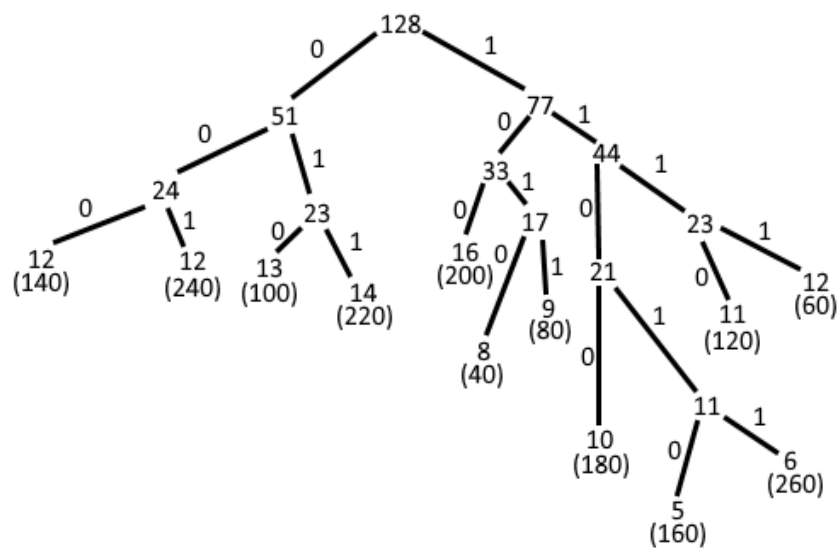
КОДЫ ШЕННОНА-ФАНО.

Символ	Вероятность	1	2	3	4
200	$\frac{16}{128} = 0.125$	0	0	0	
220	$\frac{14}{128} = 0.109$			1	
100	$\frac{13}{128} = 0.101$		1	0	
140	$\frac{12}{128} = 0.093$			1	0
240	$\frac{12}{128} = 0.093$				1
60	$\frac{12}{128} = 0.093$	1	0	0	
120	$\frac{11}{128} = 0.085$			1	0
180	$\frac{10}{128} = 0.078$				1
80	$\frac{9}{128} = 0.070$		1	0	0
40	$\frac{8}{128} = 0.062$				1
260	$\frac{6}{128} = 0.046$			1	0
160	$\frac{5}{128} = 0.039$				1

Символ	200	220	100	140	240	60	120	180	80	40	260	160
Код	000	001	010	0110	0111	100	1010	1011	1100	1101	1110	1111

Эффективный код.

КОДОВОЕ ДЕРЕВО ХАФФМАНА.



Эффективный код.

КОДЫ ХАФФМАНА ДЛЯ СИМВОЛОВ.

<i>Символ</i>	<i>Код</i>
200	100
220	011
100	010
60	1111
240	001
140	000
120	1110
180	1100
80	1011
40	1010
260	11011
160	11010

**ЗАКОДИРОВАННОЕ РАВНОМЕРНЫМ КОДОМ СООБЩЕНИЕ, ДЛИНА
КОВОГО СЛОВА, КОЛИЧЕСТВО ПЕРЕДАННОЙ ИНФОРМАЦИИ ПРИ
ПЕРЕДАЧЕ СООБЩЕНИЯ.**

Последовательность, которая была закодирована:

180 200 200 200 220 220 200 200 180 180 180 200 220 220 220 220 220 200 180 200 240
240 260 240 240 240 240 220 220 220 220 220 200 180 160 160 140 140 160 160 140 140 120
80 60 80 100 100 80 60 80 60 40 60 60 60 60 40 40 40 60 60 60 40 40 40 80 100 100 100 100
100 120 100 100 100 120 120 120 120 140 140 140 140 120 120 140 140 120 100 100 80 80 80 80
60 60 100 120 120 140 140 160 180 200 200 200 200 180 180 180 200 200 200 220 220 240 240
260 260 260 260 260 240 240 240

Закодированное сообщение:

010010111011101110101010101110110100010001001011101010101010101010101010
110100101110001000000110001000100010001000101010101010101010101010110100000000
0001110111000000000111011101010011011000111001100100110110001101100010011001
1001100110001000100010011001100110001000100010001000111001100110011001100101
0110011001100101010101010101010111011101110111010101010111011101011001100100
1100110011001101100110100101010101011101110000010010111011101110110100010001
00101110111011101010101000100000010001000100010001100010001000

Длина кодового слова: 4.

Количество информации при передаче сообщений: 512 бит.

**ЗАКОДИРОВАННАЯ ПОЛУЧЕННЫМ КОДОМ ШЕННОНА–ФАНО
ПОСЛЕДОВАТЕЛЬНОСТЬ, ДЛИНА КОДОВОГО СЛОВА, КОЛИЧЕСТВО
ПЕРЕДАННОЙ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ СООБЩЕНИЯ.**

Последовательность, которая была закодирована:

180 200 200 200 220 220 200 200 180 180 180 200 220 220 220 220 220 200 180 200 240
240 260 240 240 240 240 240 220 220 220 220 220 200 180 160 160 140 140 160 160 140 140 120
80 60 80 100 100 80 60 80 60 40 60 60 60 60 40 40 40 60 60 60 40 40 40 80 100 100 100 100
100 120 100 100 100 120 120 120 120 140 140 140 140 120 120 140 140 120 100 100 80 80 80 80
60 60 100 120 120 140 140 160 180 200 200 200 200 180 180 180 200 200 200 220 220 240 240
260 260 260 260 260 240 240 240

Закодированное сообщение:

1011000000000001001000000101110111011000001001001001001000101100001110
111111001110111011101110111001001001001001000101111111110110011011111111011
0011010101100100110001001011001001100100110110010010010011011101110110010010
0110111011101110111000100100100100101010010010010101010101010101001100110011
0011010101010011001101010010010110011001100110010010001010101010011001101111
10110000000000001011101110110000000000100101110111110111011101110111001110
1110111

Количество информации при передаче сообщений: 457 бит.

**ЗАКОДИРОВАННАЯ ПОЛУЧЕННЫМ КОДОМ ХАФФМАНА
ПОСЛЕДОВАТЕЛЬНОСТЬ, ДЛИНА КОДОВОГО СЛОВА, КОЛИЧЕСТВО
ПЕРЕДАННОЙ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ СООБЩЕНИЯ.**

Последовательность, которая была закодирована:

180 200 200 200 220 220 200 200 180 180 180 200 220 220 220 220 220 200 180 200 240
240 260 240 240 240 240 240 220 220 220 220 220 200 180 160 160 140 140 160 160 140 140 120
80 60 80 100 100 80 60 80 60 40 60 60 60 60 40 40 40 60 60 60 40 40 40 80 100 100 100 100
100 120 100 100 100 120 120 120 120 140 140 140 140 120 120 140 140 120 100 100 80 80 80 80
60 60 100 120 120 140 140 160 180 200 200 200 200 180 180 180 200 200 200 220 220 240 240
260 260 260 260 260 240 240 240

Закодированное сообщение:

1100100100100011011100100110011001100100011011011011011100110010000100
1110110010010010010010110110110110111001100110101101000000011010110100000001
110101111111011010010101111111011111110101111111111111111111010101010101111111
1111110101010101010101011010010010010010111001001001011101110111011100000000
0000011101110000000111001001010111011101110111111111101011101110000000110101
10010010010010011001100110010010010001101100100111011110111101111011110111011001
001001

Количество информации при передаче сообщений: 456 бит.

РАСЧЕТЫ ПО П. 8 – 10 ЗАДАНИЯ НА ЛАБОРАТОРНУЮ РАБОТУ.

$$I_{\text{ср}} = \sum_{i=1}^m f_i k_i \quad Q = 1 - \frac{\text{entropy}}{I_{\text{ср}}} \quad R = \frac{l_{\text{исходное}}}{l_{\text{закодированного}}}$$

Средняя длина кодовой комбинаций:

Шеннон-Фано:

$$I_{\text{ср}} = 3 \times \frac{16}{128} + 3 \times \frac{14}{128} + 3 \times \frac{13}{128} + 3 \times \frac{12}{128} + 4 \times \frac{12}{128} + 4 \times \frac{12}{128} + 4 \times \frac{11}{128} + 4 \times \frac{10}{128} + 4 \times \frac{9}{128} + 4 \times \frac{8}{128} + 4 \times \frac{6}{128} + 4 \times \frac{5}{128} = 3.570$$

Хаффман:

$$I_{\text{ср}} = 3 \times \frac{12}{128} + 3 \times \frac{12}{128} + 4 \times \frac{12}{128} + 5 \times \frac{5}{128} + 5 \times \frac{6}{128} + 4 \times \frac{8}{128} + 4 \times \frac{9}{128} + 4 \times \frac{10}{128} + 4 \times \frac{11}{128} + 3 \times \frac{13}{128} + 3 \times \frac{14}{128} + 3 \times \frac{16}{128} = 3.5625$$

Степень сжатия сообщений:

Шеннон-Фано:

$$R = \frac{512}{457} = 1.12035$$

Хаффман:

$$R = \frac{512}{456} = 1.12280701$$

Избыточность для сформированных кодов:

Шеннон-Фано:

$$Q = 1 - \frac{3.5204}{3.570} = 0.0138$$

Хаффман:

$$Q = 1 - \frac{3.5204}{3.5625} = 0.0118$$

ВЫВОД.

Методика Шеннона-Фано не всегда приводит к однозначному построению кода. Так как при разбиении на подгруппы можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппу. В результате среднее число символов на букву окажется другим. При этом метод Хаффмана гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву. Метод Хаффмана производит такое сжатие, что сжимает данные до их энтропий, если вероятности символов точно равны отрицательным степеням двойки. По моему мнению метод Хаффмана всегда будет лучше результатов по методу Шеннона-Фано.