

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа № 6
«Коды Хаффмана»
Вариант 2

Проверил:
Рыбенков Е.В.

Выполнил:
ст. гр. 850701
Филипцов Д. А.

Минск 2021

1 Цель работы

Использование кодов Хаффмана для кодирования с минимальной избыточностью.

2 Задание

Выполнить кодирование изображения с помощью алгоритма Хаффмана. Выполнить декодирование изображения. Написать собственную реализацию функций `huffmandict`, `huffmanenco`, `huffmandeco`.

3 Ход работы

MATLAB-код:

```
J = rgb2gray(im2double(imread('kodim02.png')));
I = J(1:256, 1:256);
I = round(I.*255);
histogram = zeros(1, 256);
for i = 1:size(I, 1)
    for j = 1:size(I, 2)
        histogram(I(i, j)+1) = histogram(I(i, j)+1) + 1;
    end
end
histogram = histogram./(size(I, 1)*size(I, 2));
H = 0;
for i = 1:length(histogram)
    if histogram(i) ~= 0
        H = H - histogram(i)*log2(histogram(i));
    end
end
symbols = unique(I)';
prob = zeros(1, length(symbols));
for i = 1:length(symbols)
    prob(i) = histogram(symbols(i)+1);
end
sig = zeros(1, size(I, 1)*size(I, 2));
for i = 1:size(I, 1)
    for j = 1:size(I, 2)
        sig((i-1)*size(I, 1) + j) = I(i, j);
    end
end
dict = my_huffmandict(symbols, prob);
code = my_huffmanenco(sig, dict);
sig = my_huffmandeco(code, dict);
```

```

avglen = 0;
for i = 1:length(dict)
    avglen = avglen + prob(i) * length(dict{i,2});
end
bits_num = length(code);
bits = ceil(log2(length(dict)));
bits_num = bits_num + bits*length(dict);
for i = 1:length(dict)
    bits_num = bits_num + length(dict{i,2});
end
C = size(I, 1)*size(I, 2)*8/bits_num*100;
subplot(1, 2, 1); imshow(I, [0 (256 - 1)]);
subplot(1, 2, 2); plot(histogram, 'o');

function dict = my_huffmandict(symbols, p)
n = size (p, 2);
HT = zeros (2 * n-1,4);
for i=1:n
    HT(i,1)=p(i);
end
HT0=HT;

for i=1:n-1
    a=HT0(:,1);
    [b,1]=sort(a,'descend');
        s = b (n-i + 1) + b (n-i);
        HT0 (n + i, 1) = s;
        HT0 (1 (n-i + 1), 1) = 0;
        HT0 (1 (n-i), 1) = 0;

        HT0 (1 (n-i + 1), 2) = n + i;
        HT0 (1 (n-i), 2) = n + i;

        HT0 (n + i, 3) = 1 (n-i + 1);
        HT0 (n + i, 4) = 1 (n-i);

    HT(n+i,1)=s;
    HT(1(n-i+1),2)=n+i;
    HT(1(n-i),2)=n+i;
    HT(n+i,3)=1(n-i+1);
    HT(n+i,4)=1(n-i);
end

a={;};
for i=1:n
    a{1,i}=' ';
end

```

```

for i=1:n
    f=i;
    while (HT (f, 2) ~= 0)
        q = HT (f, 2);

        if HT(q,3)==f
            a {i} = strcat ('0', a {i});
        else
            a {i} = strcat ('1', a {i});
        end
        f = q;
    end
end
dict = cell(n, 2);
for i = 1:n
    dict(i, 1) = {symbols(i)};
    dict(i, 2) = a(i);
end
end

function code = my_huffmanenco(sig, dict)
code = '';
for i = 1:length(sig)
    str = '';
    for j = 1:length(dict)
        if sig(i) == dict{j, 1}
            str = dict{j, 2};
        end
    end
    code = strcat(code, str);
end
end

function sig = my_huffmandeco(code, dict)
sig = zeros(1);
str = '';
cnt = 1;
for i = 1:length(code)
    str = strcat(str, code(i));
    for j = 1:length(dict)
        if length(str) == length(dict{j, 2})
            if str == dict{j, 2}
                sig(cnt) = dict{j, 1};
                cnt = cnt + 1;
                str = '';
            break
        end
    end
end

```

```
end
end
end
end
end
```

Выполнение:

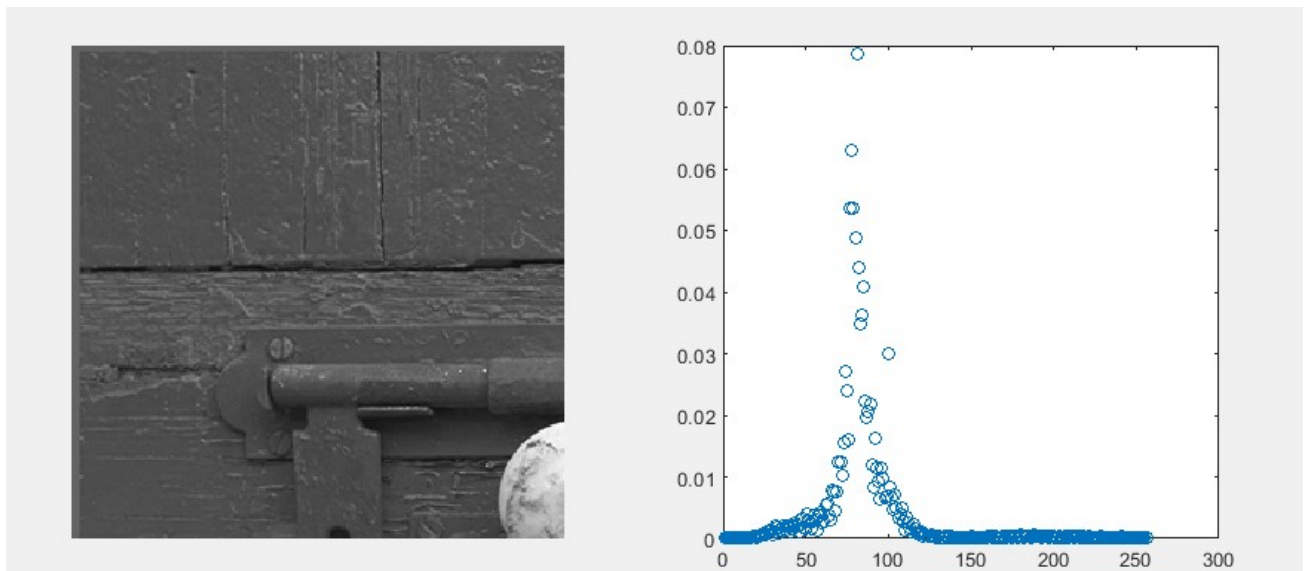


Рисунок 3.1 – Исходное изображение и его гистограмма

4 Вывод

В ходе лабораторной работы мы научились использовать коды Хаффмана для кодирования с минимальной избыточностью.

Энтропия источника – 6,9989.

Средняя длина кода – 7,0261.

Коэффициент сжатия – 112,9%.

Значения равны при реализации встроенными функциями и написанными самостоятельно.