

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

Веселов А.И.

инициалы, фамилия

Отчет по лабораторной работе №1

Методы блоковой обработки при сжатии с потерями на
примере стандарта JPEG
по дисциплине: Мультимедиа технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. _____ 5322

подпись, дата

Пятаков В.С.

инициалы, фамилия

Санкт-Петербург 2016

Цель работы:

Изучение алгоритмов, используемых в базовом режиме стандарта JPEG, анализ статистических свойств, используемых при сжатии коэффициентов дискретного косинусного преобразования, а также получение практических навыков разработки методов блоковой обработки при сжатии изображений с потерями.

Описание алгоритма кодека JPEG:

JPEG (JointPhotographicExpertsGroup) — один из популярных графических форматов, применяемый для хранения фотоизображений и подобных им изображений. Алгоритм JPEG позволяет сжимать изображение как с потерями, так и без потерь (режим сжатия lossless JPEG). Поддерживаются изображения с линейным размером не более 65535×65535 пикселей.

При сжатии изображение преобразуется из цветового пространства RGB в YCbCr. После преобразования RGB→YCbCr для каналов изображения Cb и Cr, отвечающих за цвет, может выполняться «прореживание». Далее яркостный компонент Y и отвечающие за цвет компоненты Cb и Cr разбиваются на блоки 8×8 пикселей. Каждый такой блок подвергается дискретному косинусному преобразованию (ДКП). Полученные коэффициенты ДКП квантуются (для Y, Cb и Cr в общем случае используются разные матрицы квантования) и пакуются с использованием кодирования серий и кодов Хаффмана. Стандарт JPEG допускает также использование значительно более эффективного арифметического кодирования. 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.

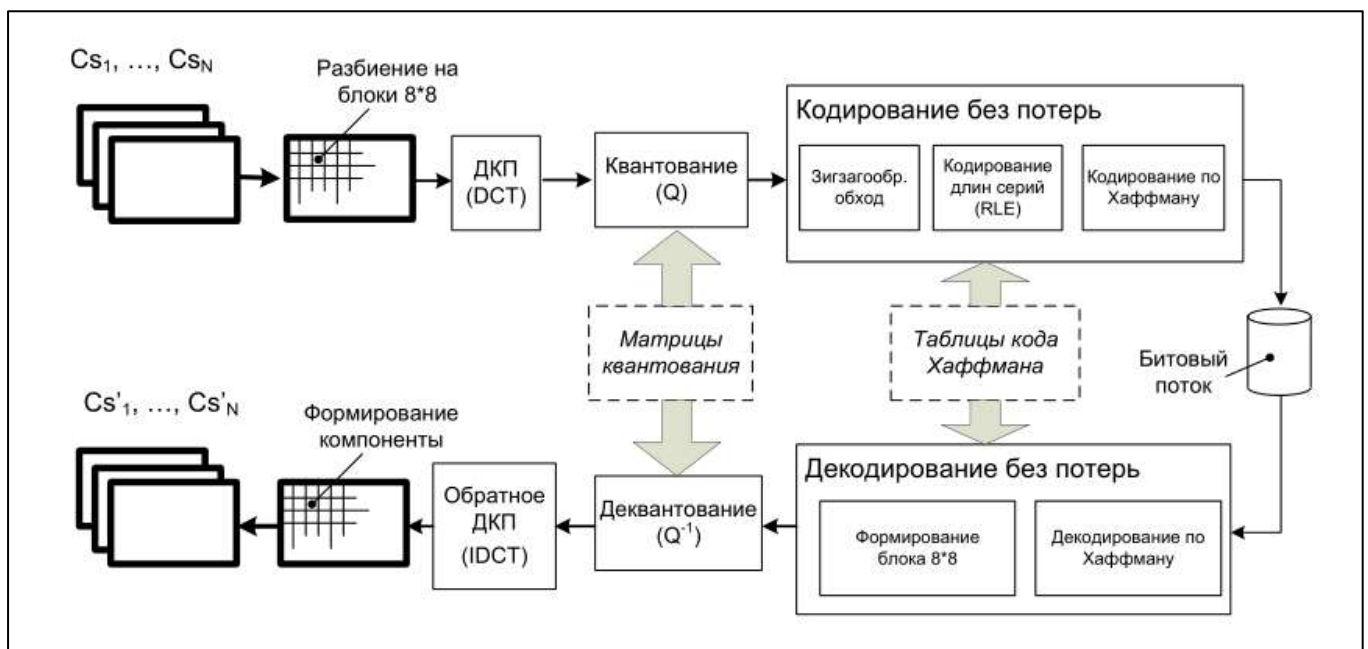


Рис.1.: Схема сжатия в базовом режиме стандарта JPEG

Результаты проделанной работы:

1) Дискретное косинусное преобразование.

Исходные изображения:



Рис.2.: Исходные изображения.

В первом пункте заданий необходимо сделать прямое и обратное Дискретно Косинусное преобразование представленных изображений, результаты каждого преобразования являются целочисленными, манипуляции производятся в YCbCr пространстве.

По полученным значениям после обратного ДКП необходимо сравнить их с исходными значениями каждой компоненты.

В первом пункте необходимо выполнить преобразования над яркостной компонентой изображения lena.bmp и посчитать PSNR его с исходным.

После обратного ДКП PSNR Y компоненты равен = 58.9 dB.

2) Квантования спектральных коэффициентов.

Для изображения *lena.bmp* построить графики зависимости PSNR получившегося изображения после деквантования и исходного от параметра матрицы квантования *R*.

R	1	10	20	40	60	80	100	200
Y	41,1	34,5	32,7	30,8	29,7	28,9	28,3	26,7
CR	43,4	39,6	38,8	38	37,6	37,1	36,8	36,3
CB	43,8	39,6	38,8	37,9	37,4	37,1	36,9	36,4

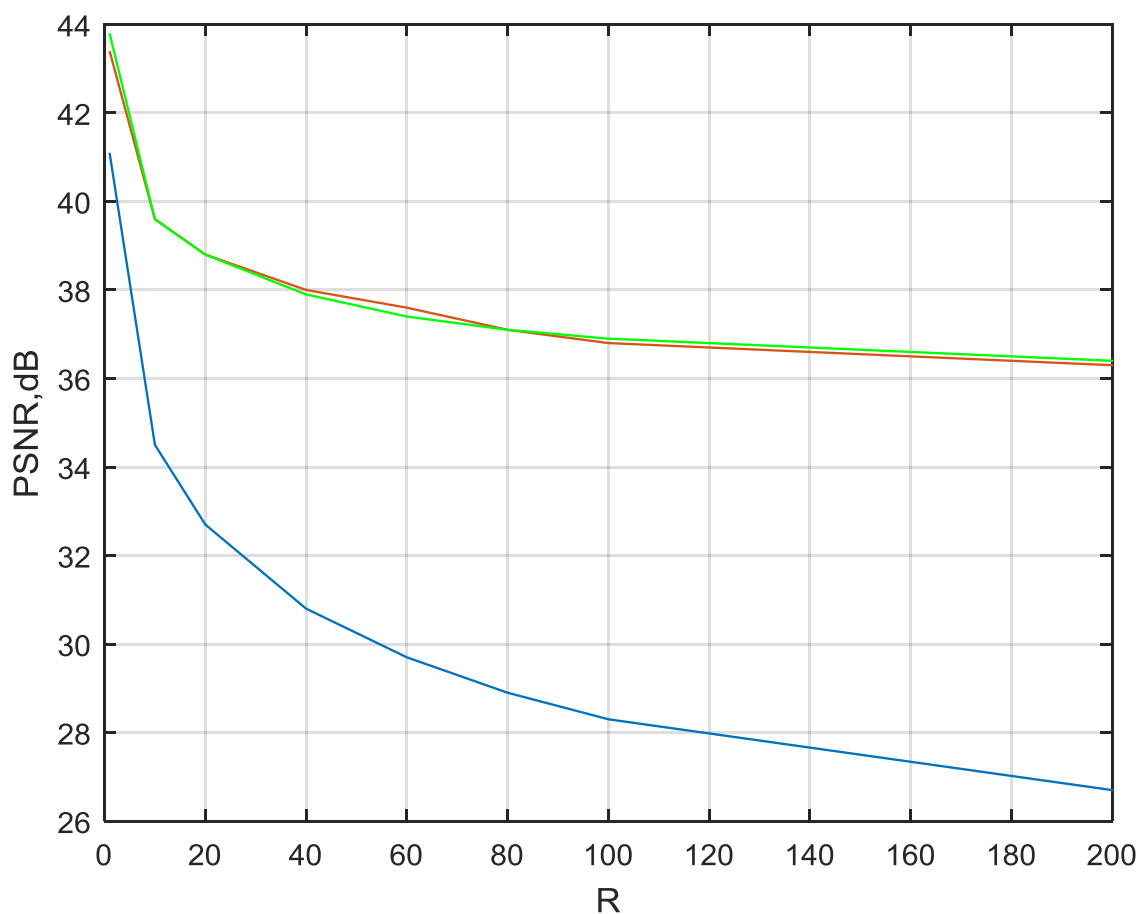


Рис..2.: Зависимость PSNR от *R* –целочисленного параметра, управляющего качеством обработки, для изображения “*lena.bmp*”

3)Сжатие без потерь.

Оценка эффективности использования разностного кодирования для коэффициентов постоянного тока.

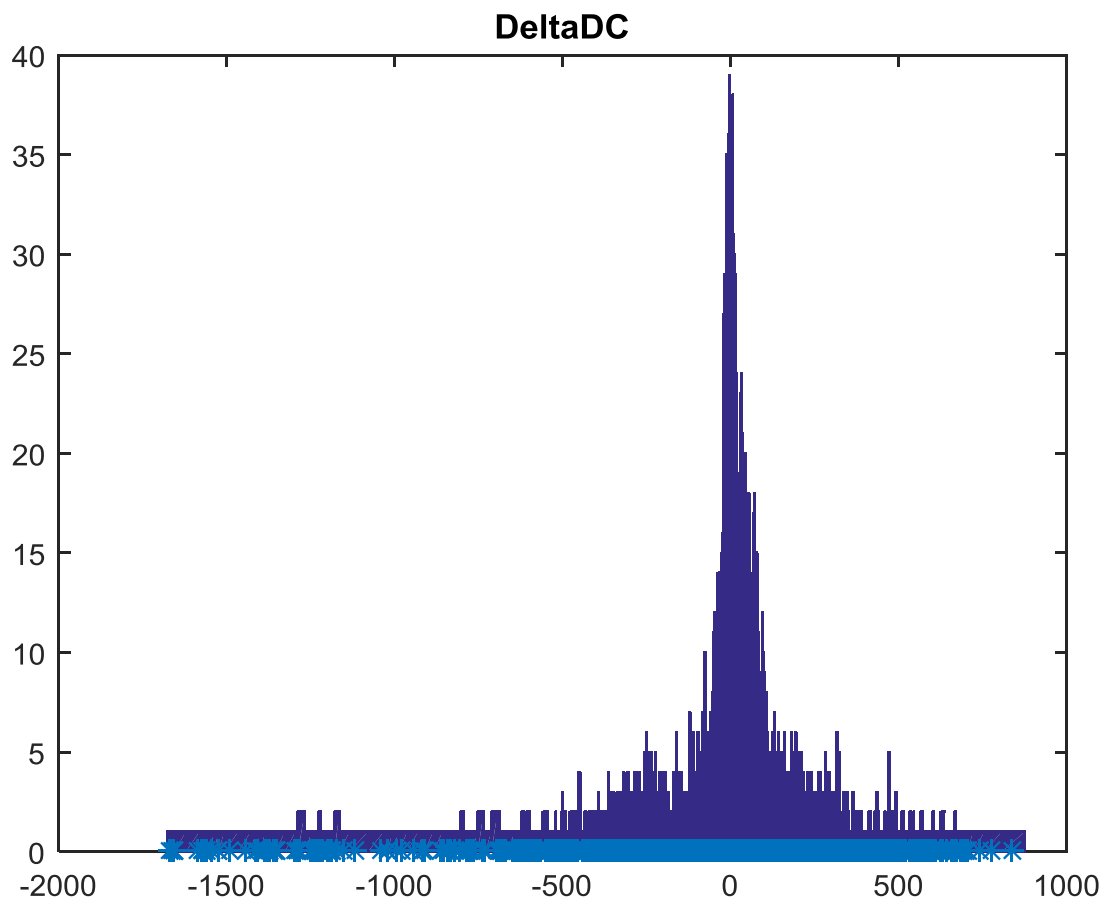


Рис.3.3.: Гистограмма частот $f(\Delta_{DC})$ для блоков 8*8 яркостной составляющей изображения "lena.bmp"

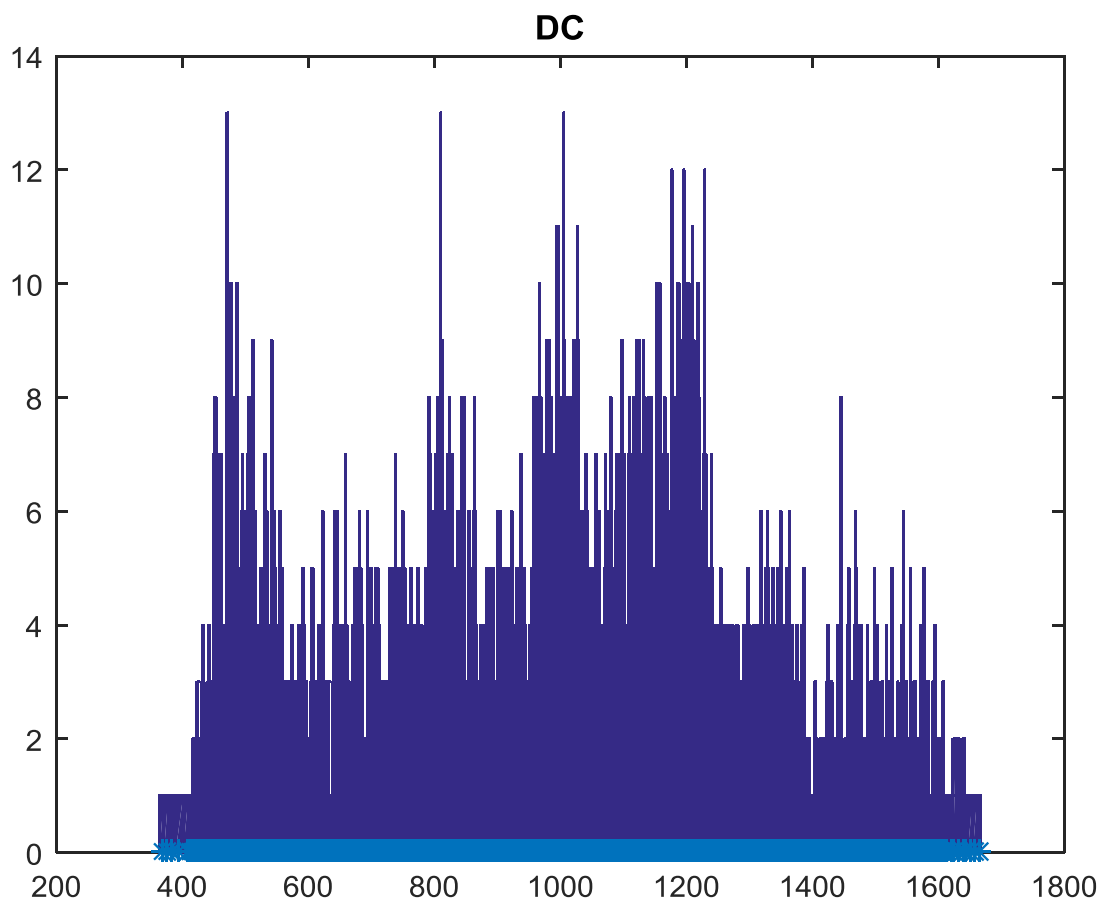


Рис.3.4.: Гистограмма частот $f(DC^q)$ для блоков 8×8 яркостной составляющей изображения "lena.bmp"

Оценка энтропии для множества $\{DC^q\}=8.9$.

Оценка энтропии для множества $\{\Delta_{DC}\}=8.8$.

Соотношение размеров в сжатом битовом потоке при $R=1$ для следующих данных:

- $BC(\Delta_{DC}) = 3.6\%$
- $Magnitude(\Delta_{DC}) = 6.2\%$
- $(Run, BC(Level)) = 57.5\%$
- $Magnitude(Level) = 32.6\%$

Соотношение размеров в сжатом битовом потоке при $R=5$ для следующих данных:

- $BC(\Delta_{DC}) = 15,2398 \%$
- $Magnitude(\Delta_{DC}) = 10,1254 \%$
- $(Run, BC(Level)) = 60,8231 \%$
- $Magnitude(Level) = 13,8117 \%$

4) Индивидуальное задание.

Оценить число бит, необходимых для представления спектральных коэффициентов на выходе ДКП, которое позволило бы организовать преобразование без потерь данных.

Преобразование без потерь означает, что полученные результаты должны полностью совпадать с исходными данными, следовательно, среднеквадратичная ошибка для каждого пикселя равна 0, а $PSNR = \infty$. Такие результаты получаются, если для хранения спектральных коэффициентов выделить 32 бита.

$255 = 2^8 \Rightarrow 8.0$ бит входит в преобразование
 $0,4090 = 2^{12} \Rightarrow 0.12$ бит на выходе при матричном умножении первом, получим
8.24 при втором умножении на транспонированную матрицу \Rightarrow
 $8+24=32$ для хранения спектральных коэффициентов.

Пример работы программы:



Выводы:

- 1) При использовании прямого и обратного дискретно косинусного преобразования на Y компоненте исходного изображения lena.bmp было получено значение PSNR равным 58.9 dB
- 2) Во втором пункте где мы занимались квантованием спектральных коэффициентов матрица квантования зависела от целочисленного параметра R. В ходе моделирования получилась зависимость такая, что при увеличении этого параметра, значения PSNR уменьшались.

Так же по полученным графикам можно сделать выводы, что квантование для Y компоненты выражен более резкий спад значений PSNR, чем при таких же значений R у компонент Cb и Cr, их значения практически равны.

3) Из построенных гистограм частот $f(DC^q)$ и $f(\Delta_{DC})$ можно сделать выводы об их распределении. Для яркостной компоненты изображения lena.bmp $f(DC^q)$ имеет вид равномерного распределения, а $f(\Delta_{DC})$ ближе к гаусовскому колоколу.

4) В последнем пункте оценивая соотношения размеров, пришли к выводу, что большую часть занимают данные $(Run, BC(Level))$.

Листинг программы:

```
clear;
L_rgb=imread('C:\Users\Виктор\Desktop\курсыои\lena.bmp');
[H W s]=size(L_rgb);
L_ycbcr=rgb2ycbcr(L_rgb);%перевод в YCBСR пространство
N=8;

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% прямое ДКП
DCT=zeros(H,W);
Y_image=zeros(H,W);
T=zeros(8,8);
for i=1:1:8
    for j=1:1:8
        if i==1
            T(i,j)=sqrt(1/N);
        else
            T(i,j)=sqrt(2/N)*cos(((2*(j-1)+1)*pi)/(2*N))*(i-1));
            T(i,j)=T(i,j);
        end
    end
end

X=zeros(8,8);
Y=zeros(8,8);
YY=zeros(8,8);
for u=1:8:H

    for l=1:8:W

        for i=1:1:8

            for j=1:1:8

                p=L_ycbcr(i+u-1,j+l-1,1);
                X(i,j)=p;
                Y_image(i+u-1,j+l-1)=p;

            end

        end

        Y=(T*X)*T';

        for h=1:1:8
            for v=1:1:8
                DCT(h+u-1,v+l-1)=round(Y(h,v));
            end
        end

    end

end
```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%обратное дкп
DCT_back=zeros(H,W);
Y1=zeros(8,8);
X1=zeros(8,8);

for u=1:8:H

    for l=1:8:W

        for i=1:1:8

            for j=1:1:8

                p=DCT(i+u-1,j+l-1,1);
                Y1(i,j)=p;

            end

        end
        X1=(T'*Y1)*T;

        for h=1:1:8
            for v=1:1:8
                DCT_back(h+u-1,v+l-1)=round(X1(h,v));
            end
        end

    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% psnr
PSNR=psnr_mse_maxerr(Y_image(:, :, 1), DCT_back(:, :, 1));

% A=double(Y_image);
% B=double(Y_image);
% d=sum((A(:)-B(:)).^2)/numel(size(A));
% psnr=10*log10(H*W/d);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% пункт 2
% КВАНТОВАНИЕ
R=1;
q=zeros(8,8);
for i=1:1:8
    for j=1:1:8

        q(i,j)=1+((i-1)+(j-1))*R;

    end
end

```

```

Y_dct=zeros(8,8);
Y_q=zeros(8,8);
Qantov=zeros(H,W);
for u=1:8:H

    for l=1:8:W

        for i=1:1:8

            for j=1:1:8

                p=DCT(i+u-1,j+l-1,1);
                Y_dct(i,j)=p;

            end

        end

        for h=1:1:8
            for v=1:1:8
                Y_q(h,v)=round(Y_dct(h,v)/q(h,v));
                Qantov(h+u-1,v+l-1)=Y_q(h,v);
            end
        end

    end
end
% % % % % % деквантование
D_Qantov=zeros(H,W);
Y_qant=zeros(8,8);
Y_dq=zeros(8,8);
for u=1:8:H

    for l=1:8:W

        for i=1:1:8

            for j=1:1:8

                p=Qantov(i+u-1,j+l-1,1);
                Y_qant(i,j)=p;

            end

        end

        for h=1:1:8

```

```

        for v=1:1:8
            Y_dq(h,v)=Y_qant(h,v)*q(h,v);
            D_Qantov(h+u-1,v+l-1)=Y_dq(h,v);
        end
    end

end

DCT_back_qantov=zeros(H,W);
Y2=zeros(8,8);
X2=zeros(8,8);

for u=1:8:H

    for l=1:8:W

        for i=1:1:8

            for j=1:1:8

                p=D_Qantov(i+u-1,j+l-1,1);
                Y2(i,j)=p;

            end

        end

        X2=(T'*Y2)*T;
        for h=1:1:8
            for v=1:1:8
                DCT_back_qantov(h+u-1,v+l-1)=round(X2(h,v));
            end
        end

    end

end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% psnr
PSNR2=psnr_mse_maxerr(Y_image(:,:,1),DCT_back_qantov(:,:,1));
figure(1);

R1=[1 10 20 40 60 80 100 200];
P1=[41.1 34.5 32.7 30.8 29.7 28.9 28.3 26.7];
P2=[43.4 39.6 38.8 38 37.6 37.1 36.8 36.3];
P3=[43.8 39.6 38.8 37.9 37.4 37.1 36.9 36.4];
plot(R1,P1,R1,P2,R1,P3,'g');
grid on;
xlabel('R');
ylabel('PSNR,dB');
% A=double(Y_image);
% B=double(DCT_back_qantov);

```

```

% d=sum((A(:)-B(:)).^2)/numel(size(A));
% psnr_2=10*log10(H*W/d);

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % 3 пункт
M=W/8*H/8;%размерность массива
DC_sred=(8*8*255*(1/8));
delta_DC=zeros(1,W/8*H/8);
n=1;
DC=zeros(1,M);
for i=1:8:H
    for j=1:8:W
        z=Qantov(i,j);
        if j==1
            d=z-DC_sred;
        else
            d=z-zpred;
        end
        zpred=z;
        delta_DC(1,n)=d;
        DC(1,n)=z;
        n=n+1;
    end
end

BC=zeros(1,M);
for i=1:1:M
    BC(1,i)=ceil(log2(abs(delta_DC(1,i))+1));
end
Magnitude=zeros(1,M);
for i=1:1:M
    Magnitude(1,i)=abs(delta_DC(1,i));%BC?
end
figure(2);
hist(delta_DC,unique(delta_DC));title('DeltaDC');
figure(3);
hist(DC,unique(DC));title('DC');
%%подсчет встречаемости
%%подсчет энтропии (у hist есть встречаемость (возвращаемые
параметры,
%%число раз поделить на общее число
[counts,centers]=hist(delta_DC,unique(delta_DC));
[counts1,centers1]=hist(DC,unique(DC));
%для интропии поделить на их количество counts
length_DC=length(counts);
% подсчет энтропии для _dc
H_sum_DC=zeros(1,length_DC);
for i=1:1:length_DC
    p=counts1(1,i)/M;

```

```

        sum2=p*log2(p);
        H_sum_DC(1,i)=sum2;

end
C=0;

for i=1:1:length_DC
    A=H_sum_DC(1,i);
    B=A+C;
    C=B;

end
entropi_DC=-1*B;
% подсчет энтропии для delta_dc
H_sum_DC=zeros(1,length_DC);
for i=1:1:length_DC
    p=counts(1,i)/M;
    sum1=p*log2(p);
    H_sum_DC(1,i)=sum1;

end
C=0;

for i=1:1:length_DC
    A=H_sum_DC(1,i);
    B=A+C;
    C=B;

end
entropi_delta_DC=-1*B;

% % % % % % % % % % % % % % пункт 3 с кодирование длин серий

% считаем из прямого квандования блоков 8*8 элементы в одномерный массив
% зиг загообразным обходом
k=1;
M2=(H*W)-M;
AC=zeros(1,M2);
for i=1:8:H
    for j=1:8:W

        AC(1,k)=Qantov(i,j+1);%1
        k=k+1;
        AC(1,k)=Qantov(i+1,j);%2
        k=k+1;
        AC(1,k)=Qantov(i+2,j);%3
        k=k+1;
        AC(1,k)=Qantov(i+1,j+1);%4
        k=k+1;
    end
end

```

```
AC(1,k)=Qantov(i,j+2);%5
    k=k+1;
AC(1,k)=Qantov(i,j+3);%6
    k=k+1;
AC(1,k)=Qantov(i+1,j+2);%7
    k=k+1;
AC(1,k)=Qantov(i+2,j+1);%8
    k=k+1;
AC(1,k)=Qantov(i+3,j);%9
    k=k+1;
AC(1,k)=Qantov(i+4,j);%10
    k=k+1;
AC(1,k)=Qantov(i+3,j+1);%11
    k=k+1;
AC(1,k)=Qantov(i+2,j+2);%12
    k=k+1;
AC(1,k)=Qantov(i+1,j+3);%13
    k=k+1;
AC(1,k)=Qantov(i,j+4);%14
    k=k+1;
AC(1,k)=Qantov(i,j+5);%15
    k=k+1;
AC(1,k)=Qantov(i+1,j+4);%16
    k=k+1;
AC(1,k)=Qantov(i+2,j+3);%17
    k=k+1;
AC(1,k)=Qantov(i+3,j+2);%18
    k=k+1;
AC(1,k)=Qantov(i+4,j+1);%19
    k=k+1;
AC(1,k)=Qantov(i+5,j);%20
    k=k+1;
AC(1,k)=Qantov(i+6,j);%21
    k=k+1;
AC(1,k)=Qantov(i+5,j+1);%22
    k=k+1;
AC(1,k)=Qantov(i+4,j+2);%23
    k=k+1;
AC(1,k)=Qantov(i+3,j+3);%24
    k=k+1;
AC(1,k)=Qantov(i+2,j+4);%25
    k=k+1;
AC(1,k)=Qantov(i+1,j+5);%26
    k=k+1;
AC(1,k)=Qantov(i,j+6);%27
    k=k+1;
AC(1,k)=Qantov(i,j+7);%28
    k=k+1;
AC(1,k)=Qantov(i+1,j+6);%29
    k=k+1;
AC(1,k)=Qantov(i+2,j+5);%30
```

```

    k=k+1;
AC(1,k)=Qantov(i+3,j+4);%31
    k=k+1;
AC(1,k)=Qantov(i+4,j+3);%32
    k=k+1;
AC(1,k)=Qantov(i+5,j+2);%33
    k=k+1;
AC(1,k)=Qantov(i+6,j+1);%34
    k=k+1;
AC(1,k)=Qantov(i+7,j);%35
    k=k+1;
AC(1,k)=Qantov(i+7,j+1);%36
    k=k+1;
AC(1,k)=Qantov(i+6,j+2);%37
    k=k+1;
AC(1,k)=Qantov(i+5,j+3);%38
    k=k+1;
AC(1,k)=Qantov(i+4,j+4);%39
    k=k+1;
AC(1,k)=Qantov(i+3,j+5);%40
    k=k+1;
AC(1,k)=Qantov(i+2,j+6);%41
    k=k+1;
AC(1,k)=Qantov(i+1,j+7);%42
    k=k+1;
AC(1,k)=Qantov(i+2,j+7);%43
    k=k+1;
AC(1,k)=Qantov(i+3,j+6);%44
    k=k+1;
AC(1,k)=Qantov(i+4,j+5);%45
    k=k+1;
AC(1,k)=Qantov(i+5,j+4);%46
    k=k+1;
AC(1,k)=Qantov(i+6,j+3);%47
    k=k+1;
AC(1,k)=Qantov(i+7,j+2);%48
    k=k+1;
AC(1,k)=Qantov(i+7,j+3);%49
    k=k+1;

AC(1,k)=Qantov(i+6,j+4);%50
    k=k+1;
AC(1,k)=Qantov(i+5,j+5);%51
    k=k+1;
AC(1,k)=Qantov(i+4,j+6);%52
    k=k+1;
AC(1,k)=Qantov(i+3,j+7);%53
    k=k+1;
AC(1,k)=Qantov(i+4,j+7);%54
    k=k+1;
AC(1,k)=Qantov(i+5,j+6);%55

```

```

        k=k+1;
        AC(1,k)=Qantov(i+6,j+5);%56
        k=k+1;
        AC(1,k)=Qantov(i+7,j+4);%57
        k=k+1;
        AC(1,k)=Qantov(i+7,j+5);%58
        k=k+1;
        AC(1,k)=Qantov(i+6,j+6);%59
        k=k+1;
        AC(1,k)=Qantov(i+5,j+7);%60
        k=k+1;
        AC(1,k)=Qantov(i+6,j+7);%61
        k=k+1;
        AC(1,k)=Qantov(i+7,j+6);%61
        k=k+1;
        AC(1,k)=Qantov(i+7,j+7);%63
        k=k+1;

```

```

    end

```

```

end

```

```

% % % % % % % % формирование пар run/level

```

```

run=zeros(M2,1);

```

```

level=zeros(M2,1);

```

```

b=0;

```

```

w=1;

```

```

for i=1:1:M2

```

```

    if rem(i,63)==0

```

```

        if run(w-3,1)==15

```

```

            level(w-3,1)=0;

```

```

            run(w-3,1)=0;

```

```

            w=w-2;

```

```

            b=0;

```

```

            n=0;

```

```

        elseif run(w-2,1)==15

```

```

            level(w-2,1)=0;

```

```

            run(w-2,1)=0;

```

```

            w=w-1;

```

```

            b=0;

```

```

            n=0;

```

```

        elseif run(w-1,1)==15

```

```

            level(w-1,1)=0;

```

```

            run(w-1,1)=0;

```

```

        b=0;

```

```

        n=0;

```

```

    else

```

```

        level(w,1)=0;

```

```

        run(w,1)=0;

```



```
w=w+1;
b=0;
n=0;
end
else

if AC(1,i)~=0
    if i>1
        if AC(1,i-1)~=0
            b=0;
            n=0;

        end
        end
        n=0;
        level(w,1)=AC(1,i);
        run(w,1)=b;
        w=w+1;
    else


        n=n+1;
        b=n;
        if b==16

            run(w,1)=15;
            level(w,1)=0;
            w=w+1;
            b=0;
            n=0;

        end


    end
end
end

% запишем в удобную форму
RUN_LEVEL=zeros(w-1,2);
for i=1:1:w-1
    RUN_LEVEL(i,1)=run(i,1);
    RUN_LEVEL(i,2)=level(i,1);
end

% % % % % % % % % % % % % % % формирование новых пар с левел ВС и
амплитуда
% битовые категории на основании левел
BC_level=zeros(1,w);
n=1;
for i=1:1:w

    if level(i,1)~=0
```

```

        BC_level(1,n)=ceil(log2(abs(level(i,1))+1));
        n=n+1;
    end
end
% амплитуда левел
v=1;
Magnitude_level=zeros(1,n);
for i=1:1:n
    if level(i,1)~=0
        Magnitude_level(1,v)=abs(level(i,1));%BC?
        v=v+1;
    end
end

% определить соотношение 3(d) пункт
% figure(4);
% hist(BC,unique(BC));
[counts2,centers2]=hist(BC,unique(BC));

summ_BC_deltaDC=sum(ceil(-1*log2(counts2/M)).*counts2);%I(BC_dc)
% figure(5);
% hist(Magnitude,unique(Magnitude));
Summ_Magnitude_deltaDC=sum(BC);%,bit magnitud(delta_dc)
Summ_Magnitude_level=sum(BC_level);%,bit magnitud(level)
run2_bc_level=zeros(61306,2);
o=1;
for i=1:1:68262
    if run(i,1)==0 && level(i,1)==0
        o=o-1;
    else
        run2_bc_level(o,1)=run(i,1);
        run2_bc_level(o,2)=BC_level(1,i);
        o=o+1;
    end
end

end

% % % % для пар run(bc(level)) посчитать I
% создадим таблицу 16*16 где будем строка вследел столец ран и
% когда встретили значение прибавить на 1

BC_level_Run_table=zeros(16,16);
MMM=61306;
for i=1:1:MMM

BC_level_Run_table((run2_bc_level(i,1)+1),run2_bc_level(i,2))=BC_level_Run_table((run2_bc_level(i,1)+1),run2_bc_level(i,2))+1;
end
BCLR=zeros(1,109);
g=1;

```

```

for i=1:1:16
    for j=1:1:16
        if BC_level_Run_table(i,j)==0;
            else
                BCLR(1,g)= BC_level_Run_table(i,j);
                g=g+1;
            end
        end
    end
end
summ_BC_level1=sum(ceil(-1*log2(BCLR/MMM)).*BCLR);%I(BC_level)
counts будет одномерный для предыдущего

% % % % % % % % % % %ИТОГ

BIT_ALL=summ_BC_level1+Summ_Magnitude_deltaDC+Summ_Magnitude_level+
summ_BC_deltaDC;

BIT_ALL_proc=BIT_ALL/100;
I_BC_delta_DC_proc=summ_BC_deltaDC/BIT_ALL_proc;
I_magnitude_delta_DC_proc=Summ_Magnitude_deltaDC/BIT_ALL_proc;
I_magnitude_level_proc=Summ_Magnitude_level/BIT_ALL_proc;
I_BC_level_proc=summ_BC_level1/BIT_ALL_proc;

```