

ANEXA V

TRANSFORMATA COSINUS DISCRETĂ

Transformata Cosinus Discretă (DCT) se utilizează pentru compresia imaginilor digitale prin algoritmi de tip JPEG, MPEG.

I. Aplicați transformata DCT bidimensională, directă și inversă, pe matrici de 4*4 elemente și observați efectul funcției de rotunjire a numerelor zecimale asupra matricii refăcute.

```
help dct2
a=[1 2 3 4; 2 3 4 5; 3 4 5 6; 4 5 6 7]
d=dct2(a)
dr=round(d)
i=idct2(dr)
ir=round(i)
```

II. Deduceți vectorul obținut după transformare prin scanare în zig-zag. Ce observați?

```
v=[dr(1,1), dr(1,2), dr(2,1), dr(3,1), dr(2,2), dr(1,3), dr(1,4), dr(2,3),
dr(3,2), dr(4,1), dr(4,2), dr(3,3), dr(2,4), dr(3,4), dr(4,3), dr(4,4)]
```

III. Care este numărul de biți transmiși cu și fără compresie? Calculați rata de compresie în bpp și procentual.

```
nib=4*4*8
ncb=1+4+3*8
rc=nib/ncb % (bpp)
rp=1/rc*100 % (%)
```

IV. Analizați o imagine color:

```
[x,map]=imread('e:\scd\imag\start.bmp','bmp');
s=size(x)
l=s(1),c=s(2)
nip=l*c
nie=np*3
max(max(max(x)))
min(min(min(x)))
nib=ne*8
nikB=np/1024
image(x);
```

V. Aplicați 2D-DCT pe matricile de imagine (RGB). Câte valori nule conține vectorul transformat după rotunjire?

```
for j=1:3
    t(:,j)=dct2(x(:,j));
end
max(max(max(t)))
min(min(min(t)))
zt=0;
```

```

for i=1:l
    for j=1:c
        for k=1:3
            if round(t(i,j,k))==0
                zt=zt+1;
            end
        end
    end
end
zt
nie-zt

```

VI. Modificați pragul de cuantizare și deduceți distorsiunea pătratică medie și raportul semnal-zgomot în dB. Câte valori nule sunt în matricea tridimensională după cuantizare? Estimați rata de compresie.

```

p=1/10;
tr=round(t*p);
ztr=0;
for i=1:l
    for j=1:c
        for k=1:3
            if tr(i,j,k)==0
                ztr=ztr+1;
            end
        end
    end
end

```

```

        end
ztr
nie-ztr
for j=1:3;
    r(:,j)=round(idct2(tr(:,j)/p));
end;
image(r);
max(max(max(r)))
min(min(min(r)))
dr=double(r);
dx=double(x);
max(max(max(dr)))
min(min(min(dr)))
max(max(max(dx)))
min(min(min(dx)))
d=dr-dx;
M=max(max(max(d)));
m=min(min(min(d)));
id=(d-m)/(M-m);
image(id);
e=0;s=0;
for i=1:l
    for j=1:c
        for k=1:3
            s=s+(dx(i,j,k))^2;
            e=e+(d(i,j,k))^2;
        end
    end
end

```

```
end
end
distors=100*e/s
snr=10*log10(s/e) % dB
```

VII. Proiectați un cod de compresie a imaginilor digitale bazat pe transformata 2D-DCT aplicată submatricilor pătratice de 16 sau 64 pixeli, cu lungime variabilă a cuvintelor de cod. Implementați algoritmul de codare. Deduceți valorile teoretice și cele experimentale ale ratei de compresie și raportului semnal/zgomot.