

Elaborazione di Segnali Multimediali
a.a. 2017/2018

La compressione Soluzioni

1 Codifica nel dominio DCT

Il taglio dei coefficienti sotto una soglia pari a gamma può essere realizzato nel seguente modo:

```
y = blkproc(x,[8 8],@dct2);           % DCT a blocchi
yth = blkproc(y,[8 8],@azzerata,gamma); % azzeramento valori sotto soglia
perc = sum(yth(:)==0)/prod(size(x))*100;
fprintf('percentuale coeff. posti a zero con passo %1d: %1.2f', Delta, perc);
xr = blkproc(yth,[8 8],@idct2);       % IDCT a blocchi

function y = azzerata(x,gamma);

x(abs(x) > gamma) = 0;
```

2 Lo standard JPEG

Di seguito trovate la funzione che simula il comportamento dello standard JPEG in matlab:

```
function y = simJPEG(x, L);

[M N] = size(x);

% Passo 1 - Sottrarre la media dell'immagine originale
x = x - 128;
```

```
% Passo 2 - Estensione ai bordi
m = 8*ceil(M/8) - M;
if m>0
    x = padarray(x, [m 0], 'replicate', 'post');
end

n = 8*ceil(N/8) - N;
if n>0
    x = padarray(x, [0 n], 'replicate', 'post');
end

% Passo 3 - DCT a blocchi 8x8
xdct = blkproc(x, [8 8], @dct2);

% Passo 4 - Quantizzazione uniforme
Q = [ 16 11 10 16 24 40 51 61
      12 12 14 19 26 58 60 55
      14 13 16 24 40 57 69 56
      14 17 22 29 51 87 80 62
      18 22 37 56 68 109 103 77
      24 35 55 64 81 104 113 92
      49 64 78 87 103 121 120 101
      72 92 95 98 112 100 103 99
    ];

if L > 0 && L < 50
    s = 5000 / L;
elseif L <= 50 && L < 100
    s = 200 - 2*L;
else
    s = 1;
end
Qs = (s .* Q + 50) ./ 100;
xq = blkproc(xdct, [8 8], @qu, Qs);

% Passo 5 - IDCT a blocchi 8x8
y = blkproc(xq, [8 8], @idct2);

% Passo 6 - Ripristino delle dimensioni e dei valori reali
y = y(1:M, 1:N);
y = y + 128;

function y = qu(x, Qs)
y = Qs .* round( x ./ Qs );
```

3 Esercizi proposti

1. *Pseudo-codifica a basso bit-rate.*

```

fp = fopen('lena.y','rb');
x = fread(fp,[512 512],'uint8'); x = x';
mse_list = ones(4,1); snr_list = zeros(4,1);
varx = std2(x).^2;
y1 = blkproc(x,[8 8],@elab1);
y2 = blkproc(x,[16 16],@elab2);
y3 = blkproc(x,[32 32],@elab3);
y4 = blkproc(x,[32 32],@elab4);
mse_list(1) = mean((x(:)-y1(:)).^2);
mse_list(2) = mean((x(:)-y2(:)).^2);
mse_list(3) = mean((x(:)-y3(:)).^2);
mse_list(4) = mean((x(:)-y4(:)).^2);
snr_list = 10.*log10(varx./mse_list);

figure;
subplot(2,2,1); imshow(y1,[0 255]); title(['snr=' num2str(snr_list(1))]);
subplot(2,2,2); imshow(y2,[0 255]); title(['snr=' num2str(snr_list(2))]);
subplot(2,2,3); imshow(y3,[0 255]); title(['snr=' num2str(snr_list(3))]);
subplot(2,2,4); imshow(y4,[0 255]); title(['snr=' num2str(snr_list(4))]);

function y = elab1(x)

xdct = dct2(x);
mask = zeros(size(xdct)); mask(1,1) = 1;
ydct = mask.*xdct;
y = idct2(ydct);

function y = elab2(x)

xdct = dct2(x);
mask = zeros(size(xdct));
mask(1:2,1:2) = [1 1; 1 0];
ydct = mask.*xdct;
y = idct2(ydct);

function y = elab3(x)

xdct = dct2(x);
[M N] = size(xdct);
mask = [fliplr(triu(ones(5,5))), zeros(5,N-5); zeros(M-5,5), zeros(M-5,N-5)];
ydct = mask.*xdct;
y = idct2(ydct);

function y = elab4(x)

xdct = dct2(x);
[values indexs] = sort(abs(xdct(:)));
h = numel(indexs);
indexs = indexs(1:h-10);
ydct = xdct; ydct(indexs) = 0;
y = idct2(ydct);

```

2. Compressione DCT su blocco con diversi rapporti d'aspetto.

```
fid = fopen('peppers.y','r');
x = fread(fid,[512 512],'uint8'); x = x';
figure; imshow(x,[0 255]); title('immagine originale');
K = [8 4 2 1]; L = [8 16 32 64];
for i = 1:length(K),
    y = codec(x,K(i),L(i));
    mse(i) = mean2((x-y).^2);
    figure; imshow(y,[0 255]);
    title(['Blocco: ' num2str(K(i)) 'x' num2str(L(i)) ' mse= ' num2str(mse(i))]);
end

function y = codec(x,K,L)
xc = blkproc(x,[K L],@codifica);
yq = quantizzazione(xc);
y = idct2(yq);

function y = codifica(x)
z = dct2(x);
y = zeros(size(z));
blocco_mod = abs(z);
for h = 1:4,
    k = max(blocco_mod(:));
    [i,j] = find(blocco_mod==k);
    y(i,j) = z(i,j);
    blocco_mod(i,j) = 0;
end

function y = quantizzazione(x)
delta = 16;
y = delta*round(x./delta);
```

3. *Codifica con Trasformata Wavelet.*

```
fid = fopen('peppers.y','r');
x = fread(fid,[512 512],'uint8'); x = x';
figure; imshow(x,[0 255]); title('immagine originale');

[M N] = size(x);
lev = 5; soglia = 20;

% trasformata dell'immagine
w = fwt2d(x,lev);

% eliminazione coefficienti wavelet
y = w;
mask = abs(w) < soglia;
y(mask)=0;
y(1:M/(2^lev), 1:N/(2^lev)) = w(1:M/(2^lev), 1:N/(2^lev));
rho = sum(mask(:))/prod(size(x)) *100;
figure(2); imshow(mask,[]); title('maschera');

% antitrasformata dell'immagine
xw = iwt2d(y,lev);
figure(3); imshow(xw,[0 255]);
title(sprintf('Eliminato il %1.2f %% dei coefficienti', rho));
```