

Le trasformate DCT e Wavelet Soluzioni

1 La Trasformata coseno bidimensionale

Per realizzare la DCT a blocchi si può usare la funzione blkproc:

```
y = blkproc(x,[K L],@dct2);    % DCT a blocchi diretta  
x = blkproc(y,[K L],@idct2);  % DCT a blocchi inversa
```

1. Watermarking nel dominio DCT.

```
clear all; close all; clc;  
  
x = double(imread('cowper.bmp'));  
figure; imshow(x,[]); title('Immagine originale');  
[M N] = size(x);  
  
xdct = dct2(x);  
L = 16350; alfa = 0.2;  
w = randn(size(x));  
c = xdct + alfa*abs(xdct).*w;  
  
mask1 = zeros(M,N);  
mask1(1:180,1:180) = fliplr(triu(ones(180)));  
mask2 = zeros(M,N);  
mask2(1:255,1:255) = fliplr(triu(ones(255)));  
mask = mask2-mask1;  
figure; imshow(mask); title('Maschera coefficienti da modificare');  
  
y=xdct.*not(mask)+mask.*c;  
xrec=idct2(y);  
  
figure; imshow(xrec,[]); title('Immagine ricostruita');  
MSE=mean2((x-xrec).^2)
```

2. Filtraggio nel dominio DCT.

```
clear; close all; clc;
x = double(imread('libro.jpg'));
[M,N] = size(x);

% Per questa immagine non è necessaria l'estensione

s = blkproc(x, [8 8], @std2);
s = s.^2; % matrice della varianza calcolata su blocchi 8x8

%calcolo soglie:
var = sort(s(:));
n1 = round(0.25*length(var)); % n.ro di campioni prima soglia s1
s1 = var(end-n1); % soglia s1 (superata dal 25%)
n2 = round(0.05*length(var)); % n.ro di campioni seconda soglia s2
s2 = var(end-n2); % soglia s2 (superata dal 5%)

for k = 1:5,
    y = blkproc(x,[8 8],@elabora,s1,s2,k);
    mse(k) = mean2((y - x).^2);
end
figure; plot(1:5,mse,'-*'); xlabel('k'); ylabel('mse');
```

```
function y = elabora(x,s1,s2,k)

[M N] = size(x);
s = std2(x)^2;

z = dct2(x);
if s <= s1
    y = zeros(M, N);
    y(1,1) = z(1,1);
else if (s1 <= s & s <= s2)
    mask = fliplr(triu(ones(k)));
    mask = padarray(mask,[8-k 8-k],'post');
    y = z.*mask;
else
    y = z;
end
end
y = idct2(y);
```

2 La Trasformata Wavelet bidimensionale

Di seguito la trasformata Wavelet diretta:

```
function W = fwt2d(X,lev)
% FWT2d trasformata wavelet 2D diretta
% W = fwt2d(X,lev) realizza la trasformata su lev livelli dell'ingresso X

W = X;
[M,N]=size(W);
for l=1:lev,
    for i=1:M                % filtraggio lungo le righe
        riga=W(i,1:N);      % riga da trasformare
        [wtriga]=fwt(riga,1); % WT monodimensionale con lev=1
        W(i,1:N)=wtriga;    % posizionamento
    end
    for j=1:N,                % filtraggio lungo le colonne
        riga = W(1:M,j);     % colonna da trasformare
        [wtriga] = fwt(riga',1);
        W(1:N,j) = wtriga';  % posizionamento
    end;
    M = M/2;                  % dimezza le dimensioni
    N = N/2;                  % per il passo successivo
end;
```

e quella inversa:

```
function Y = iwt2d(W,lev);
% IWT2d trasformata wavelet 2D inversa
% Y = iwt2d(W,lev) inverte la trasformata su lev livelli dell'ingresso W

[M,N] = size(W);
N = N/2^(lev-1); M = M/2^(lev-1);
for l=1:lev,
    for i=1:M                % filtraggio lungo le righe
        riga=W(i,1:N);      % riga da anti-trasformare
        [wtriga]=iwt(riga,1); % IWT monodimensionale con lev=1
        W(i,1:N)=wtriga;    % posizionamento
    end
    for j=1:N,                % filtraggio lungo le colonne
        riga = W(1:M,j);     % colonna da anti-trasformare
        [wtriga] = iwt(riga',1);
        W(1:N,j) = wtriga';  % posizionamento
    end;
    M = M*2;                  % raddoppia le dimensioni
    N = N*2;                  % per il passo successivo
end;
Y = W;
```

1. *Visualizzazione wavelet 2D.*

```
function y = wtvview(x,lev);

[M N] = size(x);
for r=1:lev,
    M=M/2; N=N/2;
end
y=x;
for r=1:lev,
    LL=y(1:M,1:N);      LL = fshs(LL);
    LH=y(M+1:2*M,1:N);  LH = fshs(LH);
    HL=y(1:M,N+1:2*N);  HL = fshs(HL);
    HH=y(M+1:2*M,N+1:2*N); HH = fshs(HH);
    W=[LL,HL;LH,HH];
    M=M*2; N=N*2;
    y(1:M,1:N)=W;
end
figure; imshow(W);
```

2. *Watermarking.*

```
x = double(imread('cowper.bmp'));
figure(1); imshow(x,[]); title('immagine originale');
[M N] = size(x);
lev = 3;

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

% generazione del watermark
rand('state',0);
watermark = rand(M,N);

% casting
alpha = 0.2;
T1 = 40;
y = (w + alpha*abs(w).*watermark).*(abs(w) > T1);
y(1:M/(2^lev), 1:N/(2^lev)) = w(1:M/(2^lev), 1:N/(2^lev));
mask = abs(w) > T1;
figure(3); imshow(mask,[]); title('maschera');

% antitrasformata dell'immagine
xw = iwt2d(y,lev);
figure(2); imshow(x,[]); title('immagine marcata');
```