## Approccio algebrico alla deconvoluzione: caso 2D

Nel caso 2D si usa ancora lo stimatore basato sulla pseudoinversa di Moore-Penrose analogamente al caso monodimensionale. Tuttavia il peso computazionale nel caso di immagini potrebbe essere molto oneroso. Se supponiamo di operare con immagini 256x256, la pseudoinversa è una matrice di 2562x2562 che può essere molto difficile da gestire e memorizzare (in double abbiamo 2564x8 bit = 34Gb di memoria RAM occupata).

In questi casi si ricorre a metodi iterativi, oppure alla diagonalizzazione della matrice di sistema.

### Discretizzazione del problema inverso bidimensionale per sistemi convolutivi

Nel caso di un processo bidimensionale LTI, supponendo che  sia costituita da AxB campioni,  sia definita su CxD campioni, si può scrivere:



dove  e  sono le versioni estese rispettivamente di  e , ambedue di dimensione MxN con .

La soluzione alla precedente rappresentazione convolutiva può essere ottenuta ricorrendo a matrici circolanti a blocchi. Nel caso 2D la matrice è circolante a blocchi il cui numero di blocchi dipende dalle dimensioni della maschera del filtro. Ogni elemento  della matrice circolante è a sua volta una matrice circolante.

La matrice H2D è una matrice di Toeplitz a blocchi i cui blocchi sono a loro volta matrici di Toeplitz. La dimensione di **H2D è N2xN2.** Si può dimostrare, sfruttando opportune simmetrie, che tale matrice è diagonalizzabile applicando la trasformata di Fourier come per il caso monodimensionale (vedere Appendice).

### Metodo Least square

L’implementazione del metodo Least square nel caso 2D segue la procedura già applicata al caso monodimensionale. Anche in questo caso la formula, in norma-2, è data da:



dove **H**2D è la matrice di Toeplitz con i blocchi di Toeplitz. Dal momento che **H**2D è diagonalizzabile, possiamo semplificare la procedura (evitando di dover ricostruire la matrice **H**2D) procedendo alla diagonalizzazione e successivamente operando nel dominio della frequenza, seguendo la procedura del caso monodimensionale. In pratica si tratta di trovare una matrice **W** tale che **D** = **WTH2DW** sia diagonale, in cui la matrice diagonale **D** ha dimensione N2xN2.

Essa, come nel caso monodimensionale, contiene la trasformata di fourier di **H** distribuita lungo la diagonale. Quindi, procedendo con l’operazione di reshape su **D**, **si ricostruisce il piano di Fourier**, ovvero la trasformata di fourier 2D di **H**, essendo **H** la trasformata discreta della risposta bidimensionale del sistema. Con queste ipotesi, è possibile sostituire a **D** la trasformata di fourier 2D di **H** in modo da poter applicare la seguente formula: , dove **Y** è la trasformata di fourier 2D dell’immagine di uscita **y**.

### Metodo Least square con regolarizzazione

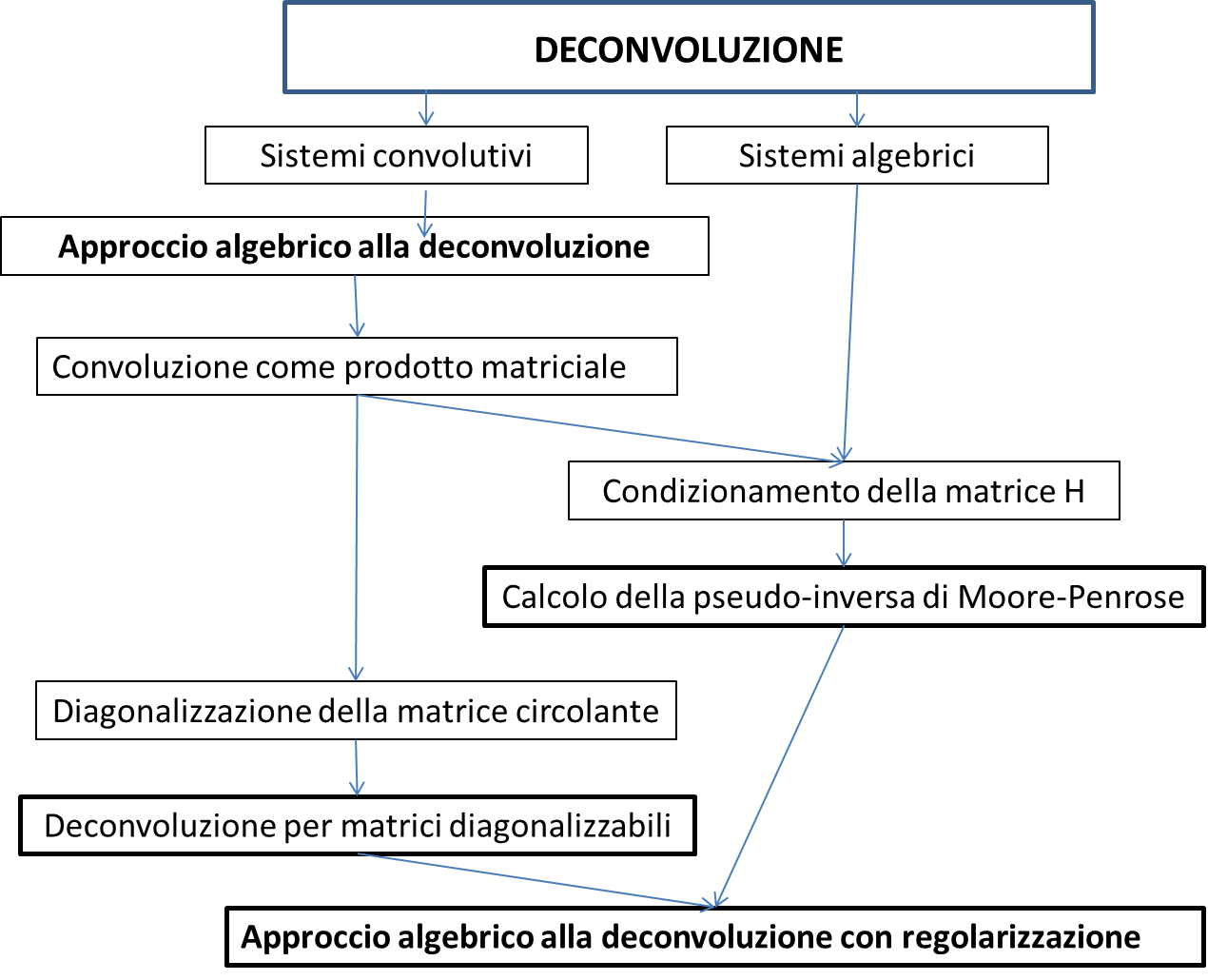
Si può dimostrare che in presenza della regolarizzazione la formula per la stima di **X** nel caso di un sistema di misura convolutivo bidimensionale diventa:



dove **C** è l’operatore di regolarizzazione nel dominio bidimensionale di fourier, **D** diventa la trasformata di fourier del sistema di misura e è il parametro di peso.

Per esempio, se si usa la matrice identità (regolarizzazione di Tikhonov) nella precedente formula, allora non si effettua alcuna discriminazione in frequenza, ma si agisce sull’ampiezza dell’immagine e il risultato è una maggiore robustezza nei confronti del rumore. Se invece la matrice **C** rappresenta un filtro passa alto, allora verranno penalizzate le alte frequenze e quindi dovremmo avere risultati migliori rispetto a Tikhonov tanto maggiori quanto minore è l’estensione di banda del contenuto informativo dell’immagine.

Schema a blocco di sintesi:



### Appendice

Anche nel caso 2D si può rappresentare l’operazione di convoluzione 2D mediante il prodotto matriciale **y** = **H2Dx + n**, dove **H**2D è la matrice circolante. Tuttavia, per fare in modo che la matrice circolante abbia solo due dimensioni, è necessario fare le seguenti considerazioni.

Il prodotto più interno può essere visto come prodotto scalare tra le colonne di **x** e le colonne di **h**:







A questo punto se chiamiamo  l’immagine vettorizzata (ovvero  con Matlab, si può scrivere:





che ha la dimensione 1xN2 in quanto ciascun termine  è un vettore di dimensione 1xN. Quindi l’immagine vettorizzata in uscita sarà:



##### Esercitazione n 2.3.1: problema inverso bidimensionale senza e con regolarizzazione

**1° parte**

clc

clear

close all

%%%%%%%%%%%%%%%%%% PREPARAZIONE PROCESSO DI DEGRADAZIONE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N=256;

dt=1;

I0=zeros(N);

p=randperm(N\*N);

I0(p(1:60))=20;

figure,imagesc(abs(I0)),title('input image')

colormap gray

K0=fftshift(fft2(fftshift(I0)))/N;

%blurring kernel

alpha=5;%std of the gausswin in pixel.

[x,y]=meshgrid([-N/2:N/2-1],[-N/2:N/2-1]);

IH=exp(-((x.^2+y.^2)/alpha^2)); %H=H/mean(H(:));

KH=fftshift(fft2(fftshift(IH)))/N;

Kh=KH+eps;% degradation system frequency domain

figure,imagesc(abs(KH)),title('Blurring Kernel frequency domain')

colormap gray

Kblurr=K0.\*KH; % degraded signal frequency domain

figure,imagesc(abs(Kblurr))

colormap gray

Ib=fftshift(ifft2(ifftshift(Kblurr)))\*N\*N; %degraded signal space domain

figure,imagesc(abs(Ib)),title('degraded image')

colormap gray

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Rumore bianco e rumore di poisson %%%%%%%%%%%%%%%%%%%%

sigma=0.5; %% peso della componente di rumore, sigma = 1 valore elevato

% sigma=10\*(1e-2\*max(abs(Ib(:))));

In=sigma\*randn(N);

% In=sigma\*poissrnd(10,N,N);

KPO=abs(fftshift(fft2(In)));

figure,imagesc(KPO),title('noise image')

colormap gray

Ibn=Ib+In;

figure,imagesc(abs(Ibn)),title('degraded image + noise')

col

Kblurr\_noise=fftshift(fft2(ifftshift(Ibn)))/N;

figure,imagesc(abs(Kblurr)),title('degraded + noise, frequency domain')

colormap gray

ormap gray

figure

subplot(1,3,1),imagesc(abs(I0)),title('I0')

subplot(1,3,2),imagesc(abs(Ib)),title('I0 blurred')

subplot(1,3,3),imagesc(abs(Ibn)),title('I0 blurred with noise')

colormap gray

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%





##### Esercitazione n 2.3.2: problema inverso bidimensionale senza e con regolarizzazione

**2° parte**

Caso con deconvoluzione senza LS

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% II PARTE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% De-convolution is ill-conditioned

%noise free case: Idec1==I0

Idec1=fftshift(ifft2(ifftshift(Kblurr./Kh)))\*N;

%with noise: the problem is ill-conditioned

Idec2=fftshift(ifft2(ifftshift(Kblurr\_noise./Kh)))\*N;

%

% tmp=Kblurr\_noise./abs(Kh).^2;

figure

subplot(1,3,1),imagesc(abs(I0)),title('I0')

subplot(1,3,2),imagesc(abs(Idec1)),title('Deconv Noise free')

subplot(1,3,3),imagesc(abs(Idec2)),title('Deconv Noisy'),colormap gray



**3° parte**

Risolvere il problema LS nei tre casi:

* + Senza regolarizzazione
  + Con regolarizzazione di Tikhonov
  + Con regolarizzazione basata su filtro passa alto

Risultati attesi

Senza regolarizzazione:

.

Con regolarizzazione di Tikhonov: 



Con regolarizzazione basata su filtro passa alto

