

Elaborazione di Segnali e Immagini (ESI) LABORATORIO

Lezione 8

Manuele Bicego

Corso di Laurea in Informatica

Dipartimento di Informatica - Università di Verona

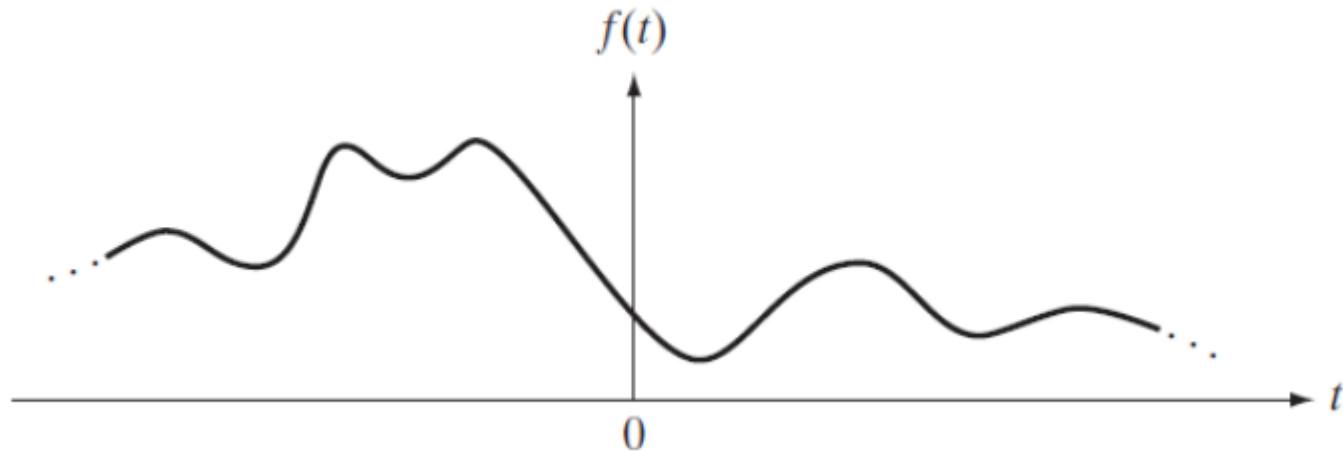
Trasformata di Fourier 2D
Trasformata di Hough per rette

Trasformata di Fourier 2D

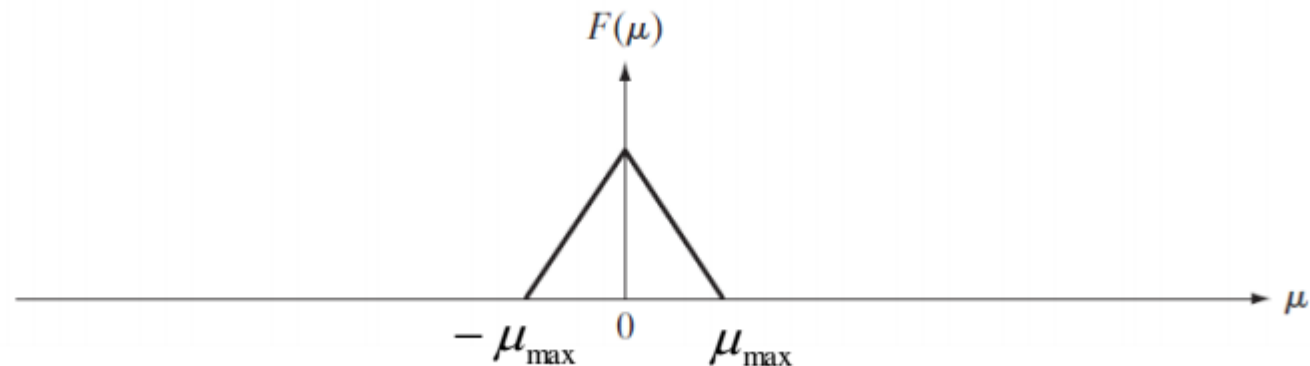
... quello che abbiamo visto nella lezione sulla
trasformata di Fourier 1D

Trasformata di Fourier: descrizione di un segnale nel dominio del tempo e nel dominio delle frequenze

Dominio
del tempo

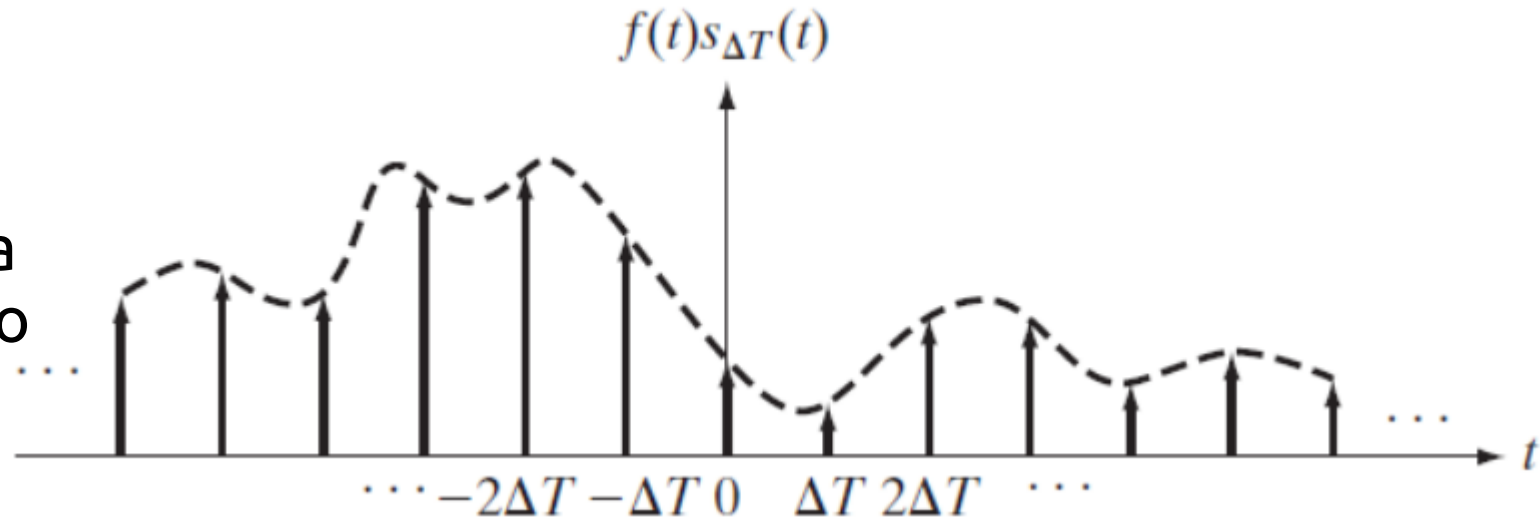


Dominio
delle frequenze

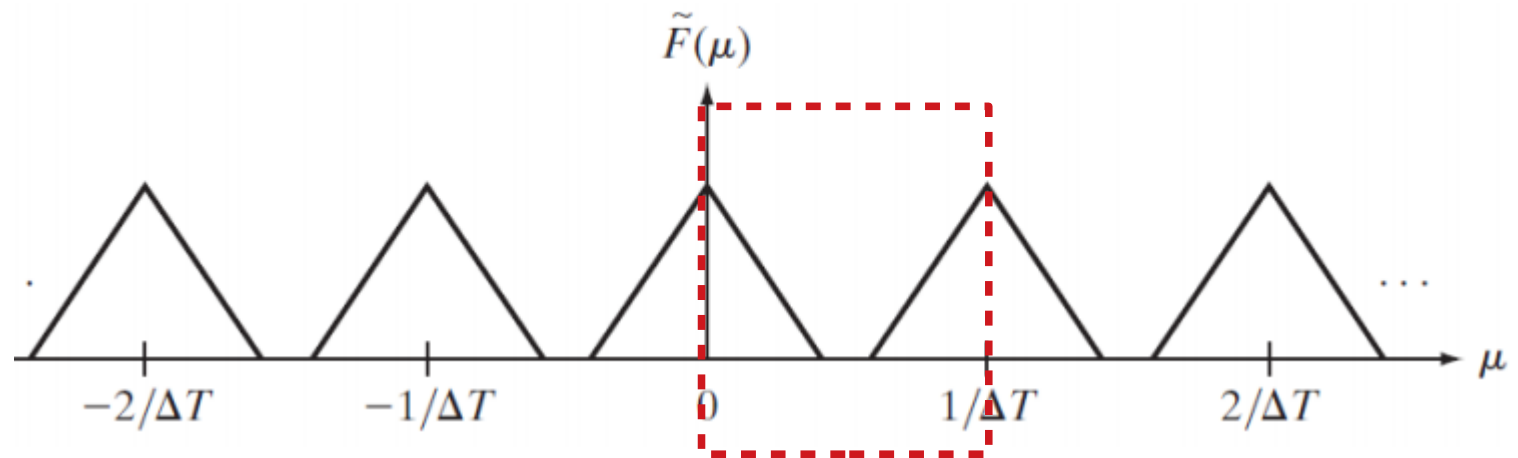


Cosa succede in un calcolatore? Campionamento e trasformata di Fourier discreta (DFT)

Campionamento
del segnale
originale: prendo
un punto ogni
 ΔT (frequenza
di campionamento
 f_s : $1/\Delta T$)

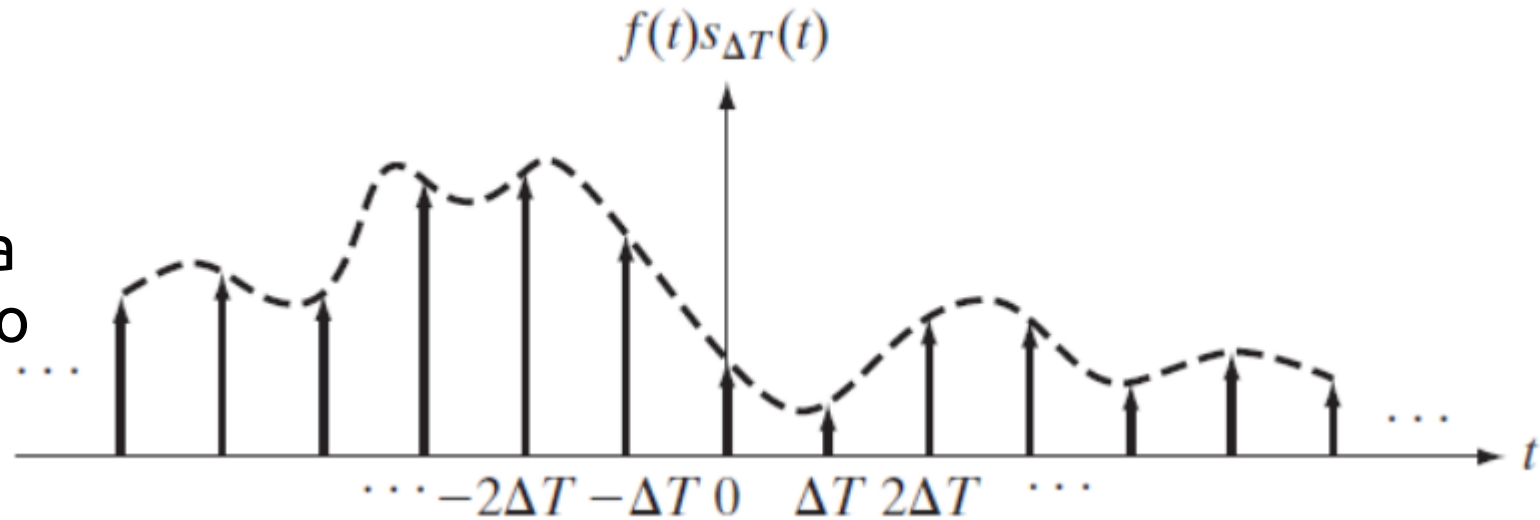


Trasformata di
Fourier discreta:
repliche dello
spettro di
frequenza, una
ogni f_s

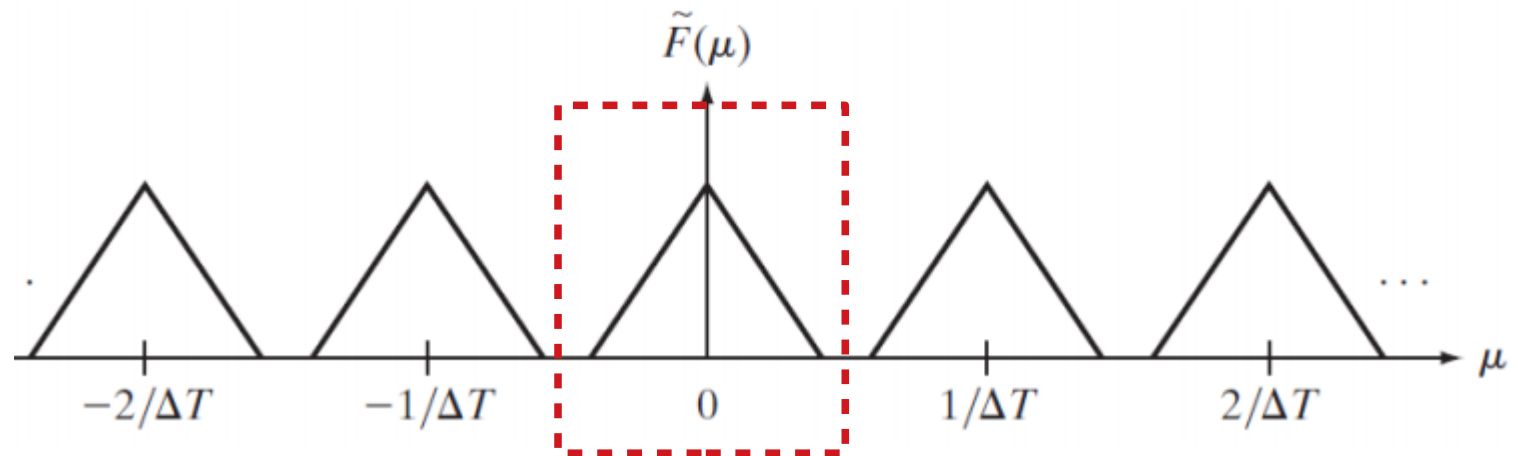


fft di matlab mi ritorna questo pezzo: se il
segnale originale ha N punti ho N bin in
frequenza : da 0 a f_s (con step f_s/N)

Campionamento
del segnale
originale: prendo
un punto ogni
 ΔT (frequenza
di campionamento
 f_s : $1/\Delta T$)



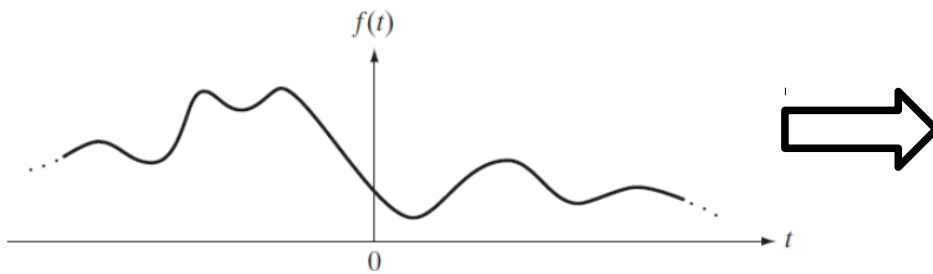
Trasformata di
Fourier discreta:
repliche dello
spettro di
frequenza, una
ogni f_s



Con `fftshift` posso ottenere questo:
N bins da $-f_s/2$ a $f_s/2$ (sempre con
step f_s/N)

Da 1D a 2D

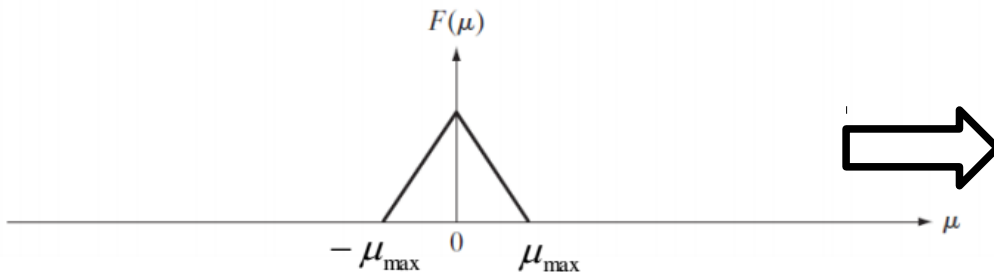
Nella trasformata 2D l'analisi si sviluppa sia in orizzontale che in verticale



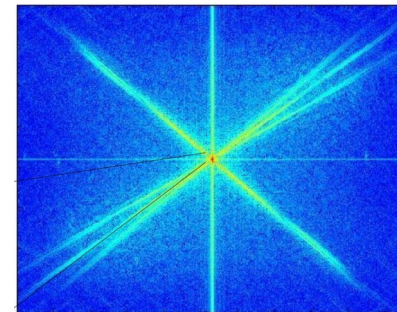
$f(t)$



$f(x,y)$



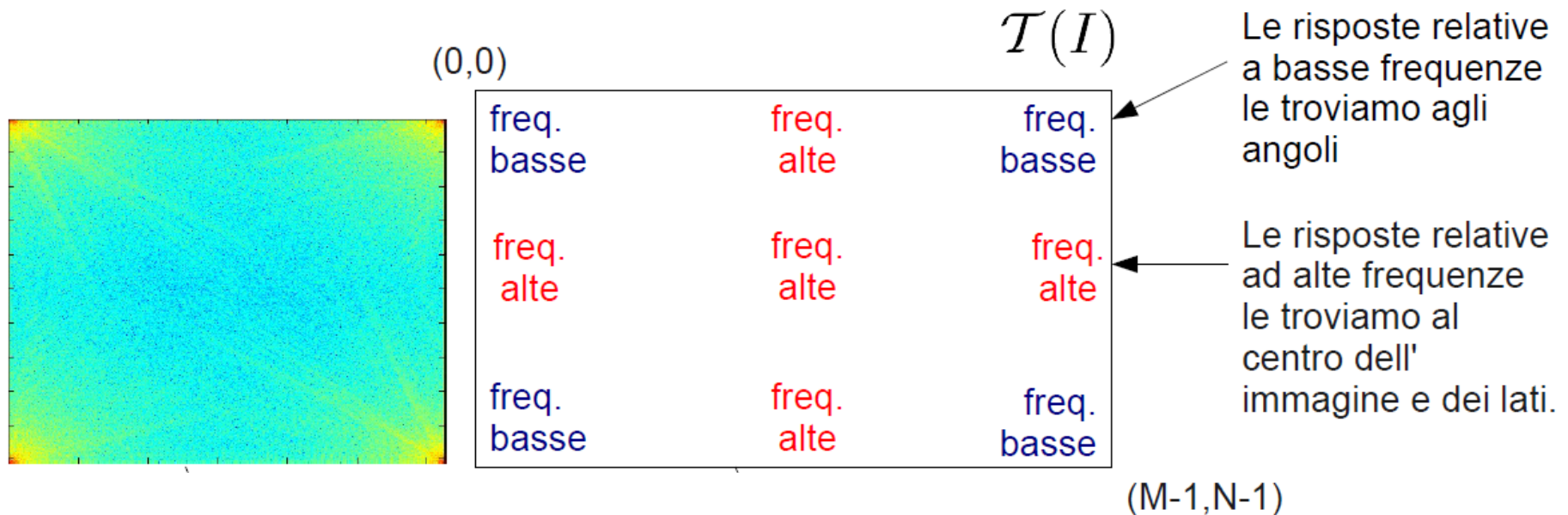
$F(\mu)$



$F(u,v)$

In Matlab

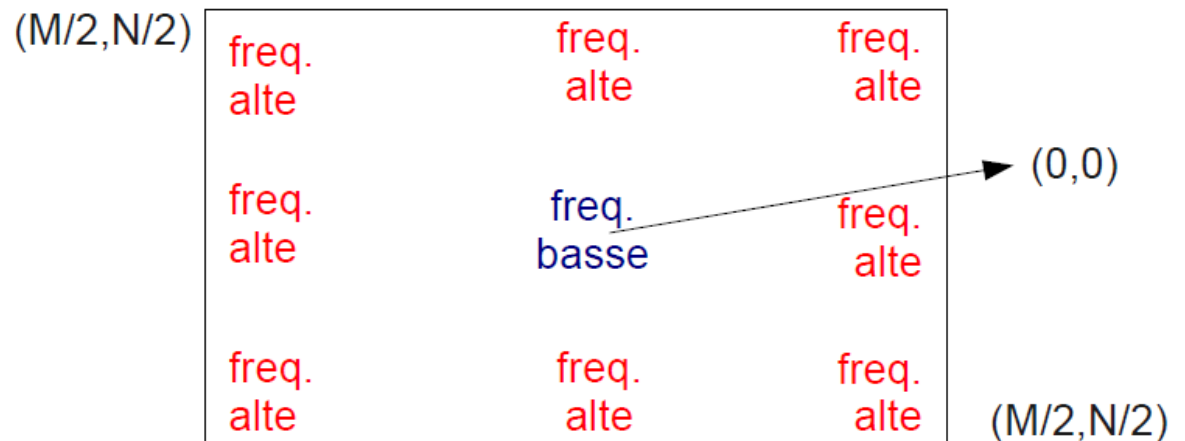
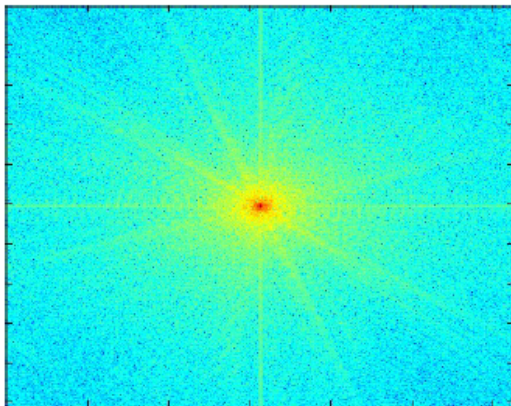
- **`F = fft2(image)`** calcola la DFT 2D
- Come si interpreta?



In Matlab

- Per una migliore visualizzazione si effettua uno shift dello spettro

Fsh = fftshift(F)



In Matlab

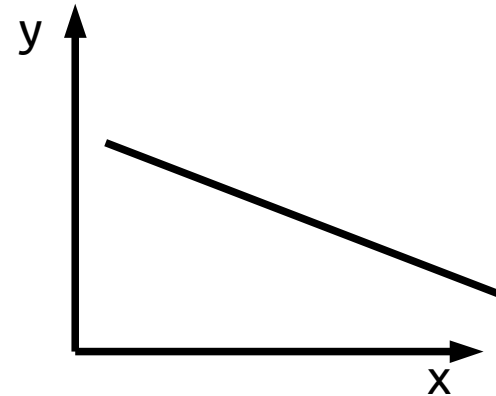
- Analisi di Fourier per un'immagine:
 - Calcolare la trasformata 2D
 - Visualizzare spettro delle ampiezze e spettro di fase
 - Ricostruire l'immagine a partire dalle frequenze
- Esempio nel file
Lezione8_EserciziPrincipali.m

La trasformata di Hough per rette

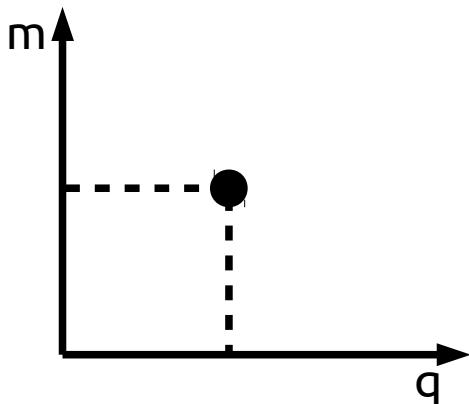
- Serve per identificare le rette in un'immagine
- Idea generale

- Equazione di una retta

$$y = mx + q$$



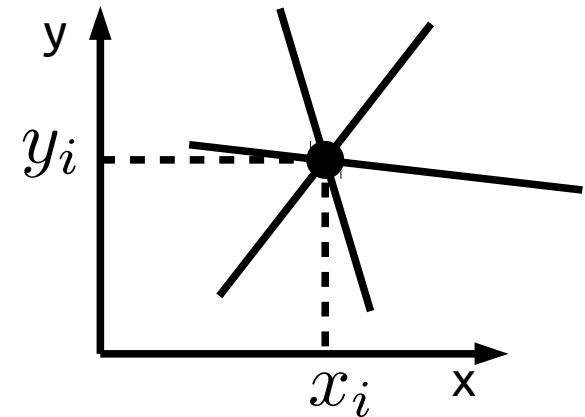
- Se si considera lo spazio dei parametri $[m \ q]$, una retta è rappresentata da un punto



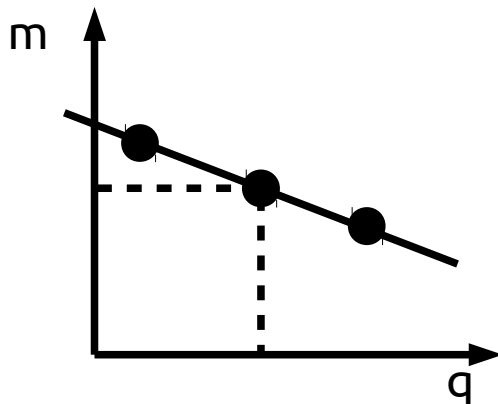
La trasformata di Hough per rette

- Per un punto $[x_i, y_i]$ passano infinite rette, tutte quelle per cui vale

$$y_i = mx_i + q$$



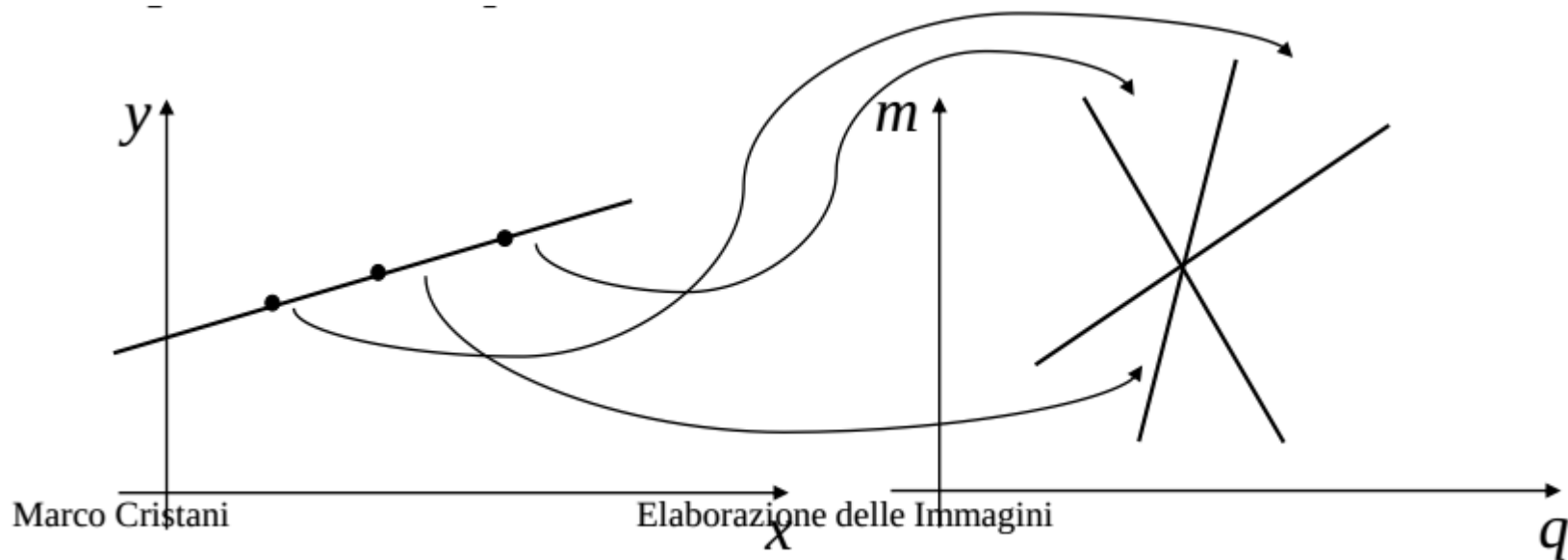
- Nello spazio dei parametri queste rette sono punti disposti lungo una retta



$$q = -x_i m + y_i$$

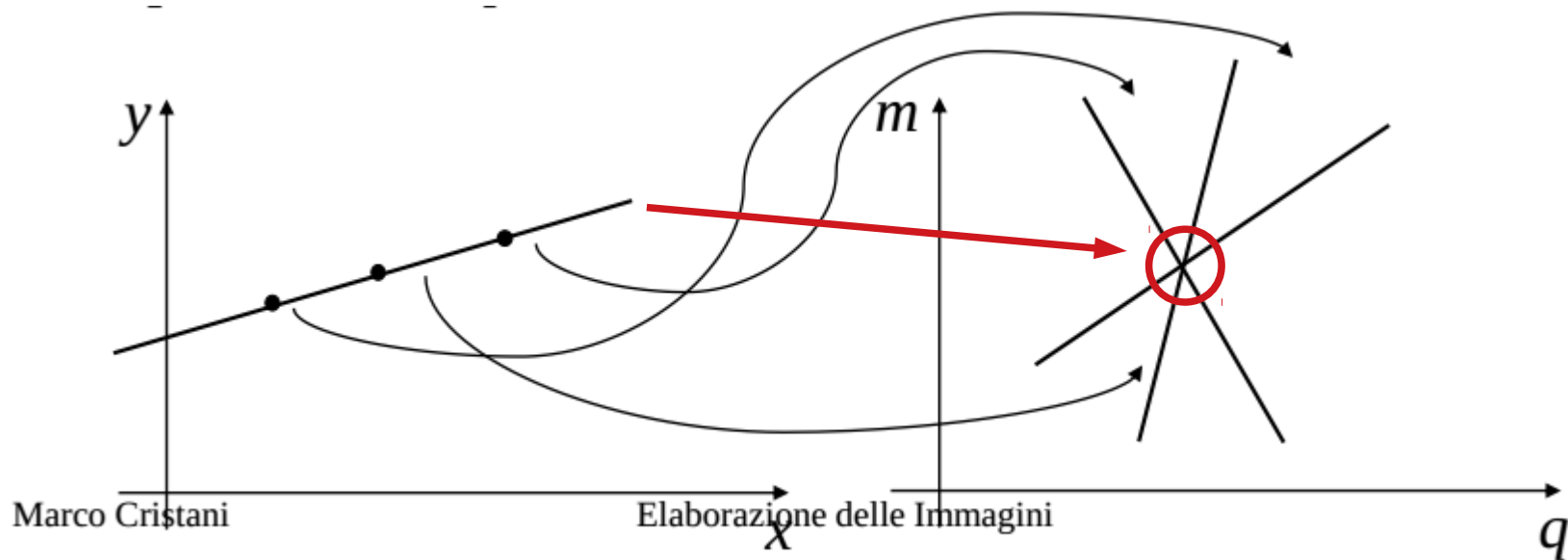
La trasformata di Hough per rette

- Se ho tre punti allineati (che si trovano lungo una retta), ho tre rette nello spazio dei parametri



La trasformata di Hough per rette

- La retta che contiene tutti e tre i punti si può determinare trovando l'intersezione tra le rette nello spazio $[mq]$



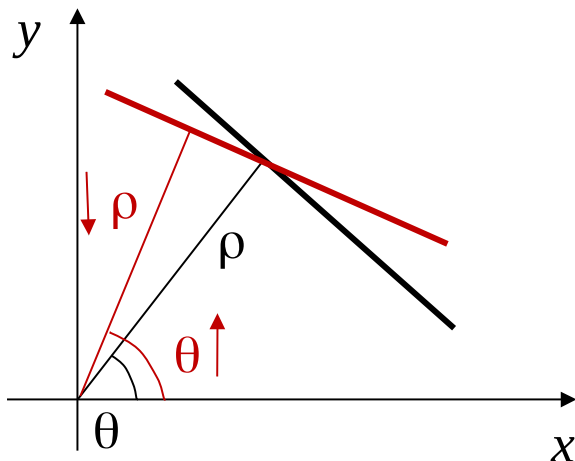
La trasformata di Hough per rette

- Nella trasformata di Hough:
 - Lo spazio dei parametri viene quantizzato, cioè diviso in celle
 - Ogni punto di edge “vota” per un insieme di rette, cioè vota per un insieme di celle
 - Dopo che tutti i punti hanno votato si possono estrarre i massimi: le rette che prendono più voti vincono (sono le rette che “spiegano” più punti)

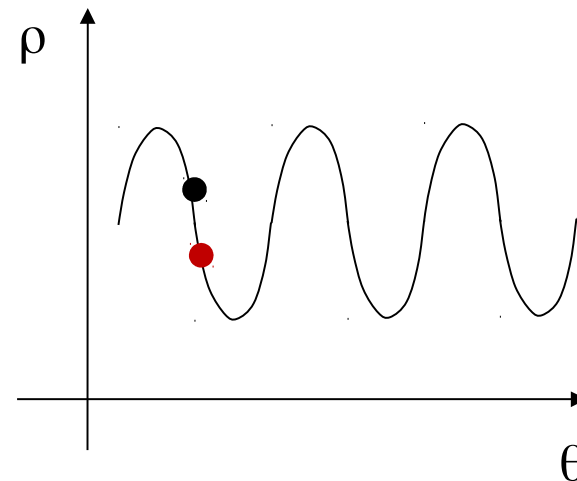
Dettagli

- In pratica si usa la forma polare per la parametrizzazione, perché è più comoda:
 - θ è naturalmente delimitato
 - ρ si può delimitare specificando una regione circolare attorno all'origine (sensato)

$$x \cos \theta + y \sin \theta - \rho = 0$$



$$(x, y) \rightarrow (\rho, \theta)$$



Algoritmo

- Input: I img binaria $M \times N$ (1 edge, 0 no edge)

- Spazio dei parametri

$$\begin{aligned} \rho &\in \left[-\sqrt{M^2 + N^2}, \sqrt{M^2 + N^2} \right] \\ \theta &\in [-90^\circ, 90^\circ] \end{aligned}$$

- Si discretizza (ρ, θ) in (ρ_d, θ_d) usando un passo accettabile per il problema da risolvere (precisione vs. computazione).
- Sia A la matrice $R \times T$ risultante, inizializzata a zero.
- Per ogni pixel $I(x,y) = 1$ e per $h = 1, \dots, T$:
 - sia $\rho = x \cos \theta_d(h) + y \sin \theta_d(h)$
 - trovare l'indice k t.c. $\rho_d(k)$ è l'elemento più vicino a ρ
 - incrementare $A(k,h)$ di una unità
- Trovare tutti i max locali (k_p, h_p) t.c. $A(k_p, h_p) > \tau$ soglia

Esercizi principali

Esercizio 1

- Caricare l'immagine 'imageA1.png'
- Visualizzare (con il logaritmo) lo spettro delle ampiezze centrato
- Estrarre una sola porzione delle frequenze e ricostruire l'immagine a partire da queste frequenze selezionate.
- Osservare cosa succede se cambiamo il contenuto in frequenza scelto. In particolare: cosa succede se escludo le basse frequenze? E se escludo le alte?

Esercizio 1

- Suggerimento per selezionare le frequenze:
 - creare una matrice, della stessa dimensione della `fft2`, contenente tutti zeri;
 - copiare nella posizione della porzione considerata le frequenze corrispondenti dalla matrice della `fft2`

Esercizio 1

- Porzioni da considerare (sulla trasformata shiftata):
 - 1) righe: 260-360, colonne: 410-520
 - 2) righe: 150-460, colonne: 270-670
 - 3) righe: 105-310, colonne: 175-46
 - 4) ...
- Nota: provare anche ad usare **getrect** per selezionare la porzione: la funzione **getrect** serve per estrarre un rettangolo da un'immagine, si veda l'help della funzione

Esercizio 2

- Utilizzare la trasformata di Hough per estrarre le rette dalle seguenti 3 immagini (una sintetica, due reali):
 - "line.jpg"
 - "building.jpg"
 - "Crane.png"
- Utilizzare le funzioni Matlab già implementate per la trasformata (**hough**, **houghpeaks**, **houghlines**) – più info nel file **Lezione8_EserciziPrincipali.m**

Esercizio 2

- Nota importante: l'input è un'immagine binaria che contiene gli edge, da calcolare sull'immagine originale (per estrarre gli edge utilizzare il Canny edge detector `BW = edge(I, 'canny')`)
- Visualizzare le rette estratte sovrapposte all'immagine di partenza
- Valutare come cambiano i risultati al variare dei parametri della trasformata (numero di picchi, soglia, livello di quantizzazione di rho e di theta, etc etc -- si veda l'help delle funzioni).

Esercizi extra

Esercizio 3

- Implementare manualmente la trasformata di Hough, e applicarla all'immagine simulata "line.png"
- Suggerimento: utilizzare la seguente pipeline:
- **PASSO 1.** Definisco due vettori per rho e theta con lo spazio dei parametri discretizzato
 - theta varia tra $-90^\circ \leq \theta < 90^\circ$
 - rho varia tra $-\text{diagonal} \leq \rho \leq \text{diagonal}$
(diagonal è la lunghezza della diagonale -- $\sqrt{M^2+N^2}$)

Esercizio 3

NOTA: Occorre decidere il passo di discretizzazione (quanto approssimato è lo spazio di accumulazione). Inizialmente metterlo a 1, poi provare a vedere cosa succede variandolo

- **PASSO 2.** Creo una matrice 2D che rappresenta lo spazio di accumulazione.
 - La matrice avrà dimensione $R \times T$, dove R e T sono il numero di diversi valori di ρ e di θ

Esercizio 3

- **PASSO 3.** Calcolo lo spazio di accumulazione dei parametri:
 - per tutti i punti dell'immagine che sono a 1
 - per ogni valore di theta
 - calcolo rho
 - trovo il rho più vicino
 - incremento la cella corrispondente
- **PASSO 4.** Trovo il max nella trasformata di Hough

Esercizio 3

- **PASSO 5.** Visualizzo le linee sovrapposte all'immagine di partenza:
 - scelgo due punti x , calcolo le corrispondenti y e uso il comando plot
 - le y si trovano ricavando la y dalla formula
$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$
$$y = (\rho - x \cdot \cos(\theta)) / \sin(\theta)$$

Altri esempi da guardare

- Nel file **Lezione8_EserciziExtra.m** sono presenti ulteriori esercizi (già svolti) per approfondire l'argomento:
 - Esempio 1 - Filtraggio Immagini in Frequenza