

# IMMAGINI - INTRODUZIONE

19-01-2017

Immagine = segnale bidimensionale, matrice di numeri

- monocomponente: i valori dei pixel corrispondono a toni di grigio (0=nero, 1=bianco)
- colore: ogni pixel corrisponde ad una terna di numeri RGB (3 matrici)  
(se tutti i pesi sono uguali  $\rightarrow$  grigio)  $\cdot$   
es. le tre matrici sono uguali

MATLAB:  $I = \text{imread}(\text{'nomefile'})$

$\text{imshow}(I, \text{'filename'}, \text{'fmt'})$  per salvare un'immagine.  
(formato (JPG,...))

$I = \text{rgb2gray}(RGB)$  convertire l'immagine rgb in toni di grigio

$\text{imshow}(I)$  per visualizzare l'immagine

$h = \text{fspecial}(\text{'type'}, \text{parameters})$  per creare il filtro bidimensionale  $h$

$I_f = \text{imfilter}(I, h)$  per filtrare l'immagine  $I$  con il filtro  $h$

## ESEMPLO

`clear all, close all, clc`

`[filename path] = uigetfile('*', 'Choose image to load');`

`filepathname = [path filename];`

`I = imread(filepathname);`

`imshow(I);` % visualizzare l'immagine

`I = I(:, :, 1);` prendo una sola delle tre matrici, un solo colore  
 $\hookrightarrow$  passo a toni di grigio

`I = rgb2gray(I);` lo metto, tanto se  $\bar{e}$   $\frac{255}{255}$  toni di grigio non funziona

`h = fspecial('average');` % di default  $\bar{e}$  una matrice 3x3

`I_f = imfilter(I, h);`

`imshow(I_f, 'Filtered.jpg', 'jpg');`

## FORMATI DI IMMAGINI DIGITALI

Ci sono molti formati, molti algoritmi di compressione  $\rightarrow$  per quale scopo

- codifica dei singoli pixel (bmp): bit assegnati ad ogni pixel dimensione  
 $\hookrightarrow$  grigi: 8 bit, rgb: 24 bit  $\rightarrow$  non posso cambiare codifica senza perdita, grandi
- codifica di sottoblocchi: immagine suddivisa in sottoblocchi, si riduce lo dimensione (ef)  
 $\hookrightarrow$  8 bit per ogni blocco: perdita di dettaglio
- codifica mediante operazioni speciali: in dettagli successivi, da + fini a + grossolani (p)  
 $\hookrightarrow$  in ambito fotografico quelli successivi di dettaglio, ultima compressione

Algoritmi di compressione  $\rightarrow$  spazio di memorizzazione, trasmesso via rete, qualità variabile  
 $\rightarrow$  per immagini mediche STANDARD DICOM

DICOM: raggruppa le info in due parti

- Header DICOM: contiene tanti attributi, info (es. dati del paziente)
- Immagine DICOM: contiene i pixel dell'immagine

$\rightarrow$  i dati del paziente non possono essere separati dall'immagine

MATLAB: dicominfo → per leggere l'header  
dicomread → per leggere l'immagine

```
if ~strcmp (filename (end-2 : end), 'dcm') % se non è uguale a dcm  
    % faccio quello che avevo fatto prima  
    I = imread...  
else  
    I = dicomread (filepathname),  
    info = dicominfo ( " " ),  
end
```

info è uno struct → variabile che contiene all'interno diverse variabili  
per accedere: info.rows per esempio → uso il punto

per visualizzare immagini DICOM:

microdicom → windows  
osirix → mac

- Sono più forti da interpretare dei segnali
  - La matematica è molto + complicata, ma nella pratica non è molto formale
- IMAGE PROCESSING: analisi delle immagini, rispetto alla teoria.

10-05-2012

→ MEDICAL IMAGING: applicazioni usate nell'analisi delle immagini in ambito medico

Evoluzione abbastanza rapida dell'imaging → perché mancano i soldi e si vuole ridurre l'intervento dell'operatore.

↳ 70% clinico; 30% tecnico (in alcuni casi la parte di elaborazione è il 70% dell'asino)

es. mammografia: indice di correlazione tra due radiologi del 40%.

↳ tecniche per il supporto alla diagnostica clinica.

Computer Aided Detection → CAD  
Diagnosis

Il 75% della diagnostica di un ospedale è per immagini (esclusi i prelievi ematici)

- studi morfologici
- studi funzionali
- misure
- segmentazione = scomporre l'immagine nei costituenti fondamentali → riconoscere cosa interessa togliendo tutto il resto
- ricostruzione 3D → tecniche tomografiche → il problema è che non abbiamo supporti di visualizzazione 2D

# OCCHIO UMANO

È un sensore di un campo elettromagnetico in uno banda ben visibile (dal rosso al viola)

↳ è un sensore perfetto nelle bande del visibile

I fotoni penetrano sullo retina (con: intensità luminosa → energia)

(bastonelli: frequenza → colore)

↳ comunica di nervo ottico che manda ai centri della vista (zona occipitale)

↳ l'uso cronico è + spesso

Patologie → ottiche

degenerative (con età → cataratto)

L'occhio è fatto in modo di far entrare i fotoni facendo riflettere il meno possibile

↳ se cambiano gli indici di rifrazione tra cristallino / umor acqueo / umor vitreo si riflettono

+ fotoni → si vede di meno

Sistema visivo: due occhi e centri della vista

↳ due: per avere una visione stereoscopica (un occhio solo: visione prospettica)

centri: vicino al chiasma dove le info destre e sinistre dei due occhi si separano e

viene poi ricomposta nei centri della vista

Intelligenza visiva

→ il cervello non è bravo a misurare, è bravo ad interpretare

⇒ non riusciamo a fare confronti assoluti, ma solo relativi

immagini digitali → processabili numericamente

Nel mondo medico → im. digitali

im. analogiche → digitalizzazione (scanner)

Digitalizzare un'immagine = effettuare un campionamento → ma il 99% delle immagini ha banda

infinita

Nelle immagini si può avere un delta di densità: tutto nero con punto bianco → banda è infinita (nei segnali no)

Immagine = matrice di numeri (se è 2D matrice  $i \times j \times k$ )

a singola punti si chiamano Picture Element (PIXEL) in immagine 3D: Volume Element (VOXEL)

Ogni pixel ha un valore → ha un solo colore ⇒ dimensione fisica del pixel: nel mondo

reale a cosa corrisponde? → dipende

immagini monocromatiche → ogni pixel ha un solo colore che indica il tono di grigio

⇒ la corrispondenza tra numero e colore deve essere costante! (x tutti i dispositivi di visualizzazione)

numeri interi: 0 = nero (B)

$$\left. \begin{array}{l} \vdots \\ \vdots \\ \vdots \end{array} \right\} 2^N - 1 \quad \text{con } N = \text{numero di bit}$$

$N=8 \Rightarrow 255 = \text{bianco (W)}$

↳ immagini radiologiche, ultrasuoni

risoluzione: 10-16 bit  
tc: 12 bit

Immagini a colori: per codificare un colore nel sistema di meno 3

• codifica RGB (rosso, verde, blu)

$[0 \ 0 \ 0] = \text{nero}$  ;  $[255 \ 255 \ 255] = \text{bianco}$

matrice con tre stadi

→ gli algoritmi di compressione comprimono il colore (jpeg)

• immagini bidimensionali: o bianco o nero → immagini logiche, servono per creare delle maschere

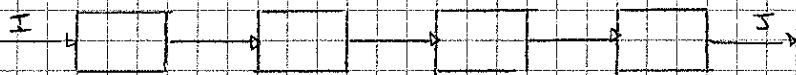
(es. se lo sfondo non interessa lo metto = 0)

• scala di grigio

(ci sono quelle in falso colore → es. medicina nucleare)

• spazi colore

Matlab: immagine intera ad 8 bit → formato uint8



se ho una colonna mi interessa che l'ultimo sia a 8 bit di intero

imadd

imsubtract

immultiply

prendo due immagini dello stesso tipo,  
fanno le operazioni, calcolano e danno in  
uscita il formato d'entrata

[un radiologo /  
medico + assistente  
vengono rappresentati  
con + pixel di quanti  
ne hanno in realtà]

im2double: prende una qualsiasi immagine e la converte in double

↳ CASTING

im2uint8: per riportare l'immagine nel formato che ci interessa

pixel sono le limite inferiore della risoluzione spaziale

C'è anche la risoluzione data dal n° di bit → dinamica del pixel (8bit → dinamico di 256)

↳ RISOLUZIONE IN AMPIEZZA o DINAMICA DELL'IMMAGINE

Per l'occhio umano non fa differenza il contrasto <sup>di scala</sup> dell'occhio è il 2% → sottili di grigio

↳ la dinamica conta per l'algoritmo di processing.

Mostre immagini sono referate su monitor da referazione: da 23-25 pollici in su, con una scala

colore a 16 bit (ottimizzato per immagini mediche) → costano 10-20 volte in più

[un videoproiettore: 7 dpi (dot per inch) → in un pollice (2.5cm) ci sono 72 punti

Mostre immagini mediche hanno al loro interno il fattore di compressione → per misurare la risol. spaziale

FWHM: si crea un'immagine contenente un punto puntuale (o una colonna) invece di ottenere

un punto ottengo una macchia → si prende una riga di matrice e si rappresenta il

profilo di intensità dei numeri (si saltano i numeri)

la risoluzione spaziale è la larghezza a metà del massimo → se ho il fattore di compressio

lo posso riportare in millimetri

Risoluzione d'Impulso → dinamico

Risoluzione temporale = capacità di generare tante immagini nell'unità di tempo (Frame Rate)

↳ standard video: 25 fps

altri (es. x il cuore): 100 fps

## PARAMETRI

• Parametri del numero dell'immagine

• che caratterizzano la qualità dell'immagine

↳ concettualmente si varia la qualità con un REAMPONIAMENTO

### DENSITÀ OTTICA INTEGRATA (IOD)

è la somma dei pixel dell'immagine

Si usa sulle radiografie → densitometria ottica: quanto luce arriva sullo scudo: misura del peso dell'immagine

### LUMINOSITÀ

valore numerico corrispondente al pixel, aumentare o diminuire la luminosità = spostare tutti i pixel dello stesso valore (in alto o in basso)

La variazione di luminosità è un processo irreversibile se c'è saturazione (è reversibile se non c'è saturazione)

### CONTRASTO

è una misura relativa di differenza di colore tra l'oggetto e lo sfondo

$$C = \frac{I_o - I_s}{I_s}$$

immagine      sfondo

↳ contrasti bassi: oggetti poco riconoscibili sullo sfondo

↳ " alti: sfondi troppo sensibili

Come si varia il contrasto: si fissa una soglia (LUMINANZA → valore medio del segnale =  $\frac{IOD}{N \text{ pixel}}$ )

se il valore del pixel è superiore alla luminosità viene aumentato (in modo proporzionale alla distanza dalla luminosità); se è inferiore viene diminuito → in percentuale

Anche con il contrasto è possibile avere saturazione

È reversibile se non ho saturazione → ma non è immediato trovarlo (vario la luminosità con la percentuale)

### RAPPORTO SEGNALE-RUMORE

Non è così importante come nei segnali, si vede ad occhio

aumentando il contrasto, il rapporto segnale rumore aumenta

### VISIONE PROBABILISTICA DELL'IMMAGINE - ISTOGRAMMA DELLA LUMINOSITÀ

Immagine = distribuzione di pixel → variabile casuale (del valore del pixel)

densità di probabilità ⇒ istogramma della luminosità: conta il n° di volte che un numero compare (come classi tutti i numeri da 0 a N-1)

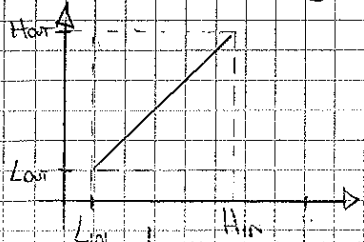
→ è la base fondamentale per operazioni di processing: è ciò che dice come vanno le cose

L'asse orizzontale può essere scelto in numero o in colore.

Parametri dell'istogramma: media, deviazione standard, mediana, n° pixel, intensità, ...

Ad ogni immagine corrisponde un solo istogramma della luminosità ma non viceversa → si perde

Es geometria dell'immagine.



↳ immagine in uscita uguale all'immagine in ingresso

MATLAB: `imadjust (immagine, [L_in], [L_out], gamma)`

variando gamma varia la curvatura → rendendo corrispondenza non lineare

parametri ingresso  
retta (gamma)  
parametri uscita  
scala che va da 0 a 1

COLORMAP (mappa colore) = vettore di numeri che il dispositivo di visualizzazione deve sapere

↳ corrispondenza tra numero e toni di grigio

con gamma ≠ 1 i toni di grigio/numero non sono più lineari

Se modifico la luminosità, l'istogramma trasla

Se modifico il contrasto, l'istogramma si stira (se aumento) rispetto alla luminosità (valore centrale)

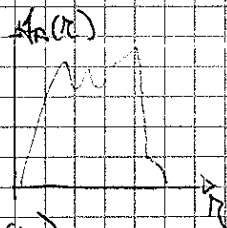
o si schiaccia (se diminuisco) ⇒ cambia forma; il valore medio probabilmente rimane uguale, la deviazione standard cambia

### EQUALIZZAZIONE DI UN ISTOGRAMMA

[si rendono egualmente rappresentative le varie frequenze dello banda → audio]

si vuole ottenere un istogramma della luminosità piatto → distribuzione uniforme.

$x$ : variabile casuale che misura il valore di un pixel → [0, 1]

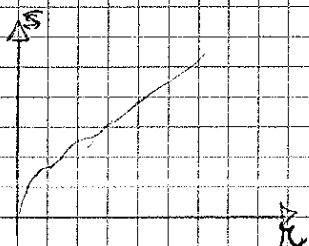


se disegno l'istogramma della luminosità = funzione densità di probabilità  $f_X(x)$

$$S = \int_0^x f_X(x') dx'$$

variabile casuale ⇒ funzione di densità cumulativa (contando l'area sottesa dalla curva)

↳ è una funzione monotona crescente compresa tra 0 e 1



Se uso la variabile  $S$  → ho un istogramma piatto

→ è proporzionale alla distribuzione progressiva dei colori

Colore e l'istogramma dell'immagine avendo come classi  $x$ ; colore  $S$  cumulato

Se rappresento l'immagine usando come colore map  $S$  → ho l'immagine equalizzata

•  $I \quad x \rightarrow S$

`imshow(I)`

`colormap(S)`

•  $x \rightarrow S$

$I \rightarrow J$  (sostituendo ad  $x$  il valore di  $S$ )

`imshow(J)`

L'istogramma non verrà proprio piatto, perché siamo nel discreto → approssimazione

MATLAB: `imhist` : per ottenere l'istogramma

`histeq` : per equalizzare l'istogramma

# FILTER

10-05-2012

È un po' più complicato aumentare il rapporto segnale-rumore  $\Rightarrow$  filtraggio digitale  
(sono + fastidi di quelli sui segnali)

$\Rightarrow$  OPERAZIONI di ENHANCEMENT

• OPERAZIONI PUNTUALI: trasformano l'immagine di potenza in un'altra formata da elementi i cui valori sono funzione degli elementi corrispondenti...

• OPERAZIONI LOCALI: ... Funzione di un determinato gruppo di elementi

$\hookrightarrow$  può capitare che filtri locali usino tutta l'immagine

Si usano soprattutto operazioni locali.

La variazione di luminosità è un'operazione puntuale; la variazione di contrasto è locale

Il filtraggio si fa per CONVOLUZIONE (nel dominio dello spazio)

$$g[m, n] = h[k, p] * a[m, n] = \sum_{k=-K}^K \sum_{p=-L}^L h[k, p] a[m-k, n-p]$$

$\hookrightarrow$  si chiama KERNEL (è una risposta di impulso)

normalmente ha dimensioni dispari (xk è così si ha il centro)

È un'operazione locale

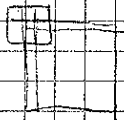
(si fa il prodotto del kernel centrato su un elemento per la matrice somma i valori che ha ottenuto  $\rightarrow$  valore filtrato dove è centrato il kernel)

MATRICE: matrix

Il kernel ha dimensioni più piccole della matrice  $10 \times 10$  o  $20 \times 20$  è un kernel grande

• ha una sua densità ottica (integrata (100))  $\rightarrow$  con la convoluzione viene spostata sull'immagine (per non perdere in modo significativo l'immagine  $\rightarrow 100=1$ )

• all'inizio e alla fine ci sono alcuni elementi che sono moltiplicati per niente



$\rightarrow$  zero padding: aggiungo una corona di zero intorno alla matrice questa operazione non è usata

$\rightarrow$  dato che la matrice è molto grande, si parte dal primo elemento per cui il kernel è tutto contenuto nella matrice

$\hookrightarrow$  il contorno non viene processato



è una soluzione accettabile dato che le dimensioni zero + grandi del kernel

$\rightarrow$  se l'oggetto è al centro e del contorno non ci importa, si fa una

convoluzione circolare: si fa finta di estendere l'immagine usando la parte sopra per il sotto e viceversa

strategia molto usata con le immagini mediche

Se il kernel è separabile può essere diviso nel prodotto di un vettore riga e un vettore colonna

$\rightarrow$  l'operazione di filtraggio è una doppia convoluzione lineare (sono indipendenti righe e colonne)  $\rightarrow$  l'ordine con cui le eseguo è indifferente  $\rightarrow$  commutativo

$\hookrightarrow$  è + rapida, + leggera in termini di memoria

SISTEMA LINEARE SPAZIO INVARIANTE

Esiste l'equivalente in frequenza (Fourier bidimensionale), ma non si usa: è complesso e lento

## FILTRO PASSABASSO (SMOOTHING) di AVERAGING

rendono l'immagine fissa, omogenea. Servono a fare denoising → rumore visivo

↳ sono usati per preparare l'immagine ad algoritmi successivi (l'occhio umano non percepisce molto il rumore, ma gli algoritmi sì) → rumore numerico



gli elementi del kernel vengono moltiplicati per quelli dell'immagine e si vuole avere come risultato l'elemento medio

↳ i valori del kernel sono  $\frac{1}{m^2}$  dove il kernel è  $m \times m$



↳ densità ottica integrata vale  $\leq 1$  → non si ottiene o aumento di luminosità

MATLAB: `fspecial('average', 3)`

si possono creare anche matrici del kernel rettangolare e non quadrate → l'effetto passabasso è diverso nelle due direzioni  
la dimensione del kernel dice la potenza del sfocaggio → più è grande, + omogeneo

↳ non devo esagerare, se no ho blurring (tende tutto a diventare un'orma) → è controproducente

## FILTRO GAUSSIANO

Come prima, ma invece di fare una media, fa una media pesata.

tre parametri: le due dimensioni e l'apertura  $\sigma$

↳ è un kernel separabile: se lo è la matrice è simmetrica

$$e^{-\frac{(k^2 + l^2)}{2\sigma^2}}$$

↳ più piccolo il filtro tende a riproporzionare

filtro di più in filtro grande, e uno con uno  $\sigma$  grande

`fspecial('gaussian', 4, 1)`

è più simile all'origine del filtro di averaging: se il rumore è gaussiano, il filtro gaussiano lo toglie meglio

## FILTRO MEDIANO

Nel kernel si mettono solo 0 o 1 → l'uscita è la mediana della distribuzione dell'intensità dei pixel. È una maschera: può essere quadrata, circolare, a rombo, ...

→ non è fatto per il rumore gaussiano: produce un'immagine mosaico

È fatto per il rumore SALE e PEPE (dei pixel o vengono spinti a bianco o forzati a nero)

↳ è il classico rumore tecnologico di acquisizione: pixel bruciati (nero o bianco o secondi di come è bruciato il transistor)

funziona molto bene, perché la mediana è a metà: i bianchi e i neri (se hanno la stessa 'concentrazione' non cambiano la mediana)

(rumore sale e pepe → può essere abbastanza presente in risonanza magnetica)

[su moderni schermi LCD → lo schermo nero consuma di più]