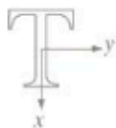







# TUTTE LE FORMULE INTERAZIONE E MULTIMEDIA

Formula per rappresentazione di immagini usando la luce incidente (i) e quella riflessa (r)

$$f(x, y) = i(x, y)r(x, y)$$

Formule delle trasformazioni affini

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Forward mapping (T è la matrice affine)

$$[xy1] = [vw1] * T$$

Inverse mapping (T è la matrice affine)

$$[vw1] = [xy1] * inversa(T)$$

Formula Interpolazione Bilineare

$$v(x, y) = ax + by + cxy + d$$

Formula Interpolazione Bicubica

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

Formula del MSE

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [I'(x, y) - I(x, y)]^2$$

Formula del PSNR

$$PSNR = 10 \log_{10} \left( \frac{S^2}{MSE} \right)$$

Nell'occhio i coni sono: 6/7 milioni

I bastoncelli 75/150 milioni

Raggio foro "reale" del pinhole

$$r = \sqrt{\lambda d}$$

Equazione lente sottile

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

Formula magnificazione

$$m = \frac{v}{u}$$

$$f = \frac{um}{(m + 1)}$$

Nyquist Rate è il doppio della frequenza massima nel segnale

Formula quantizzazione uniforme

$$L' = \frac{L * K}{N}$$

Formula quantizzazione NON uniforme

$$L' = \frac{f(L) * K}{f(N)}$$

oppure generalizzando:

$$L' = f(L, N, K)$$

Una quantizzazione non uniforme è quella logaritmica

$$L' = \frac{\log_2 L * K}{\log_2 N}$$

Formule per i valori del tristimolo

$$X = \int_{380}^{780} L_{e,\lambda} \bar{x}(\lambda) d\lambda$$

$$Y = \int_{380}^{780} L_{e,\lambda} \bar{y}(\lambda) d\lambda$$

$$Z = \int_{380}^{780} L_{e,\lambda} \bar{z}(\lambda) d\lambda$$

Formula per trovare la z nel diagramma cromatico CIE (2D)

$$z = 1 - (x + y)$$

Formule per passare dai valori del tristimolo XYZ a xyz

$$x = X/(X + Y + Z)$$

$$y = Y/(X + Y + Z)$$

$$z = Z/(X + Y + Z)$$

Formule per passare da CIE XYZ allo spazio colore CIELAB

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{1/3} - 16$$

$$a^* = 500 \left[ \left( \frac{X}{X_n} \right)^{1/3} - \left( \frac{Y}{Y_n} \right)^{1/3} \right]$$

$$b^* = 200 \left[ \left( \frac{Y}{Y_n} \right)^{1/3} - \left( \frac{Z}{Z_n} \right)^{1/3} \right]$$

$X_n$ ,  $Y_n$  e  $Z_n$  definiscono il punto bianco

Formula per la distanza Euclidea (tra 2 colori in CIELAB)

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$$

Formule per chroma e hue angle in  $L^*C^*h$  avendo il colore in CIELAB

$$C^* = \sqrt{(a^*)^2 + (b^*)^2}$$

$$h = \tan^{-1} \left( \frac{b^*}{a^*} \right)$$

I colori sicuri per il web si esprimono esclusivamente con questi numeri esadecimali  
33, 66, 99, CC, FF

Formule per passare da RGB a YUV

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.169R - 0.331G + 0.5B$$

$$V = +0.5R - 0.419G - 0.081B$$

Formule per passare da YUV a  $YC_bC_r$

$$Y = 0.299R + 0.587G + 0.114B$$

$$C_b = U + 128$$

$$C_r = V + 128$$

Per gli istogrammi

Per un immagine  $I[m, n]$  si ha  $H(k)$  = numero di pixel di valore  $k$

Formula normalizzazione dell'istogramma (problemi di range)

$$v_{\text{nuovo}} = 255 \cdot \frac{(v_{\text{vecchio}} - \min_{\text{osservato}})}{(\max_{\text{osservato}} - \min_{\text{osservato}})}$$

Algoritmo di equalizzazione

$$p_r(r_k) = \frac{n_k}{MN}, \quad k = 0, 1, 2, \dots, L-1$$

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j, \quad k = 0, 1, 2, \dots, L-1$$

- Sia data una immagine a 3 bit ( $L=8$ ) con  $64 \times 64$  pixel ( $MN=4096$ ) con la seguente distribuzione di intensità:

$r_k$	$n_k$	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

- Applicando la formula si ha:

$$s_0 = T(r_0) = 7 \sum_{j=0}^0 p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7 \sum_{j=0}^1 p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

$$s_2 = 4.55, s_3 = 5.67, s_4 = 6.23, s_5 = 6.65, s_6 = 6.86, s_7 = 7.00.$$

Arrotondando:

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 6$$

$$s_1 = 3.08 \rightarrow 3 \quad s_5 = 6.65 \rightarrow 7$$

$$s_2 = 4.55 \rightarrow 5 \quad s_6 = 6.86 \rightarrow 7$$

$$s_3 = 5.67 \rightarrow 6 \quad s_7 = 7.00 \rightarrow 7$$

Operazione generica su un immagine

$$g(x, y) = T[f(x, y)]$$

Operatori puntuali:

Negativo

$$255 - f(x, y)$$

Trasformazione logaritmica

$$g(x, y) = c \cdot \log(1 + f(x, y))$$

Trasformazione di potenza

$$g(x, y) = c \cdot (f(x, y))^\gamma$$

La binarizzazione sceglie un valore soglia: i pixel al di sotto diventano neri quelli al di sopra bianchi

**FINE ARGOMENTI PROVA IN ITINERE 1**

**INIZIO ARGOMENTI PROVA IN ITINERE 2**

Un operatore  $F : V \rightarrow W$  si dice lineare se per ogni coppia di vettori  $v_1, v_2$  in  $V$  e per ogni coppia di numeri reali  $a, b$  si ha che:

$$F(a * v_1 + b * v_2) = a * F(v_1) + b * F(v_2)$$

Per indicare l' operazione di convoluzione si usa la notazione:

$$g = f \circledast h$$

La convoluzione è commutativa:

$$f \circledast h = h \circledast f$$

La convoluzione è associativa:

$$(f \circledast h) \circledast h_1 = f \circledast (h \circledast h_1)$$

Applicare un kernel  $h$  di dimensioni  $s \times t$  ad un'immagine se gli indici del kernel sono disposti in modo da avere gli indici (0,0) al centro:

$$g_{m,n} = \sum_{i=-\lfloor s/2 \rfloor}^{\lfloor s/2 \rfloor - 1} \sum_{j=-\lfloor t/2 \rfloor}^{\lfloor t/2 \rfloor - 1} (h_{i,j} * f_{m+i,n+j})$$

	-1	0	1
-1	<b>a</b>	<b>b</b>	<b>c</b>
0	<b>d</b>	<b>e</b>	<b>f</b>
1	<b>g</b>	<b>h</b>	<b>i</b>

Applicare un kernel  $h$  di dimensioni  $s \times t$  ad un'immagine se gli indici del kernel sono disposti partendo da 1 fino ad arrivare a  $s$  e  $t$ :

$$g_{m,n} = \sum_{i=1, j=1}^{s,t} h_{i,j} * f_{m+(i-s+\lfloor s/2 \rfloor), n+(j-t+\lfloor t/2 \rfloor)}$$

	1	2	3
1	<b>a</b>	<b>b</b>	<b>c</b>
2	<b>d</b>	<b>e</b>	<b>f</b>
3	<b>g</b>	<b>h</b>	<b>i</b>

Applicare un filtro lineare e shift invariante ad una immagine è equivalente a calcolare la convoluzione del kernel del filtro con l'immagine.

Esempi di operatori locali:

- Mediano → È un filtro non lineare che fornisce in uscita il valore mediano dell'intorno del pixel
- Minimo e massimo → preso un intorno  $m \times n$  di pixel si sostituiscono tutti i valori di questi pixel con il valore minimo/massimo rilevato in quell'intorno
- N-box o di media → Sono definiti da kernel  $N \times N$  con ogni elemento pari a  $1/N^2$ , sfocano le immagini
- N-binomiale o filtri gaussiani → smussano meno vigorosamente degli N-box

Formula rumore gaussiano bianco:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

Per rimuovere i 2 tipi di rumori solitamente si usano i filtri di media e mediano, quelli mediani funzionano meglio

Kernel notevoli: lati orizzontali

$$Sobel_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad Prewitt_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Kernel notevoli: lati verticali

$$Sobel_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Prewitt_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Formula magnitudo:

$$magnitudo = \sqrt{sobel_x^2 + sobel_y^2}$$

Kernel notevole: laplaciano

$$Laplaciano = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

Formula Trasformata di Fourier (da immagine a frequenza)

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad \text{per } u = 0, \dots, M-1, v = 0, \dots, N-1$$

Formula Antirasformata di Fourier (da frequenza a immagine)

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad \text{per } x = 0, \dots, M-1, y = 0, \dots, N-1$$

Formula di Eulero

$$e^{ix} = \cos(x) + i * \sin(x) \text{ e } e^{-ix} = \cos(x) - i * \sin(x)$$

Formula spettro della trasformata

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$$

Formula angolo di fase

$$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$$

Formula potenza spettrale

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

Quando si visualizza a schermo lo spettro di Fourier, come immagine di intensità, non si riescono a visualizzare le parti meno luminose, quindi si usa una compressione di tipo logaritmico che permette di visualizzarle:

$$D(u, v) = \text{clog}(1 + F(u, v))$$

Proprietà di separabilità della DFT 2-D

$$F(u, v) = \frac{1}{M} \sum_{x=0}^{M-1} k(x, v) e^{\frac{-i2\pi ux}{M}}$$

Dove:

$$k(x, v) = \left[ \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{\frac{-i2\pi vy}{N}} \right]$$

Il principale vantaggio delle proprietà di separabilità è che la  $F(u, v)$  può essere ottenuta applicando in due passi successivi la trasformata 1-D

Proprietà valor medio:

Il valore della trasformata nell'origine, cioè nel punto  $(u, v) = (0, 0)$  è dato da:

$$F(0, 0) = \frac{1}{N \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad \bar{f}(x, y) = \frac{1}{N \times N} F(0, 0)$$

- **A sinistra:** Viene calcolato  $F(0, 0)$ . Solitamente nella teoria standard della DFT (Discrete Fourier Transform),  $F(0, 0)$  è la semplice somma di tutti i pixel. In questa specifica slide, però, sembra essere già normalizzato (diviso per  $N \times N$ ).
- **A destra:** Viene definita  $\bar{f}(x, y)$ , ovvero la **media** (indicata dalla barra sopra la f), in funzione di  $F(0, 0)$ .

Fast Fourier Transform (DFT ottimizzata)

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-i2\pi ux/N]$$



Teorema convoluzione:

La trasformata della convoluzione di due segnali nel dominio spaziale equivale al prodotto delle trasformate dei due segnali.

che significa anche:

La convoluzione di due segnali nel dominio spaziale equivale all'antitrasformata del prodotto delle trasformate dei due segnali.

Formula convoluzione nel *dominio spaziale*:

$$g(x, y) = f(x, y) \otimes h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$$

ad un immagine  $f$  viene applicato un kernel  $h$

l'equivalente operazione nel *dominio delle frequenze*, moltiplicazione tra trasformate di Fourier

$$G(u, v) = F(u, v) H(u, v)$$

moltiplicazione tra antitrasformate di Fourier

$$g(x, y) = F^{-1} F(u, v) H(u, v)$$

Filtri low-pass nel dominio della frequenza

Lowpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$	$H(u, v) = e^{-D^2(u, v)/2D_0^2}$

Filtri high-pass nel dominio della frequenza

Highpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$	$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$

Filtri band reject nel dominio della frequenza

Bandreject filters.  $W$  is the width of the band,  $D$  is the distance  $D(u, v)$  from the center of the filter,  $D_0$  is the cutoff frequency, and  $n$  is the order of the Butterworth filter. We show  $D$  instead of  $D(u, v)$  to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[ \frac{DW}{D^2 - D_0^2} \right]^{2n}}$	$H(u, v) = 1 - e^{-\left[ \frac{D^2 - D_0^2}{DW} \right]^2}$

Compressione:

Formula frequenza carattere  $a_i$ :

$$f_i = (\text{occorrenze } a_i) / N$$

Formula entropia:

$$E = - \sum_i f_i \log_2(f_i) \quad i \in S$$

Teorema di Shannon

I dati possono essere rappresentati senza perdere informazione (lossless) usando almeno un numero di bit pari a:  $N * E$

Dove  $N$  è il numero di caratteri mentre  $E$  è l'entropia.

Formula Trasformata Discreta del Coseno DCT

$$F(u, v) = \frac{2}{N} \left[ \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u)C(v)f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \right]$$

Formula antitrasformata

$$f(x, y) = \frac{2}{N} \left[ \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \right]$$

Dove:

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{per } u = 0 \\ 1 & \text{altrimenti} \end{cases}$$

$$C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{per } v = 0 \\ 1 & \text{altrimenti} \end{cases}$$

Formula quantizzazione (JPEG)

$$F_{\text{quantizzato}} = \text{round}(F/Q)$$

dove  $F$  è un numero e  $Q$  è un fattore di quantizzazione

Formula codice grey

$$g_i = a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2$$

$$g_{m-1} = a_{m-1}$$

dove  $\oplus$  denota l'operatore di XOR (or esclusivo)