



UniCT – DMI
Interazione e Multimedia
ANNO ACCADEMICO
2019/20

Autori:

Alessio Tudisco

Sommario

Lezione 1 – Fondamenti	5
Breve accenno storico	5
Accenno sull'half-toning	5
Le leggi della percezione visiva	5
Rappresentazione di un'immagine	5
Immagini Raster vs Immagini Vettoriali	6
Convenzioni nella rappresentazione delle immagini	6
Tipologie di immagini	6
Lezione 2 – Operazioni sulle matrici	6
Operazioni su Immagini e matrici	6
Forward mapping e Inverse mapping	7
L'interpolazione	8
Casi particolari nell'interpolazione	8
Il Mean Square Error (MSE)	9
Peak Signal to NoiseRatio (PSNR)	9
Lezione 3 – Bayer Pattern & Color Interpolation	10
Come viene acquisita un'immagine digitale	10
Tipologie di sensori	10
Sensori a matrice CCD	10
Color Filter Array (CFA) e Color Interpolation	10
Il Bayer Pattern	10
Tipologie di Color Interpolation	11
Lezione 4 – Risoluzione	12
Il Pixel e la risoluzione	12
Tipologie di risoluzione	12
Lezione 5 – Occhio	12
L'occhio umano	12
Il Pinhole e l'acquisizione delle immagini con l'occhio umano	13
Le lenti sottili	13
La magnificazione	15
Lezione 6 – Campionamento & Quantizzazione	15
Il campionamento	15
La Nyquist rate e il teorema di Shannon	16
La quantizzazione (Uniforme, non uniforme, logaritmica)	16
Lezione 7 – Colori	17
La luce e il colore	17
La percezione umana del colore (e teorema del tristimolo di Young)	17
La composizione dei colori	18

Lo spazio di colore CIE XYZ e Spazio di colore xyz.....	18
Definizione di Modello di colore (o spazio dei colori).....	19
Lo spazio di colore CIE L*a*b*	19
Lo spazio di colore CIE L*a*b*	20
Lo spazio di colore HSV (o HSI) e piccola definizione Munshell System (H, C, V)	21
La sintesi additiva del RGB	22
La sintesi sottrattiva del CMY.....	22
Lo standard YUV.....	23
Lo standard <i>YCbCr</i>	23
La palette o Look-Up-Table (LUT), l'indicizzazione di un'immagine true color e il re-indexing.....	23
Lezione 8/9 – Operazioni sulle immagini	23
L'istogramma	23
Il contrast stretching (espansione del contrasto).....	24
La normalizzazione.....	24
L'equalizzazione	24
Operazioni sulle immagini e tipologie.....	25
Operatori puntuali	25
Operatori lineari e invarianti per traslazione.....	26
Il Kernel	27
La convoluzione e il problema dei bordi	27
Operatori locali basici	27
Il rumore.....	27
L'estrazione dei bordi.....	28
Filtri di sharpening	29
Lezione 10 – Fourier.....	29
La serie, trasformata, dominio di Fourier	29
Immagini e Segnali	29
La serie di Fourier, trasformata e anti-trasformata di Fourier in 1-D e 2-D.....	30
Le componenti della trasformata di Fourier	30
Range dinamico.....	31
Vantaggi dell'uso della Trasformata di Fourier	31
Trasformazione diretta e trasformazione inversa	31
La Discrete Fourier Transform (DFT) in 1-D e 2-D	31
Le proprietà della DFT 2-D	32
La Fast Fourier Transform	32
Filtraggio nel dominio delle frequenze	32
Il teorema della convoluzione.....	33
Come ottenere un filtro a partire da una maschera parziale	33
Filtri di smoothing o passa-basso (low-pass)	33

Filtri di sharpening o passa-alto (high-pass)	34
Filtri di band-reject.....	35
Lezione 11 – Compressione & BitPlanes.....	36
Cosa si intende per comprimere?	36
I dati ridondanti	36
Definizione di algoritmo di compressione e classificazione dei metodi di compressione.....	36
Compressione lossless e criterio per buona compressione lossless.....	37
Compressione lossless: Huffman	37
Compressione lossless: Run-length	37
Compressione lossless: Differenziale.....	37
Compressione lossy e criterio per buona compressione lossy	38
Compressione lossy: Requantization	38
Compressione lossy: JPEG (Joint Photographic Experts Group)	38
I bit-plane	39
Codifica binario puro, caratteristiche e problematiche.....	39
Codifica Gray (Gray code) e confronto con il binario puro	40

Interazione e Multimedia – Appunti

Lezione 1 – Fondamenti

Breve accenno storico

L'uomo ha fatto uso del linguaggio delle immagini fin dall'albore della civiltà, ancora prima dell'avvento del linguaggio scritto, in quale è utilizzato dall'uomo solo da pochi millenni.



La prima fotografia

La *prima fotografia non digitale* (1) risale al 1827: ritrae il punto di vista di una finestra e per effettuarla la lastra eliografica fu esposta per ben 8 ore.

La *prima applicazione di una fotografia digitale* risale al 1920 con l'invio tramite cavo di una fotografia da New York a Londra per una testata giornalistica. L'invio avvenne mediante un protocollo specifico per l'immagine e il risultato venne stampato in *half-toning*.

La *prima immagine elaborata da un calcolatore* risale al 1964, quando un computer della NASA ricevette un'immagine della Luna e la elaborò per correggere alcune distorsioni ottiche.

Accenno sull'half-toning

La stampa *half-toning* è un particolare metodo di stampa che permette la stampa di un'immagine chiaroscuro quando si ha a disposizione un solo colore, ad esempio il nero per i quotidiani. Questa tecnica si basa sulla ricostruzione dell'immagine sotto forma di puntini equidistanti e di dimensione variabile a seconda del livello di grigio che si vuole ottenere.

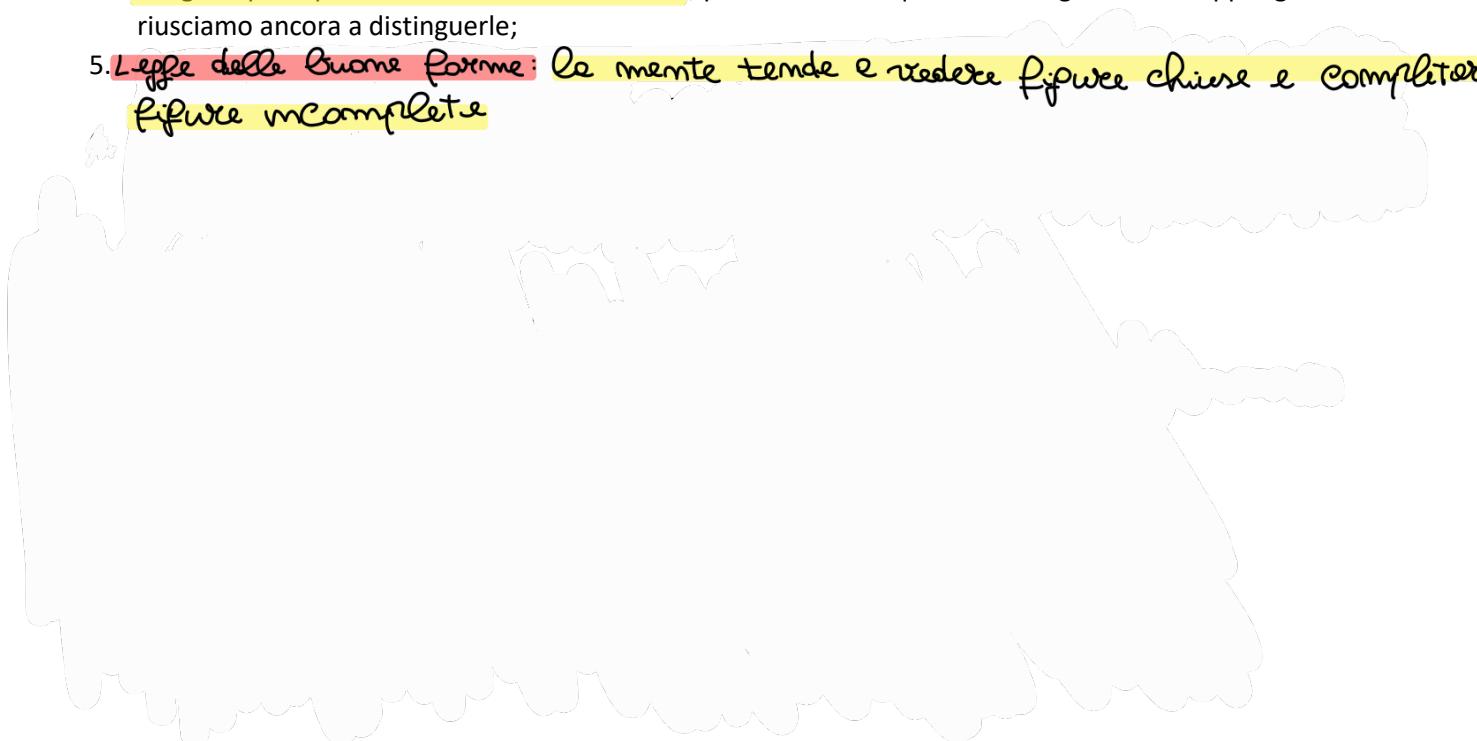


Half-toning

Le leggi della percezione visiva

Nell'uomo possiamo dire che l'occhio "vede" ma è il cervello a "elaborare" le immagini. Ciò implica che *il nostro modo di reagire agli stimoli visivi e formare immagini è soggettivo*, un esempio possono essere i daltonici. Seguono delle leggi che definiscono *alcuni comportamenti comuni nella nostra percezione visiva*:

1. **Legge della vicinanza:** le parti più vicine di un insieme che entra nel nostro campo visivo tendono ad essere percepite come raggruppamenti di unità;
2. **Legge della chiusura:** le figure chiuse, come ad esempio cerchi o quadrati, tendono ad essere percepite più facilmente di quelle aperte, prevalgono poiché si imprimono più facilmente nella visione;
3. **Legge dell'uguaglianza:** gli elementi uguali o simili tendono a essere percepiti insieme, ad esempio una serie di punti disposti orizzontalmente uno accanto all'altro vengono percepiti come una linea;
4. **Legge della continuità:** un insieme di punti posti uno di seguito all'altro in una determinata direzione vengono percepiti come una struttura unitaria, per tale motivo quando due figure di sovrappongo riusciamo ancora a distinguere;
5. **Legge delle Buone Forme:** La mente tende a vedere figure chiuse e complete, a riempire le forme incomplete.



2.2 Bande di Mach

A sostegno della tesi per cui la percezione visiva è oggettiva e differente dall'intensità luminosa effettiva, portiamo come esempio le bande di Mach: sono 5 bande grigie di 5 intensità diverse. Apparentemente l'intensità luminosa non è uniforme, e tende a cambiare in prossimità dei bordi delle bande, cosa che tuttavia, non avviene effettivamente. Questo dimostra quindi che l'intensità luminosa viene percepita in maniera differente in ambienti scuri e in ambienti chiari.

È stato inoltre scientificamente dimostrato che l'intensità luminosa percepita cresce in maniera logaritmica rispetto a quella effettiva.

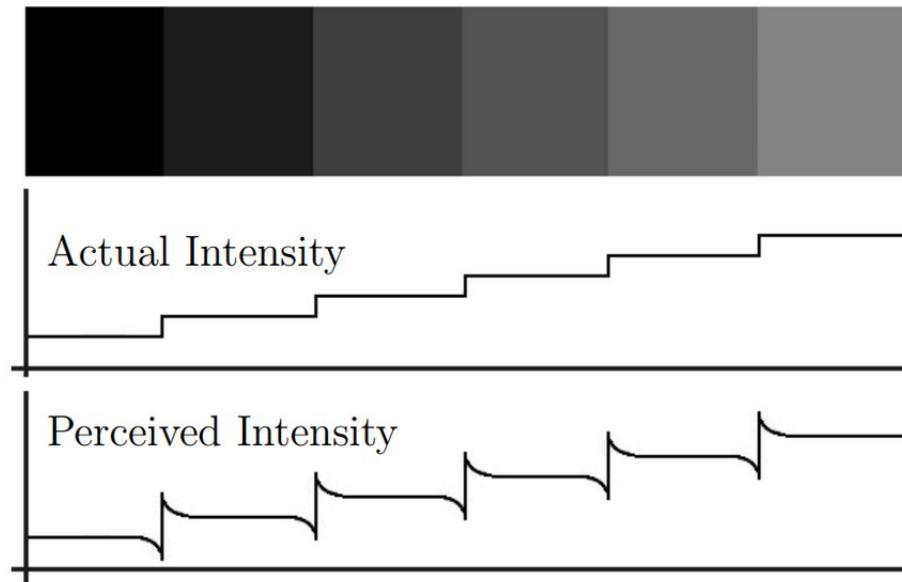
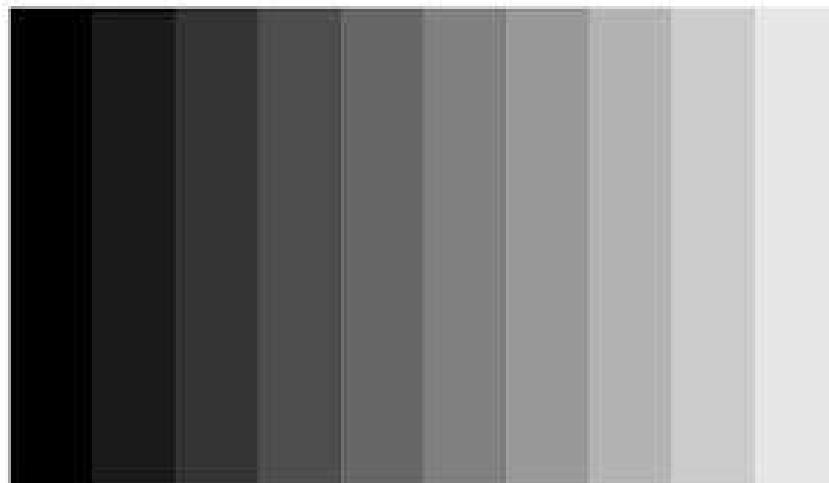


Figura 2.2: Bande di Mach: intensità effettiva vs percepita



Rappresentazione di un'immagine

Un'immagine è rappresentabile da una funzione bidimensionale $f(x, y) = L$ in cui:

- Le variabili x e y rappresentano le *coordinate spaziali* e il piano su cui esse risiedono è detto *dominio spaziale*;
- Il valore L è proporzionale alla luce incidente nell'oggetto e anche a quella riflessa

È quindi possibile ridefinire il risultato della funzione come $f(x, y) = i(x, y) * r(x, y)$, dove:

- Le variabili i e r rappresentano rispettivamente la luce incidente e quella riflessa;
 - Il valore della luce incidente i può assumere valori: $0 < i(x, y) < \infty$ (*limiti teorici*)

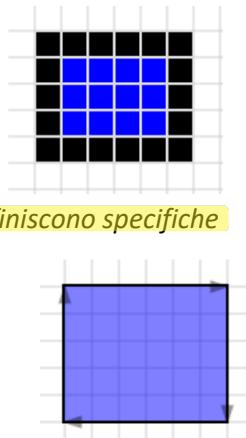
- Il valore della luce riflessa r può assumere valori: $0 < r(x, y) < 1$ (limiti teorici)

In teoria il valore L è un numero reale, ma per produrre un'immagine digitale abbiamo bisogno di valori discreti, in quanto in un calcolatore non abbiamo infiniti bit per rappresentare i numeri: il passaggio dal continuo al discreto è attuato mediante le operazioni di *campionamento e quantizzazione*.

Immagini Raster vs Immagini Vettoriali

Distinguiamo principalmente due tipologie di immagini: *immagini raster* e *immagini vettoriali*:

- Le *immagini raster* sono caratterizzate da una serie finita di pixel (il valore quantizzato da un sensore) rappresentati in una matrice. Il principale vantaggio di questo tipo di immagini è il fotorealismo, le fotocamere producono infatti immagini raster. Gli svantaggi sono invece le grandi dimensioni e la mancanza di una descrizione semantica;
- Le *immagini vettoriali* sono caratterizzate da una serie di regole matematiche che definiscono specifiche forme geometriche. Un visualizzatore di immagini vettoriali segue tali regole matematiche per mostrare l'immagine in un specifico piano di riferimento. Il principale vantaggio delle immagini vettoriali è dovuto al fatto che le trasformazioni sul piano (traslazione, rotazione, zooming, scala) risultano abbastanza semplici. I principali svantaggi delle immagini vettoriali sono: lo scarso fotorealismo, in quanto non è possibile rappresentare ogni cosa sotto forma di figure geometriche, il problema dei formati vettoriali proprietari e la necessità di una particolare stampante della plotter.



Convenzioni nella rappresentazione delle immagini

Raster
Le immagini vengono rappresentate con una matrice $M \times N$, dove M e N rappresentano la lunghezza e l'altezza dell'immagine. Possiamo porre la matrice sul quarto piano cartesiano con le seguenti caratteristiche:

- Il punto di origine è posto in alto a sinistra;
- L'ascissa va da sinistra a destra a numeri positivi;
- L'ordinata va dall'alto verso il basso a numeri positivi;
- L'elemento $x[n_1, n_2]$ è corrisponde all'elemento presente nella riga n_1 e colonna n_2 ;

	0	1	2	...	n_2	...	N_2
0							
1							
2							
⋮							
n_1							
⋮							
N_1							

$x[n_1, n_2]$

Tipologie di immagini

- Immagini in bianco e nero:** ogni pixel è rappresentato da un solo bit e può assumere come valori 0 o 1; **pixel**
- Immagini in scala di grigio:** ogni pixel è generalmente rappresentato da 8 bit e può assumere come valori un numero fra 0 e 255; **con valori da 0 a 255 per RGB**
- Immagini a colori:** generalmente vi sono 8 bit per canale di colore. Per RGB vi saranno un totale di 24bit e il pixel è rappresentato da una terna (R, G, B) con R, G, B aventi valore compreso fra 0 e 255;

Lezione 2 – Operazioni sulle matrici

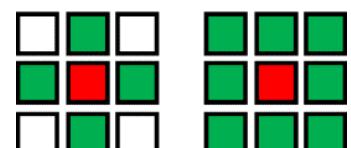
Operazioni su Immagini e matrici

Come già detto in precedenza, le *immagini raster* sono rappresentabili in una matrice, ne segue che è possibile eseguire su di esse le stesse operazioni che si possono fare con le matrici numeriche. Non è detto che tali operazioni abbiano un senso logico se applicate alle immagini, che senso avrebbe moltiplicare due immagini?

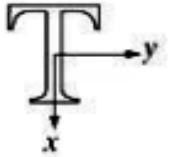
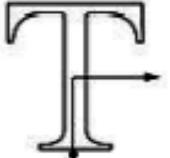
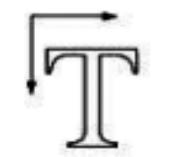
Segue una lista di operazioni su matrici che si eseguono sulle immagini:

- Prodotto fra matrici:** oltre all'operazione del prodotto riga per colonna delle matrici numeri, nell'image processing si esegue una moltiplicazione puntuale fra matrici;
- Neighborhood:** tale operazione individua i "vicini" di un pixel. Tali vicini di pixel possono essere i 4 pixel posti accanto a esso nelle quattro direzioni o gli 8 pixel che lo circondano;

$$\begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix} \cdot \begin{bmatrix} -3 & 0 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ 3 & -4 \end{bmatrix}$$



- **Trasformazioni affini:** tali operazioni mantengono inalterata l'intensità dei colori dei pixel ma ne modificano la geometria (posizione):

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity non fa nulla	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling ingrandisce	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation ruota	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation sposta un oggetto	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical) stretche in verticale	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal) stretche in orizzontale	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Applichiamo le trasformazioni offuscando

Forward mapping e Inverse mapping

Il *forward mapping* e l'*inverse mapping* sono funzioni che "mappano" la posizione del pixel da uno spazio di origine a uno spazio di destinazione. Un esempio d'uso può essere l'applicazione di una trasformazione affine.

Nello specifico il *forward mapping* scansiona tutti i pixel dell'immagine di input e ne calcola la posizione nell'immagine di output:

$$[X_0 \ Y_0 \ 1] = [x \ y \ 1] * T \text{ dove } T \text{ è la matrice affine}$$

Tale funzione ha principalmente due problematiche:

1. **Buchi neri:** l'immagine di output potrebbe presentare pixel senza intensità, e quindi buchi neri, nonostante la funzione abbia mappato tutti i pixel dell'immagine di input potrebbero esserci pixel dell'immagine di output che non sono stati individuati dalle formule;
2. **Sovraposizioni:** l'immagine di output potrebbe presentare del rumore, la funzione potrebbe aver mappato uno specifico pixel dell'immagine di output con più di un pixel dell'immagine di input;

Per evitare questi problemi si usa

Nello specifico l'*inverse mapping* scansiona tutti i pixel dell'immagine di output e ne calcola la corrispondenza con i pixel dell'immagine di input:

$$[x \ y \ 1] = [X_0 \ Y_0 \ 1] * \text{inversa}(T) \text{ dove } T \text{ è la matrice affine}$$

Capitolo 5

L'interpolazione

È un'operazione matematica che permette di costruire una funzione, detta **funzione interpolante** $V(x, y)$, che permette di calcolare i valori non noti all'interno di uno o più intervalli cui estremi sono noti, ottenendo un vero e proprio modo di ricostruire dei pixel non noti.

Queste tecniche vengono utilizzate in diversi modi, ma diventano particolarmente utile

5.1 Zooming in

Lo scaling, nel caso specifico dello **zooming in** (in forward mapping), genera un'immagine di dimensione maggiore da quella di partenza, con dei vuoti tra i pixel dell'immagine di partenza.

I seguenti algoritmi permettono di stimare in maniera più o meno adatta l'intensità dei nuovi pixel intermedi.

5.1.1 Nearest neighbour

Sceglie il valore di intensità del pixel noto più vicino. A parità di vicinanza, segue una regola arbitraria. Algoritmo più semplice, scarsi risultati.

Esempio:

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \Rightarrow \begin{bmatrix} 10 & ? & 20 & ? \\ ? & ? & ? & ? \\ 30 & ? & 40 & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Risultato:

$$\begin{bmatrix} 10 & 10 & 20 & 20 \\ 10 & 10 & 20 & 20 \\ 30 & 30 & 40 & 40 \\ 30 & 30 & 40 & 40 \end{bmatrix}$$

5.1.2 Interpolazione bilineare

Si prendono come riferimento i 4 pixel adiacenti (o più vicini) noti, per stimare l'intensità del pixel di intensità non nota. Il valore assegnato si ottiene con l'equazione $V(x, y) = ax + by + cxy + d$

Esempio:

$$\begin{array}{c} 0 \quad 1 \quad 2 \\ \bullet \begin{bmatrix} 10 & ? & 20 \\ ? & ? & ? \\ 30 & ? & 40 \end{bmatrix} \\ 1 \\ 2 \end{array}$$

$$\left\{ \begin{array}{l} V(0,0) : 10 = 0 + 0 + 0 + d \\ V(2,0) : 20 = 2a + 0 + 0 + d \\ V(0,2) : 30 = 0 + 2b + 0 + d \\ V(2,2) : 40 = 2a + 2b + 0 + d \end{array} \right. ;$$

$$\left\{ \begin{array}{l} V(0,0) : d = 10 \\ V(2,0) : 2a + d = 20 \\ V(0,2) : 2b + d = 30 \\ V(2,2) : 2a + 2b + d = 40 \end{array} \right. ;$$

$$\begin{cases} V(0,0) : d = 10 \\ V(2,0) : a = 5 \\ V(0,2) : b = 10 \\ V(2,2) : c = 0 \end{cases};$$

Otteniamo la funzione:

$$V(x,y) = 5x + 10y + 0xy + 10$$

Calcolo il pixel di coordinate (0, 1):

$$V(0,1) = 10 + 10 = 20$$

Calcolo il pixel di coordinate (1, 0):

$$V(1,0) = 5 + 10 = 15$$

Calcolo il pixel di coordinate (1, 1):

$$V(1,1) = 5 + 10 + 10 = 25$$

Calcolo il pixel di coordinate (2, 1):

$$V(2,1) = 10 + 10 + 10 = 30$$

Calcolo il pixel di coordinate (1, 2):

$$V(1,1) = 5 + 20 + 10 = 35$$

Immagine finale:

$$\begin{bmatrix} 10 & 15 & 20 \\ 20 & 25 & 30 \\ 30 & 35 & 40 \end{bmatrix}$$

5.1.3 Interpolazione bicubica

Utilizza i sedici pixel più vicini al punto.

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

Non la approfondiamo, ma sappiamo che produce una formula di interpolazione molto accurata, grazie ai numerosi valori presi in considerazione per le stime. Più pesante computazionalmente.

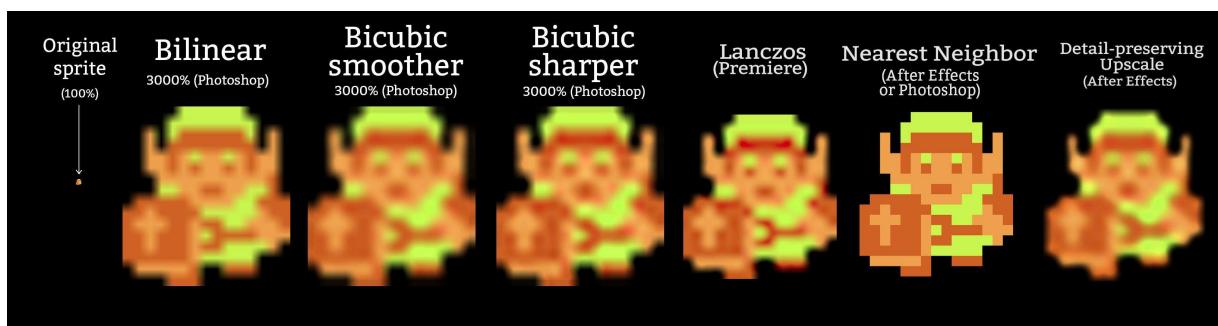


Figura 5.1: Il nearest neighbour è il più pertinente per sprite a bassa risoluzione, ma non funziona bene in immagini fotorealistiche

5.1.4 Problema dei bordi

A ridosso dei bordi dell'immagine, il numero di pixel adiacenti è minore, e non risulterà possibile usare in maniera precisa gli algoritmi di interpolazione. Due sono le possibili soluzioni:

- Non fare nulla, replicando i valori di righe e colonne adiacenti.
- Usare un algoritmo che prende come riferimento un numero minore di pixel adiacenti.

5.2 Zooming out

Esistono anche dei metodi da usare nel caso dello zooming out, che genera immagini di dimensioni minori. Esempio di soluzioni sulla seguente matrice:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

• 5.2.1 Decimazione

Teniamo un pixel seguendo una regola arbitraria (quello in alto a sinistra ogni 4 pixel, in questo caso):

$$\begin{bmatrix} 1 & 3 \\ 9 & 11 \end{bmatrix}$$

• 5.2.2 Media

In questo caso ogni due pixel in orizzontale e due in verticale, arrotondando per difetto:

$$\begin{bmatrix} 3 & 5 \\ 11 & 13 \end{bmatrix}$$

Procedure di calcolo della media hanno però un difetto, come vedremo spesso nel corso: possono generare nuovi colori inesistenti nell'immagine di partenza.

Il Mean Square Error (MSE)

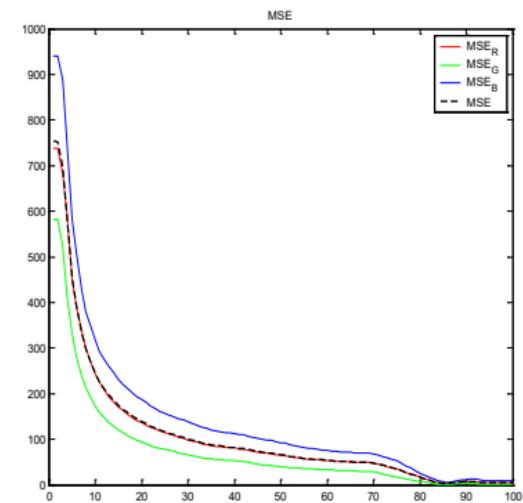
Il Mean Square Error (MSE) è un parametro usato per stimare l'errore quadratico medio fra due immagini, più il parametro MSE è basso minore è la differenza fra due immagini.

La formula per il calcolo dell'MSE è a seguente:

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

Si nota quindi che:

- L'MSE è il rapporto fra la somma dei quadrati delle differenti punto per punto per il numero totale dei pixel;
- Più è basso l'MSE più le due immagini si somigliano;
- Il valore minimo è 0, ovvero quando le due immagini sono uguali;
- Il valore massimo è pari S^2 (con S si indica il valore massimo che un pixel può assumere, es. 255), quando si effettua il controllo fra un'immagine e il suo negativo;
- L'MSE è molto sensibile allo spostamento dei pixel, la traslazione di anche di una sola posizione può sfalsare il risultato;
- Per calcolare l'MSE delle immagini RGB è possibile:
 - Calcolare la media fra gli MSE dei tre canali;
 - Utilizzare una combinazione lineare che dia maggior rilevanza al verde;



Peak Signal to Noise Ratio (PSNR)

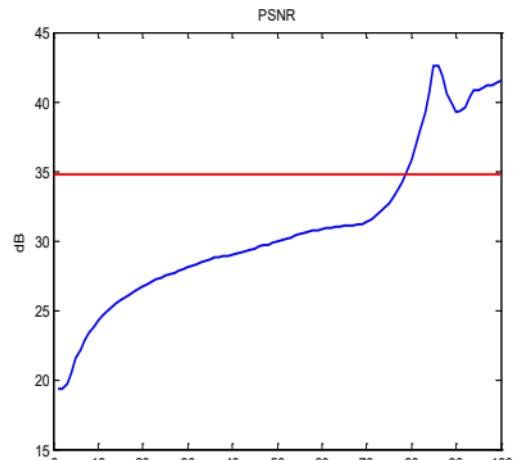
Il Peak Signal to Noise Ratio (PSNR) è un parametro usato per misurare la qualità di un algoritmo di compressione attraverso il confronto della qualità di un'immagine compressa rispetto all'originale, possiamo dire quindi che dipende dalla differenza fra l'immagine codifica e quella originale. Il PSNR non è il metodo migliore per valutare la qualità di un algoritmo di compressione ma è il più diffuso.

Le principali formule per il calcolo del PSNR è:

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

Si nota quindi che:

- Il valore S è il massimo valore che un pixel può assumere, per esempio 255 per immagini a 8 bit;
- Più è alto il PSNR più le due immagini si somigliano;
- Il valore massimo tende all'infinito, ovvero quando l'immagine è uguale all'originale;
- Un algoritmo si definisce "accettabile" se il PSNR è superiore a 35;
- Analogamente al MSE per calcolare il PSNR delle immagini RGB è possibile:
 - Calcolare la media fra i PSNR dei tre canali;
 - Utilizzare una combinazione lineare che dia maggior rilevanza al verde;

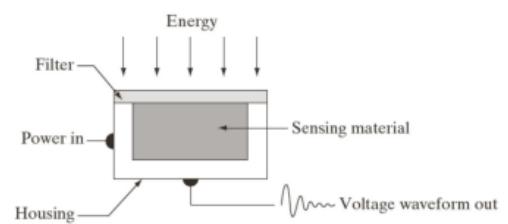


Lezione 3 – Bayer Pattern & Color Interpolation

Come viene acquisita un'immagine digitale

Quando la luce colpisce un oggetto, una parte di essa viene assorbita ed una parte viene riflessa, quest'ultima è quella che dà origine al colore percepito. Per acquisire un'immagine digitale è necessario che il sensore catturi ed elabori la luce riflessa.

Per tale motivo un sensore è costruito con un materiale particolarmente sensibile alla luce con il quale converte quest'ultima in un impulso elettrico che verrà successivamente digitalizzato.



Tipologie di sensori

Al giorno d'oggi si utilizzano le seguenti tipologie di sensori:

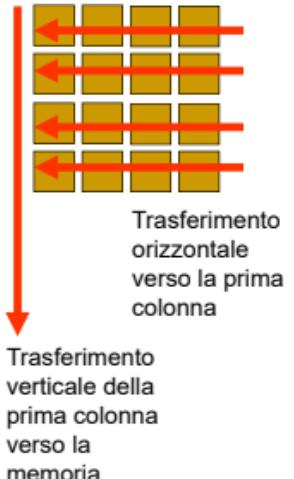
- **Sensore singolo:** il sensore si sposta fisicamente su tutta l'area da digitalizzare, un esempio è lo scanner;
- **Sensori in linea:** i sensori sono disposti in linea e si muovono in modo lineare, i sensori si muovono per scannerizzare l'oggetto, un esempio è la TAC;
- **Sensori a matrice:** i sensori sono disposti a matrice e non è necessario spostarli come nelle tipologie precedenti, questa tipologia è molto diffusa nelle fotocamere digitali;

Sensori a matrice CCD

I sensori a matrice più diffusi sono di tipo **CCD (Charged Coupled Device)**:

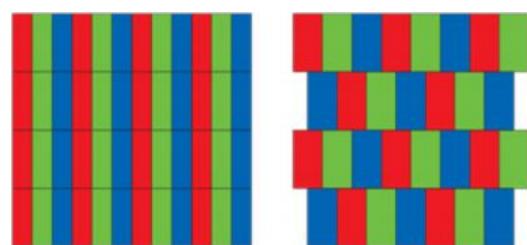
- Il dispositivo è formato da numerose **celle che se colpite dai fotoni assumono carica positiva**;
- Le **celle non possono caricarsi oltre un certo limite**, superato tale limite si ha il fenomeno della **sovra-esposizione**;
- Il **numero di celle per area** di esposizione costituisce un parametro di qualità della fotocamera e misurato in **MEGAPIXEL**;

Una volta acquisite le cariche in una matrice di celle esse devono essere trasferite in una memoria digitale. Il trasferimento avviene in C fasi, **una fase per ciascuna colonna della matrice**. Ad ogni fase viene trasferita in memoria la prima colonna della matrice, nello stesso tempo tutti gli elementi (dalla seconda colonna in poi) vengono *shiftati* di una colonna a sinistra.



Color Filter Array (CFA) e Color Interpolation

Ogni cella può memorizzare solo un colore per volta e non una terna, è quindi necessario scegliere un **modello di memorizzazione ottimale**, detto **color filter array (CFA)**. Il CFA viene posizionato davanti alle celle del sensore e ha il compito di filtrare la luce affinché passi solo un determinato colore, è quindi possibile decidere quale colore ogni cella possa memorizzare.



I due colori mancanti per completare la terna si ottengono mediante un processo di interpolazione dai pixel vicini, detto **color interpolation**. Infine, il grado di accuratezza dell'immagine finale è fortemente dipendente da quanto sia sofisticato il metodo di interpolazione.

Il Bayer Pattern

Il **Bayer Pattern** è attualmente il **migliore CFA conosciuto** ed è utilizzato dal **1980** in tutti i dispositivi elettronici.

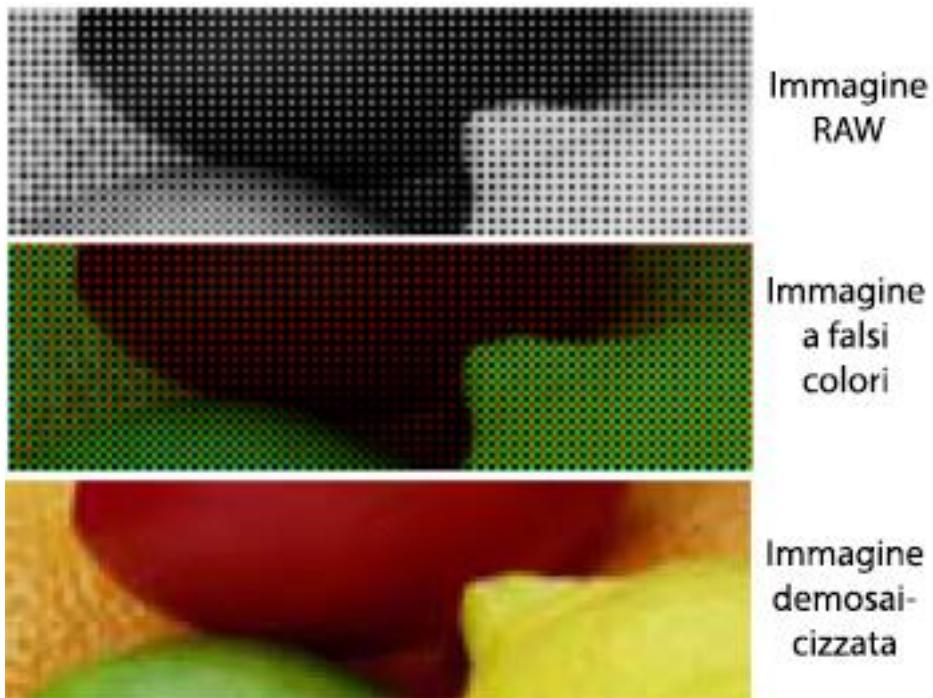
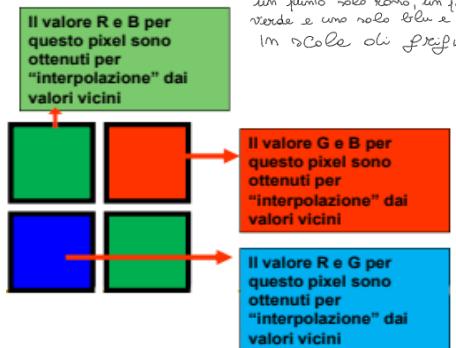
Questo CFA è costituito da una **matrice 2x2**, replicata più volte su tutto il sensore, che presenta **un rapporto 1:2:1**.

per R:G:B, privilegia le misure nel canale verde perché risulta essere quello più importante per la percezione umana.

Nella matrice 2x2, i pixel verdi sono posizionati nelle diagonali, inoltre dato che per ogni pixel viene salvata solo una componente di colore è necessario usare la color interpolation per ottenere le altre.

Le immagini in Bayer Pattern vengono salvate in un formato "raw", ovvero grezzo poiché contengono semplicemente i dati ottenuti dal sensore, privi di alcuna elaborazione.

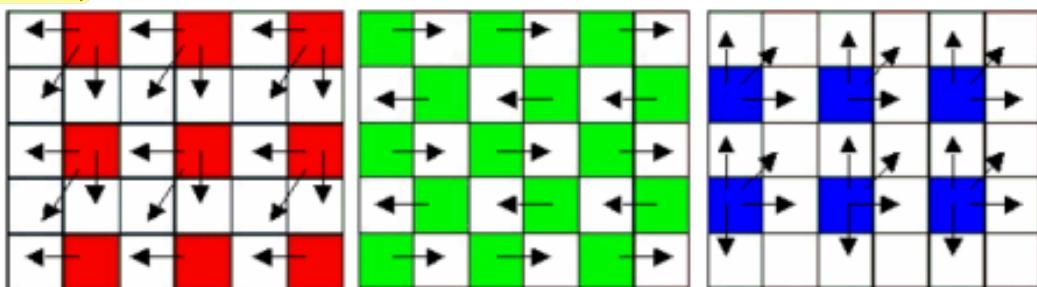
Un'immagine in formato raw è visualizzata in scala di grigi, inoltre dato che per ogni pixel si è salvato solo una componente della terna l'immagine sarà caratterizzata da un effetto mosaico, ovvero l'immagine sembra essere composta da un mosaico. Sovrapponendo il Bayer Pattern all'immagine raw si ottiene la cosiddetta "immagine a falsi colori". Infine, eseguendo la color interpolation si ottiene un'immagine a colori priva dell'effetto mosaico detta "immagine demosaicizzata", per tale motivo il processo di color interpolation viene anche detto "demosaicking".



Tipologie di Color Interpolation

La color interpolation è del tutto simile all'interpolazione trattata in precedenza, si ha:

- **Color Interpolation Replication:** Per ogni singolo pixel gli elementi mancanti della terna vengono copiati dall'intorno;



- **Color Interpolation Bilinear:** si utilizzano i 4 pixel più vicini per stimare i valori della terna delle componenti, la formula è uguale all'interpolazione bilinear trattata precedentemente e analogamente si ottiene un sistema a 4 incognite;
- **Color Interpolation Bicubic:** si utilizzano i 16 pixel più vicini per stimare i valori della terna delle componenti, la formula è uguale all'interpolazione bicubic trattata precedentemente, analogamente si ottiene un sistema a 16 incognite e anch'essa produce risultati migliori rispetto alla bilinear;

usiamo le stesse formule dell'interpolation normale

è corretto dato che questo
è l'asse dell'alternanza spaziale
dei tre colori perché vedo
un punto nero rosso, un punto nero
verde e uno nero blu e così via
in base alle frequenze

Lezione 4 – Risoluzione

Il Pixel e la risoluzione

Come già detto in precedenza, un **PIXEL** è il valore quantizzato da un sensore. Con il termine **risoluzione** si indica il **numero di pixel per unità di misura**, che possono essere:

- **Pixel al centimetro** (**px/cm**);
- **Dots per inch** (**dpi**) [**Punti per pollice**];
- **Megapixel**, sotto forma del numero totale di pixel in un'immagine;

Nelle immagini la **risoluzione** indica il **grado di qualità**, più è alta la risoluzione più saranno i pixel in essa contenuti e di conseguenza si avrà un maggior dettaglio.

Tipologie di risoluzione

Il concetto di **risoluzione** è **relativo** e va considerato a seconda della situazione, **si ha infatti:**

- **La risoluzione dell'apparecchiatura di ripresa:** la quale indica la risoluzione del dispositivo di cattura dell'immagine, per esempio un generico sensore di una fotocamera a 48 megapixel;
- **La risoluzione dell'apparecchiatura di resa:** la quale indica la risoluzione del dispositivo di visualizzazione dell'immagine, per esempio un generico monitor con i suoi 72 dpi;

Per ottenere la **massima resa da un'immagine** è necessario che le due risoluzioni combacino.

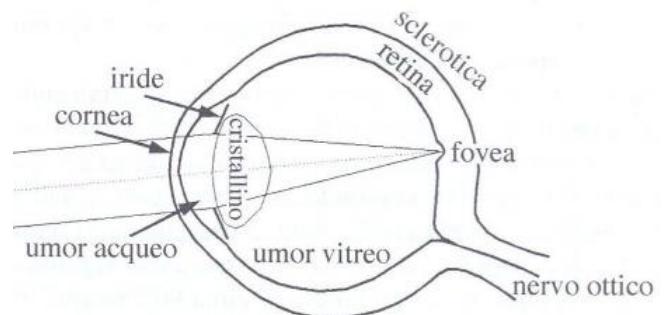
Lezione 5 – Occhio

L'occhio umano

Come abbiamo già detto, l'**essere umano vede le immagini attraverso l'occhio**. Per lo studio della percezione visiva umana è, quindi, molto importante capire com'è formato l'**occhio umano**.

Una parte fondamentale dell'occhio è la **retina**, una membrana che ricopre la parte posteriore dell'occhio. Questa membrana è costituita da due tipi di **fotorecettori**:

- **I coni:** sono circa 6/7 milioni ~~di sensori~~ e sono localizzati in una zona centrale della retina detta **fovea**, sono inoltre **fortemente sensibili al colore** e ognuno di essi è collegato ad un nervo ottico, ciò li **rende responsabili della vista policroma**;
- **I bastoncelli:** sono circa 75/150 milioni ~~di sensori~~ e sono sparsi per tutta la retina, sono **poco sensibili al colore** e sono collegati a gruppi ad un nervo ottico, ciò li **rende responsabili della vista monocroma**;



La **fovea** è una regione di circa $1,5 \text{ mm} \times 1,5 \text{ mm}$ e ha una popolazione di coni pari a 150.000 ~~sensori~~ per mm^2 , ciò denota che vi sono circa 337.000 **coni** nella fovea. Tale numero è impressionante poiché un CCD per contenere lo stesso numero di celle necessita di una superficie di almeno $5 \text{ mm} \times 5 \text{ mm}$.

9.2 Pinhole, cristallino e messa a fuoco

Il modello Pinhole è un modello che permette di astrarre il problema della formazione dell'immagine dell'occhio:

Si approssima l'occhio a una scatola scura, foriamo con uno spillo (pin hole) una parete della scatola e poniamo, nella parete interna opposta al foro, una pellicola fotosensibile (che simula la retina).

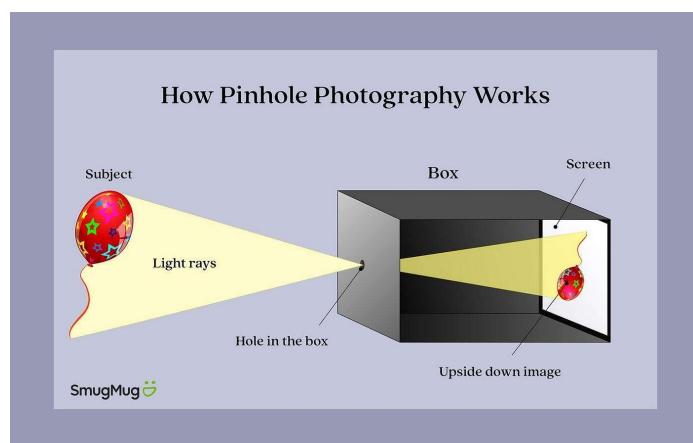


Figura 9.1: Modello Pinhole

foro della pupilla, non essendo veramente puntiforme, permette l'accesso a più proiezioni di luce dallo stesso punto: la proiezione sulla retina di un punto dovrebbe di conseguenza diventare quella di un cerchio. Tuttavia, nell'occhio, grazie alla deviazione della luce per mezzo del cristallino, la proiezione viene corretta. Il cristallino, essendo flessibile, si adatta alle distanze che vogliamo mettere a fuoco.

9.3 Sulle fotocamere

Le lenti delle fotocamere non sono deformabili: per questo motivo, tramite un sistema manuale o automatico, è possibile cambiare la distanza della superficie fotosensibile rispetto al foro, per permettere la messa a fuoco di soggetti a varie distanze. Nel modello pinhole, il raggio del foro è proporzionale alla radice quadrata della distanza per la lunghezza d'onda della luce emessa.

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

Tuttavia, per quanto un modello ideale del pinhole risulterebbe funzionante, un modello reale soffre di molti problemi:

- Il pinhole reale non presenta un foro puntiforme infinitesimale, in quanto non permetterebbe il passaggio della luce.
- Il pinhole reale però non può avere nemmeno un foro di dimensioni troppo piccole, in quanto genererebbe fenomeni di diffrazione.
- Un pinhole reale e funzionante presenta un foro di dimensioni notevoli ed una specifica forma geometrica. Per quanto questa "implementazione" sia funzionante, è soggetta a fenomeni di sovrapposizione di più fonti di luce.

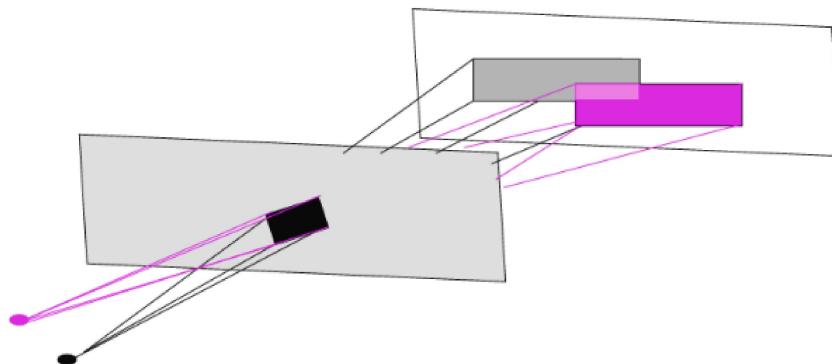


Figura 9.2: Sovrapposizione di due fonti luminose puntiformi

9.4 Lenti sottili

I fori sono fondamentalmente inadeguati a replicare realmente il funzionamento dell'occhio. Il modello pin-hole rimarrà una semplice astrazione.

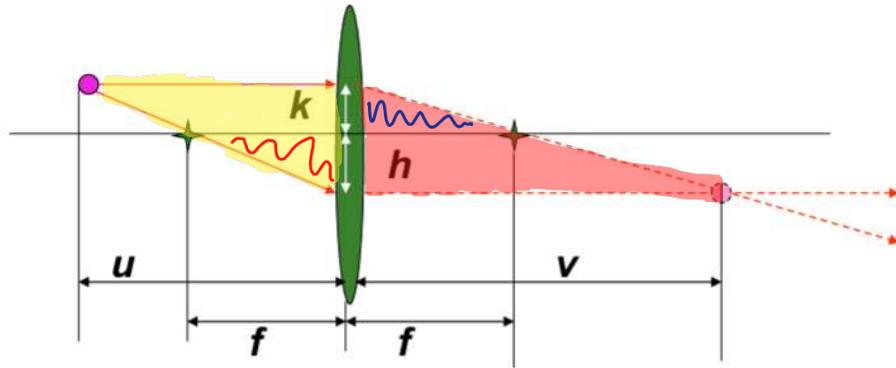
Un risultato migliore può essere ottenuto con l'ausilio di lenti sottili, ovvero lenti dallo spessore trascurabile.

Queste lenti hanno diverse proprietà:

9.4.1 Proprietà geometriche della lente sottile

in un

- Raggi paralleli all'asse della lente sottile che passano attraverso essa, vengono proiettati tutti attraverso un punto specifico oltre la lente, chiamato fuoco.
- Una lente sottile ha due fuochi equidistanti da essa.
- Raggi che passano attraverso il fuoco, sono ritrasmetti tutti paralleli nella direzione dell'asse della lente.



9.4.2 Equazione della lente sottile

Andiamo a trovare un'equazione che metta in relazione i valori indicati nella seguente immagine:
Individuando dei triangoli simili, possiamo arrivare alla seguente conclusione:

$$\frac{u}{f} = \frac{h}{h+k} \quad \text{e triangolo di base } u \text{ e altezza } h+k \text{ è simile a quelle di base } f \text{ e altezza } h$$

Conseguentemente:

$$\frac{v}{f} = \frac{k}{h+k} \quad \text{e triangolo di base } v \text{ e altezza } h+k \text{ è simile a quelle di base } f \text{ e altezza } k$$

Analogamente, diremo anche che:

$$\frac{v}{f} = \frac{u}{u+h} \rightarrow v \cdot k = u \cdot h \rightarrow \frac{v \cdot k}{u} = \frac{u \cdot h}{u+h}$$

Prendendo

Da $(h+k) = \frac{u \cdot h}{f}$ dividiamo entrambi i membri per u ottenendo:

$$\frac{h}{u} + \frac{k}{u} = \frac{h}{f}$$

Ma $\frac{k}{u} = \frac{h}{v}$, e quindi;

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

Eliminiamo il fattore comune e ottendiamo definitivamente l'equazione della lente sottile, ovvero:

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

9.4.3 Messa a fuoco, equazione della lente sottile

Se f si misura in metri, $\frac{1}{f} = 1$ diottria.

- Quando abbiamo una lente fissa, come nelle fotocamere, f è una quantità costante. Se la distanza dall'oggetto indicata con u aumenta, la distanza lente - sensore deve diminuire.,
- Se il piano dei sensori non può allontanarsi, come nel caso della pupilla, e quindi v è fissa, ciò che possiamo fare è aggiustare la lunghezza focale (f), mettendo in tensione il cristallino.
- Due oggetti a distanza u_1 e u_2 dalla lente sottile, distanze di molto superiori a f , appariranno approssimativamente sullo stesso piano (in quanto v_1 e v_2 saranno valori molto vicini); ciò non succede invece quando u_1 ed u_2 sono a distanze differenti e comparabili (meno di 30 volte la distanza della lente), allora non sono focalizzabili contemporaneamente. Si presenterà così il fenomeno della "profondità di campo".

9.4.4 Magnificazione

Col termine magnificazione andiamo a indicare la proprietà, della lente sottile, di alterare la dimensione dell'immagine di un oggetto rispetto alla dimensione effettiva di esso. Il fattore di magnificazione è dato da:

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

Moltiplichiamo per v l'equazione della lente sottile

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

Ovvvero

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

Invertiamo:

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

Moltiplichiamo per u e sostituiamo $\frac{u}{v} = \frac{1}{m}$.

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

Moltiplichiamo per m ottenendo finalmente una relazione tra il fuoco, la distanza dall'oggetto e il fattore di magnificazione.

$$f = u \cdot \frac{m}{m + 1}$$

[Annoto qua per comodità tutte le variabili:]

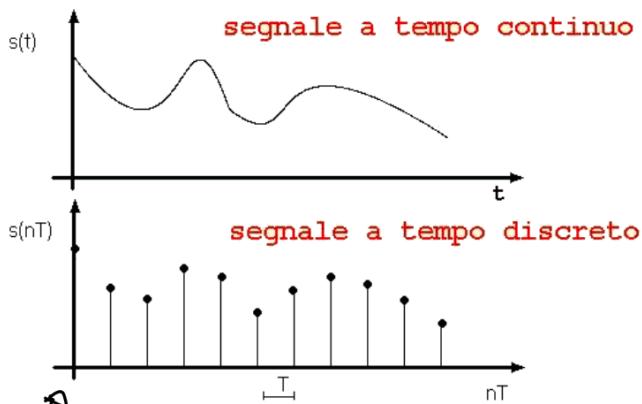
- u - distanza oggetto - lente
- v - distanza lente - sensore
- f - fuoco
- m - fattore di magnificazione

9.4.5 Esercizi sul calcolo:

- Oggetto a distanza $u = 1\text{cm}$.

Macchina analogica con $v = 2\text{cm}$.

Se la farfalla è a distanza $u_1 = 5\text{cm}$ si ha $m = \frac{2}{1} = 2$ e $f = 5 \cdot \frac{2}{3} = 3.3\text{cm}$



Questo è necessario perché un continuo non può lavorare con infiniti punti, in pratica si scelgono dei campioni rappresentativi del segnale originale

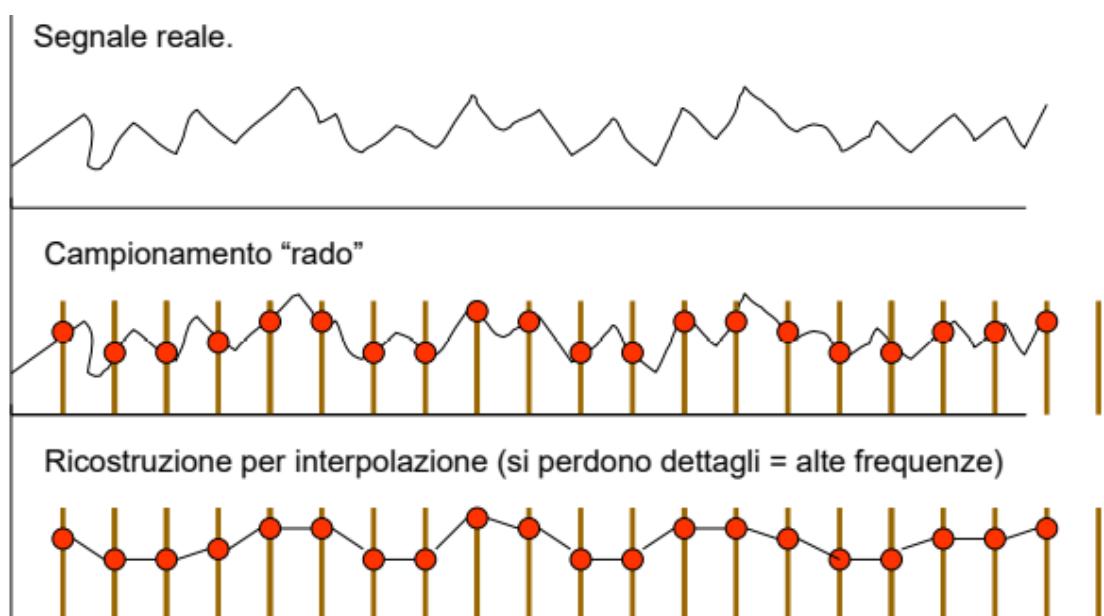
Lezione 6 – Campionamento & Quantizzazione

Il campionamento

Nella teoria dei segnali il **campionamento** è una tecnica che consiste nel convertire un **segnale continuo** nel tempo oppure nello spazio in un **segnale discreto**, valutandone l'ampiezza con un **numero finito di campioni** rappresentativi del segnale cosicché da avere **valori discreti** anziché **numeri reali**.

È importante eseguire un campionamento in modo corretto, poiché un errore nel tasso di campionamento può provocare:

1. Un **campionamento troppo basso** può causare la **perdita di dettagli ed informazioni**: è obbligo notare che, nonostante quest'ultima sia grave, è spesso necessaria poiché non è possibile conservare milioni di campioni e mantenere il database delle misure ottenute in dimensioni maneggevoli;



2. Un campionamento troppo basso può far apparire nelle immagini dettagli non presenti nell'originale, alterando il segnale in qualcosa' altro. Un esempio di ciò può essere il fenomeno di aliasing, il quale fa sì che le alte frequenze siano trattate come basse frequenze durante la fase di campionamento;

La Nyquist rate e il teorema di Shannon

Per decifrare il giusto valore di campionamento si ricorre al teorema di Shannon che si basa sulle misure del Nyquist rate

Si definisce Nyquist rate il doppio delle più alte frequenze in un segnale continuo e limitato

Teorema di Shannon: Se si realizzano campioni con frequenze più alte del Nyquist rate il segnale può essere ricostruito fedelmente in ogni suo punto

Se si campiona ad una frequenza più bassa rispetto l'aliasing che porta le alte frequenze ad essere mosse verso de brani e quindi tranne molte durante le quantizzazioni

La quantizzazione (Uniforme, non uniforme, logaritmica)

Nella teoria dei segnali, la quantizzazione è un processo che permette di passare da numeri reali, frutto delle misurazioni dei sensori, a numeri discreti e limitati all'interno di un determinato range attraverso arrotondamenti e normalizzazioni.

Vi siano dei numeri reali nel range [a-b] e si vogliano quantizzare in N livelli:

- Si fissano $n+1$ in [a-b] tali che: $t_0 = a < t_1 < t_2 \dots < t_n < t_{n+1} = b$
- Il numero x in [a-b] verrà assegnato al livello di quantizzazione k tale che: $t_k \leq x < t_{k+1}$

A ciascun livello di quantizzazione verrà assegnato un valore rappresentativo.

Nella quantizzazione uniforme il salto di quantizzazione è sempre costante:

- Range di ingresso $0 \dots N-1$
- Range in uscita $0 \dots K-1$ con $K \leq N$ (K è il numero di livelli di quantizzazione, N è il valore massimo)
- Siano L il livello d'ingresso e L' il corrispondente dopo la quantizzazione

$$L' = (L * K) / N$$

Nella quantizzazione non uniforme il salto di quantizzazione non è costante:

- Range di ingresso $0 \dots N-1$
- Range in uscita $0 \dots K-1$ con $K \leq N$ (K è il numero di livelli di quantizzazione, N è il valore massimo)
- Siano L il livello d'ingresso e L' il corrispondente dopo la ri-quantizzazione

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

- Dove $f(L, K, N)$ definisce lo schema di ri-quantizzazione

La quantizzazione logaritmica è una quantizzazione non uniforme in cui lo schema di ri-quantizzazione è:

$$L' = f(L, K, N) = \frac{\log_2(L) * K}{\log_2(N)}$$

10.3.5 Esempio di quantizzazione uniforme

Immagine a 8 bit \Rightarrow 256 livelli.

$N = 256$

$$M = \begin{bmatrix} 0 & 100 \\ 50 & 200 \end{bmatrix}$$

Vogliamo quantizzarla a 3 bit \Rightarrow 8 livelli.

$K = 8$

$$L'_{11} = (0 \cdot 8) / 256 = 0$$

$$L'_{12} = (100 \cdot 8) / 256 = 3,125$$

$$L'_{21} = (50 \cdot 8) / 256 = 1,56$$

$$L'_{22} = (200 \cdot 8) / 256 = 6,25$$

Prendiamo solo la parte intera dei numeri reali ottenuti. Matrice risultante:

$$M' = \begin{bmatrix} 0 & 3 \\ 1 & 6 \end{bmatrix}$$

10.3.7 Esempio di quantizzazione non-uniforme (logaritmica)

Immagine a 8 bit \Rightarrow 256 livelli.

$N = 256$

$$M = \begin{bmatrix} 0 & 100 \\ 50 & 200 \end{bmatrix}$$

Vogliamo quantizzarla a 3 bit \Rightarrow 8 livelli.

$K = 8$

$$L'_{11} = (\log_2(0) \cdot 8) / \log_2(256) = \text{Non calcolabile} = 0$$

$$L'_{12} = (\log_2(100) \cdot 8) / \log_2(256) = \log_2(100) = 6,64$$

$$L'_{21} = (\log_2(50) \cdot 8) / \log_2(256) = \log_2(50) =$$

$$L'_{22} = (\log_2(200) \cdot 8) / \log_2(256) = \log_2(200) =$$

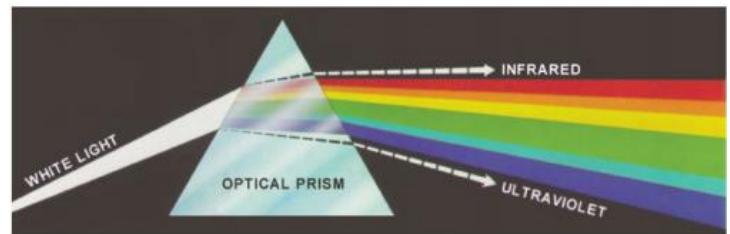
Prendiamo solo la parte intera dei numeri reali ottenuti. Matrice risultante:

$$M' = \begin{bmatrix} 0 & 3 \\ 1 & 6 \end{bmatrix}$$

Lezione 7 – Colori

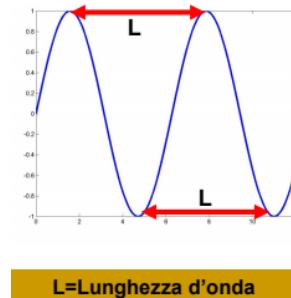
La luce e il colore

Nel 1666 Isaac Newton ha scoperto che se un raggio luminoso bianco attraversa un prisma di vetro ciò che si ottiene non è una luce bianca ma uno spettro di colori che va dal violetto al rosso. Si è quindi dedotto che la luce può essere decomposta in onde luminose di tipo differente.



Il nostro occhio percepisce solo una piccola porzione dello spettro elettromagnetico, inoltre si è scoperto che i colori sono legati alla lunghezza d'onda.

- La luce è quindi formata da tutte le lunghezze d'onda percepite dall'occhio umano;
- Lo spettro della luce visibile oscilla da un violetto a un rosso, che per comodità è suddiviso in 6 regioni: violetto, blu, verde, giallo, arancio e rosso;
- Le bande di colori che rappresentano le 6 regioni non sono della stessa grandezza e degradano nelle bande limitrofe;
- Il nostro occhio percepisce come colore di un oggetto la luce che quest'ultimo riflette;



Lung. in nanometri	Tipo radiazione
$10^{17} - 10^{13}$	Osc.elettriche
$10^{13} - 10^9$	Onde radio
$10^9 - 10^6$	Micro-onde
$10^6 - 10^3$	Infrarosso
$10^3 - 10^2$	Visibile
$10^2 - 10$	Ultravioletto
$10 - 10^{-3}$	Raggi X
$10^{-3} - 10^{-7}$	Raggi gamma e cosmici

Un nanometro = 1 metro / 1.000.000.000

Per descrivere la luce si utilizzano i seguenti valori:

- **Radianza:** ovvero la quantità di luce emessa dalla sorgente luminosa;
- **Luminanza:** ovvero la quantità di luce percepita dall'utente;
- **Brillantezza:** un valore soggettivo che indica la sensazione di colore;

La percezione umana del colore (e teorema del tristimolo di Young)

Come abbiamo già detto, l'uomo vede attraverso l'occhio ma elabora con il cervello. Nella retina dell'occhio sono presenti i coni che possono essere distinti in:

- **Coni di tipo S:** sensibili alle lunghezze d'onda corte, e quindi sensibili ai colori bluastri;
- **Coni di tipo M:** sensibili alle lunghezze d'onda media, e quindi sensibili ai colori verdastri;
- **Coni di tipo L:** sensibili alle lunghezze d'onda lunghe, e quindi sensibili ai colori rossastri;

Nel 1802 Young ha teorizzato che tutti i colori si possono ottenere mescolando i tre colori fondamentali in proporzioni differenti. Tale ipotesi è falsa, dimostrato successivamente con il diagramma di cromaticità CIE.

↑
teoreme
del tristimolo

Non esiste una scala fisica per misurare il colore, poiché il colore è una questione di percezione e di interpretazione soggettiva: il riconoscimento dei colori da parte dell'uomo dipende da molti fattori come il tipo di luce, gli oggetti,

Differenze di colore

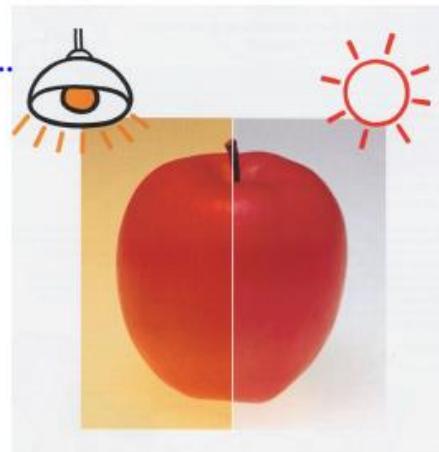
Differenze di sorgenti luminose
luce solare, lampade ad incandescenza, neon,...

Differenze di osservatore
sensibilità, vista,...

Differenze di dimensioni
Effetto area = colori che ricoprono vaste superfici appaiono più luminosi e vivaci

Differenze di sfondo
Effetto contrasto = colori appaiono più cupi se lo è sfondo scuro

Differenze di direzione
Caratteristiche direzionali di alcune vernici
costanza degli angoli di osservazione e di illuminazione



→ misure

gli occhi e cervello dell'osservatore.

Si definisce illuminante l'energia radiante con distribuzione spettrale di energia relativa definita nel campo di lunghezza d'onda capace di influenzare la visione del colore degli oggetti.

Gli illuminanti *A, B, C, D65*, sono stati definiti dal CIE.

La composizione dei colori

Illuminando una superficie bianca (coefficiente di riflessione del 100%) con una o più luci monocromatiche si ottiene per riflessione una radiazione colorata per sintesi adattiva: qualsiasi luce di spettro complesso può essere ottenuta come somma di tre luci monocromatiche agenti simultaneamente;

Lo spazio di colore CIE XYZ

portando de queste ipotesi

Nel 1931 il CIE ha fissato le lunghezze d'onda standard per i tre colori primari, per poi ratificare nel 1964:

Valori del 1931: $B = 435,8 \text{ nm}$; $R = 700 \text{ nm}$; $G = 546,1 \text{ nm}$

Valori del 1964: $B = 445 \text{ nm}$; $R = 575 \text{ nm}$; $G = 535 \text{ nm}$

Basandosi sulla sintesi adattiva, i valori del tristimolo XYZ sono così definiti:

- Siano definite le tre *color matching functions* $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$, i cui risultati indicano la quantità di colore primario necessaria a ottenere il colore desiderato;
- Sia λ della luce monocromatica;
- Sia $S(\lambda)$ la distribuzione spettrale relativa dell'illuminante;
- Sia $R(\lambda)$ la riflettanza spettrale del soggetto;

$$(\text{sensibilità rossa}) X = \int_{380}^{780} S(\lambda) \bar{x}(\lambda) R(\lambda) d\lambda$$

$$(\text{luminosità}) Y = \int_{380}^{780} S(\lambda) \bar{y}(\lambda) R(\lambda) d\lambda$$

$$(\text{stimolazione del blu}) Z = \int_{380}^{780} S(\lambda) \bar{z}(\lambda) R(\lambda) d\lambda$$

12.1.4 Diagramma cromatico CIE

- x è la quantità di rosso
- y è la quantità di verde
- $z = 1 - (x + y)$ è la quantità di blu, derivabile da x, y

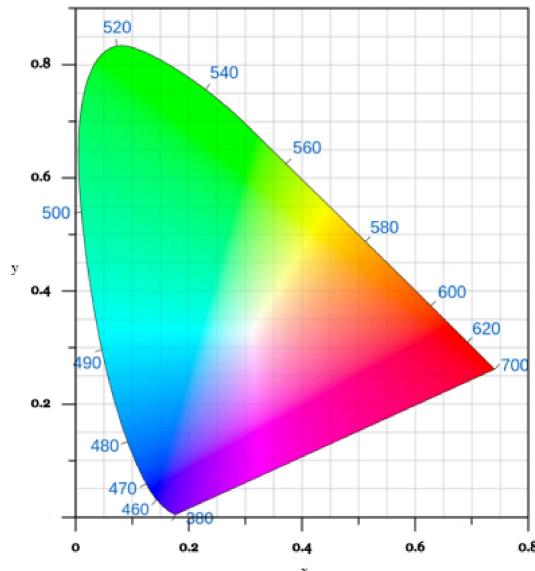
La rappresentazione grafica del colore al variare di x e y da origine al diagramma cromatico CIE, di cui possiamo denotare alcune caratteristiche interessanti:

referto

- Il punto di uguale energia tra le tre componenti contiene il bianco. Tutti i punti ai bordi non ha bianco, e sono detti colori puri.
- Unendo due colori con una linea, tutti i colori nella linea sono ottenibili mescolando i due colori.
- Unendo un colore con il bianco si ottengono tutte le tonalità di quel colore.
- Unendo tre colori con un triangolo, tutti i colori lungo il bordo e nel triangolo sono quelli ottenibili mischiando quei tre colori.
- Non è possibile creare un triangolo completamente all'interno di quest'area che copra tutti i possibili colori. Questo sfata la teoria del tristimolo.

È impossibile fissare tre colori fondamentali standard per rappresentare l'intera gamma di colori.

CIE chromaticiy diagram 1931



Questo diagramme è solo
una fetta di uno spazio
più completo ovvero il
CIE XYZ (tridimensionale)
questo è un derivato 2D
senza le luminanze viene
costituito sulle basi delle
seguente relazione:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

Definizione di Modello di colore (o spazio dei colori)

Lo scopo di un modello del colore (o spazio dei colori) è di consentirne la specificazione dei colori con modalità standardizzate, che fanno normalmente riferimento ad un sistema di coordinate 3-D (dato che tre sono comunque le caratteristiche che definiscono un colore), o meglio ad un suo sotto-spazio, nel quale ogni colore è rappresentato da un punto.

Nell'image processing, i modelli del colore sono interessati in più ambiti:

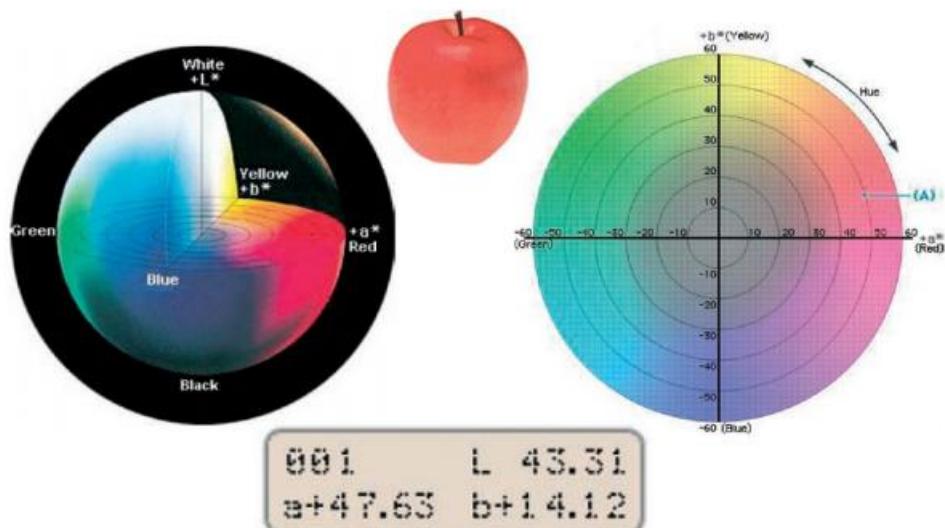
- Acquisizione (RGB) e restituzione (RGB, CMY) delle immagini;
- Trasmissione (YUV, YIQ) delle immagini;
- Compressione (YCbCr) delle immagini;
- Elaborazione (od analisi) delle immagini mediante trattamento del colore (RGB, HSI, HSV, LUV, ...);

Lo spazio di colore CIE L*a*b*

Il principale problema dello spazio di colore CIE XYZ (e dei suoi derivati) è che esso non è dotato di *uniformità percettiva*:

- Siano C_1 e C_2 due colori del CIE XYZ e ΔC una distanza;
- Presi due colori $C_3 = C_1 + \Delta C$ e $C_4 = C_2 + \Delta C$, supponendo che la distanza sia analoga per entrambi i colori, sarebbe desiderabile che i due colori C_3 e C_4 fossero percepiti egualmente distanti dai rispettivi C_1 e C_2 ;
- Ciò non avviene poiché lo spazio di colore CIE XYZ non è percettivamente uniforme;

Nel 1976, dopo anni di elaborazione, la CIE ha standardizzato uno spazio di colore percettivamente uniforme detto **CIE L*a*b***: in cui L^* rappresenta la *luminanza*, a^* e b^* rappresentano le coordinate di cromaticità, la presenza della luminanza e della crominanza fa sì che venga chiamato *rappresentazione luminanza-crominanza*. In confronto al CIE XYZ, il CIELAB rispetta maggiormente la risposta non lineare dell'occhio umano alle radiazioni.

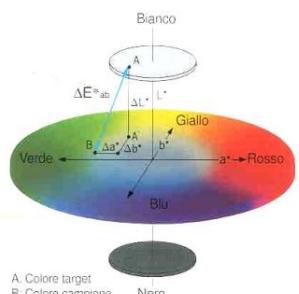


elettromagnetiche.

È possibile passare dallo spazio di colore CIE XYZ allo spazio di colore CIELAB e viceversa con le seguenti formule:

$$(\text{luminanza}) L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16; \quad a^* = 500 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} \right]; \quad b^* = 200 \left[\left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n} \right)^{\frac{1}{3}} \right]$$

- I valori $\frac{X}{X_n}$, $\frac{Y}{Y_n}$ e $\frac{Z}{Z_n}$ sono sempre > 0.01 ;
- I valori X_n , Y_n e Z_n rappresentano il punto bianco;



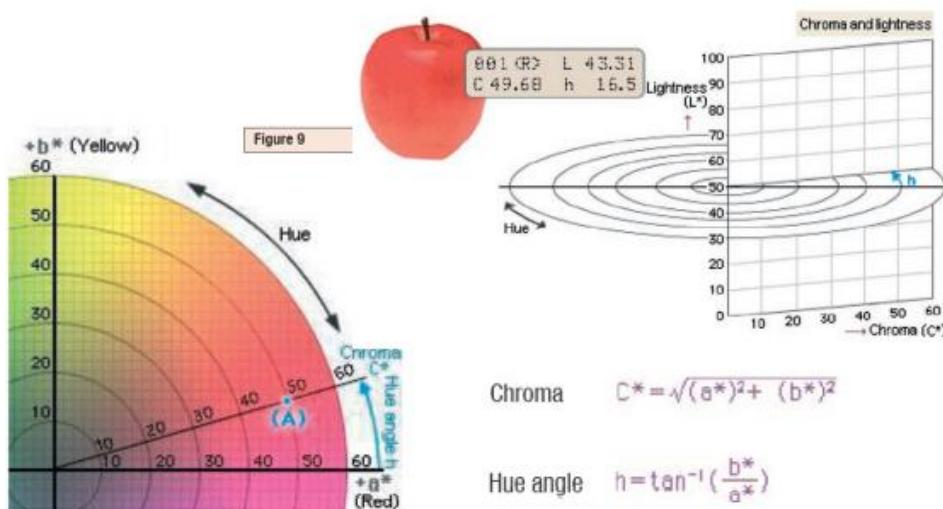
Nello spazio di colore CIELAB le differenze di colore sono definite come distanze fra due punti:

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

Lo spazio di colore CIE L*a*b*

Sempre nel 1976 la CIE ha standardizzato un altro spazio di colore percettivamente uniforme detto **CIE L*C*h***: in cui L^* rappresenta la *luminanza*, C^* rappresenta la *saturazione* e h^* rappresenta la *tinta*. C^* e h^* sono coordinate polari, rispettivamente raggio e angolo. Il CIELCH usa lo stesso diagramma del CIELAB.

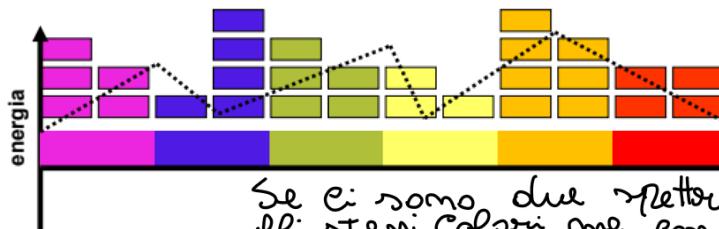
$$C^* = \sqrt{a^*{}^2 + b^*{}^2}; \quad h^* = \tan^{-1} \left(\frac{b^*}{a^*} \right)$$



Lo spazio di colore HSV (o HSI) e piccola definizione Munshell System (H, C, V)

In natura raramente vediamo colori puri, piuttosto vediamo miscele di radiazione luminosa in ogni lunghezza d'onda. Ogni lunghezza d'onda trasporta differenti quantità di energia e l'energia totale della radiazione è la somma di tutti i contributi di energie dalle diverse lunghezze d'onda.

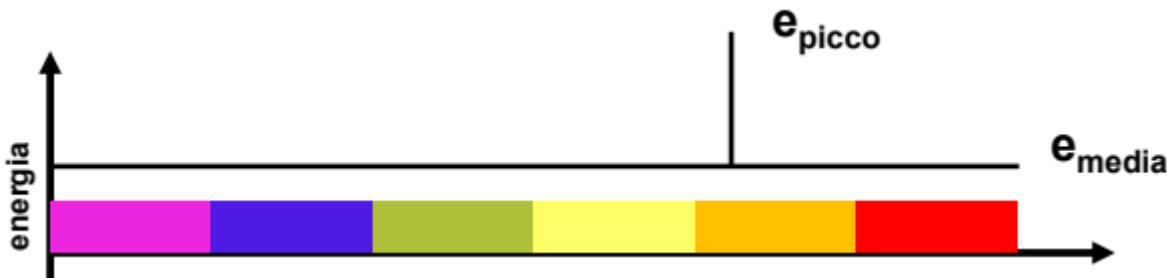
Lo spettro di un illuminante è il diagramma dei contributi di energia che esso apporta per ciascuna differente lunghezza d'onda.



Se ci sono due spettri che producono gli stessi colori ma con lunghezze d'onda differenti

Spettri diversi possono produrre colori uguali, le coppie di spettri con tale proprietà vengono chiamati **metameri**.

Per ogni colore esisterà sempre un metamero della seguente forma:



Dove:

- La lunghezza d'onda in cui si ha il picco è responsabile del *colore percepito*;
- Il rapporto $\frac{e_{picco} - e_{media}}{e_{picco} + e_{media}}$ è la *saturazione*, cioè quanto è puro il colore. Meno è la luce bianca, maggiore è il valore del rapporto;
- e_{media} è proporzionale al contenuto energetico della radiazione, può essere considerata come una misurazione della *luminosità* della radiazione;

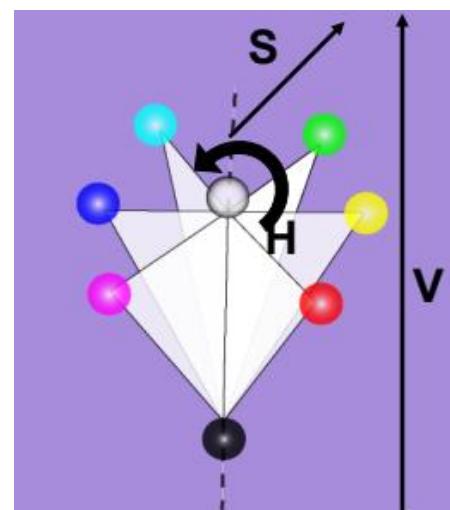
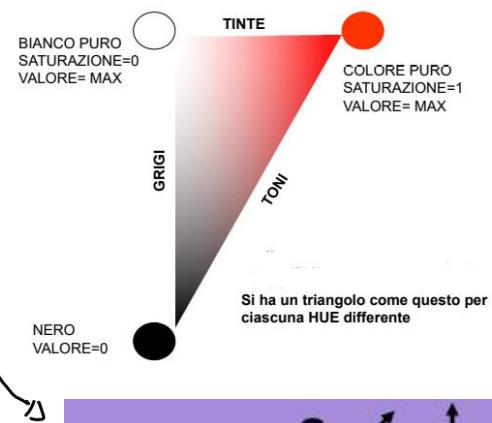
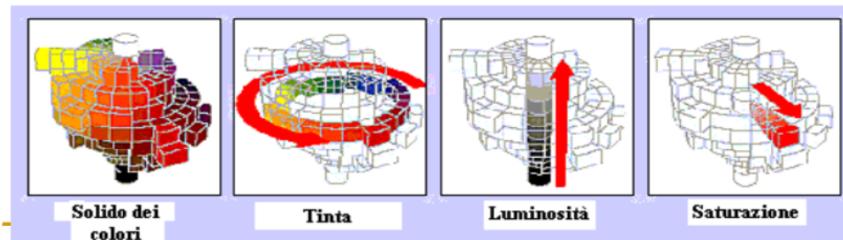
Mettendo insieme i tre concetti per ogni possibile tinta (*hue*) viene fuori un triangolo come quello a lato.

Azzerando il valore della saturazione, il cateto rappresentante il valore assume la forma di una scala di grigi comune a tutte le tinte. È quindi possibile unire i triangoli di tutte le tinte attraverso il lato comune.

Nello spazio HSV così definito si ha che ogni colore è identificato per:

- H: Hue* (o tinta), che copre tutti i colori;
- S: Saturazione*, che va da un minimo posto al centro pari al bianco ad un massimo pari al colore puro;
- V: valore/luminosità*, che va da un minimo ad un massimo;

Sulla stessa ideologia esiste il *Munshell System*, costituito da un cilindro sezionato in altezza, ogni sezione corrisponde a livello di livello di luminosità, la distanza dal centro corrisponde al croma e l'angolo alla tinta.



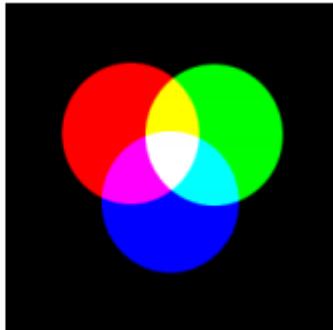
La sintesi additiva del RGB

Nello spazio dei colori RGB, i colori si formano per sintesi additiva partendo dai tre colori primari: rosso R, verde G e blu B:

- Dalla loro sovrapposizione si ottiene il bianco W;
- Dalla sovrapposizione di due dei colori si ottiene, a seconda dei colori che si sovrappongono): giallo Y, magenta M, ciano C;

In tale contesto si definisce *colore complementare* di un colore C quel colore CC tale che la somma dei due colori dia il bianco.

$$\begin{aligned}R + G + B &= W \\R + G &= W - B = Y \\R + B &= W - G = M \\B + G &= W - R = C\end{aligned}$$



Il Giallo è complementare del Blu
Il Magenta è complementare del Verde
Il Ciano è complementare del Rosso

Su questo principio si basa il funzionamento di monitor e schermi televisivi.

Nello spazio di colore RGB con sintesi additiva si definiscono *colori primari* e *colori secondari*:

- I colori primari sono il rosso R, verde G e blu B, è importante notare che combinandoli fra loro non si ottengono tutti i colori visibili;
 - Per poter rappresentare tutti i colori visibili con RGB si dovrebbe permettere anche alla lunghezza d'onda di variare, ma in questo caso non si avrebbero più tre colori primari standard fissi;
- I colori secondari sono i complementari dei primari, ovvero magenta M, giallo Y, ciano C, che costituiscono lo spazio di colore complementare del RGB detto CMY;

Lo spazio di colore RGB risulta quindi uno spazio di colore di facile utilizzo e implementazione in software e hardware ma è percettivamente poco comodo in quanto non è possibile capire guardando un colore in natura in quale proporzione vi contribuiscono i colori primari.

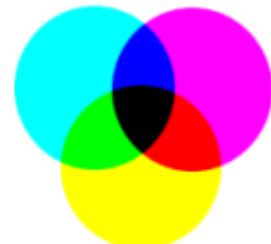
La sintesi sottrattiva del CMY

Un filtro ottico colorato è un mezzo a facce piane e parallele che attraversato dalla luce bianca ne assorbe selettivamente una porzione. Il colore del filtro è equivalente al colore prodotto dalla radiazione complementare della radiazione che si vuole assorbire.

Sovrapponendo tre filtri di colore giallo Y, magenta M, ciano C su un visore luminoso bianco si ottengono i colori per sintesi sottrattiva.

- Sovrapponendo tutti e tre i filtri verrà assorbita tutta la radiazione visibile e si otterrà il nero K;
- Soprapponendo due filtri si ottiene il colore corrispondente alla componente luminosa che non viene assorbita dai filtri, che può essere rosso R, verde G e blu B;

$$\begin{aligned}\mathbf{Y} + \mathbf{M} &= \mathbf{R} \\ \mathbf{Y} + \mathbf{C} &= \mathbf{G} \\ \mathbf{M} + \mathbf{C} &= \mathbf{B} \\ \mathbf{Y} + \mathbf{M} + \mathbf{C} &= \mathbf{K}\end{aligned}$$



Su questo principio si basa la riproduzione delle immagini a colorate nella fotografia a colori e nella stampa con inchiostro.

che tiene separata

Lo standard YUV

Lo spazio **YUV** è una famiglia di spazi che viene utilizzata per la **codifica** di immagini o video analogici, tenendo separata la **luminanza dalla crominanza**.

- La **luminanza** può essere **ottenuta attraverso una combinazione lineare delle intensità luminose dei canali rosso, verde e blu di RGB**, un'approssimazione abbastanza fedele è la seguente:

$$Y = 0.3 * R + 0.6 * G + 0.1 * B$$

- Il termine **crominanza** è definito come la differenza tra il colore e un bianco di riferimento alla stessa luminanza, U rappresenta la crominanza del blu e V la crominanza del rosso:

$$U = 0.6(B - Y) \rightarrow U = -0.2R - 0.3G + 0.5B$$

$$V = 0.7(R - Y) \rightarrow V = 0.5R - 0.4G - 0.1B$$

A seconda dei valori assunti da RGB si ha che:

- Se R, G, B sono compresi fra 0 e 1 si ottengono valori di Y compresi fra 0 e 1 e valori di U e V compresi fra -0.5 e 0.5;
- Se $R = G = B$, allora U e V valgono 0 e si ottengono solo grigi;

Lo standard YC_bC_r

Lo spazio YC_bC_r è una famiglia di spazi che costituiscono la **controparte digitale dello spazio YUV**. È possibile ottenere facilmente lo spazio YC_bC_r normalizzando ed eventualmente quantizzando i canali di YUV.

- La **luminanza** può essere **ottenuta attraverso una combinazione lineare delle intensità luminose dei canali rosso, verde e blu di RGB**, un'approssimazione abbastanza fedele è la seguente:

$$Y = 0.3 * R + 0.6 * G + 0.1 * B$$

- C_b e C_r rappresentano la crominanza del blu e del rosso, se i valori del RGB sono compresi fra 0 e 255 è possibile ottenere le crominanze shiftando U e V:

$$C_b = 128 + U$$

$$C_r = 128 + V$$

Uno dei principali impieghi di tale spazio è la compressione, è infatti utilizzato nel formato JPEG.

La palette o Look-Up-Table (LUT), l'indicizzazione di un'immagine true color e il re-indexing

In un'immagine si hanno più colori che pixel, inoltre le immagini **naturali** sono caratterizzate da una **proprietà di coerenza interna** per cui raramente si ha un colore differente per ogni differente pixel.

Su tale premessa si basa l'idea della palette, o look-up-table, che consiste di creare una tavolozza costituita da etichette che corrispondono a componenti RGB e assegnare ad ogni pixel l'etichetta anziché una componente RGB, risparmiando così spazio.

Molti software commerciali adottano una palette a 256 colori. L'utilizzo di una palette **custom**, ovvero adattata all'immagine, sarà così definita:

- Se nell'immagine true color ci sono meno di 256 colori, alcuni di essi verranno replicati;
- Se nell'immagine true color ci sono più di 256 colori, essi verranno ridotti scegliendo 256 colori rappresentativi che garantiscano una buona qualità visiva;

Il re-indexing consiste nel cambiare la posizione della palette e quindi anche l'indice corrispondente in modo da creare una matrice di indici che abbia l'entropia minima: ovvero si associano ai colori di pixel adiacenti indici numerici vicini, al fine di ottenere un miglior fattore di compressione.

Lezione 8/9 – Operazioni sulle immagini

L'istogramma

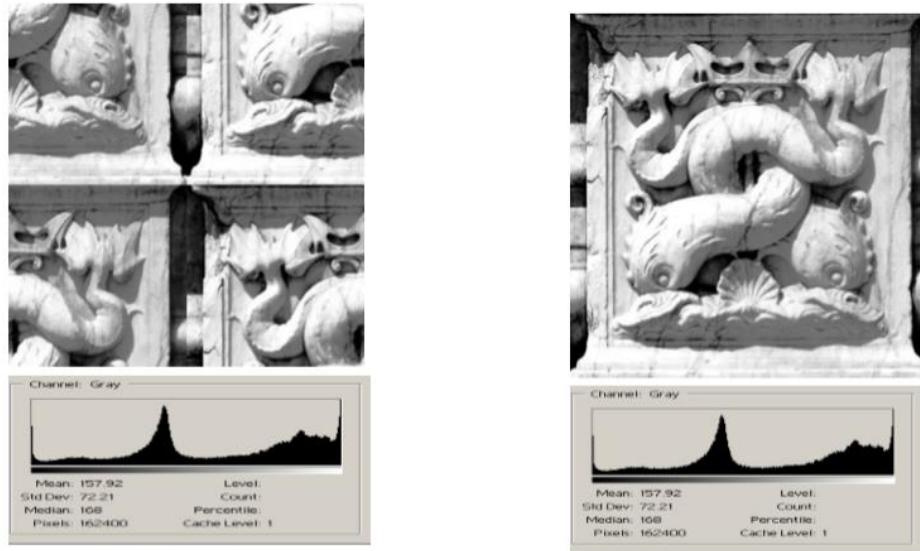
Un istogramma è un grafico che ci permette di comprendere in maniera immediata le caratteristiche di un'immagine. L'istogramma ci fornisce per ogni livello di grigio il numero di pixel che hanno quell'intensità.

$$H(k) = \text{il numero di pixel aventi valore } k$$

L'istogramma non tiene conto della distribuzione spaziale, due immagini diverse possono avere lo stesso istogramma.

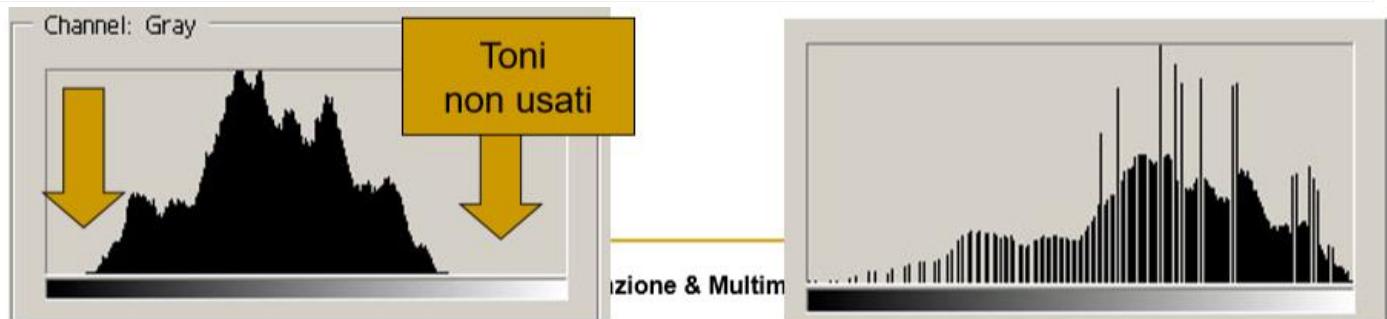
Una "lettura" veloce dell'istogramma può dirci se l'immagine è chiara, qualora l'istogramma fosse più denso a destra, o ~~sopra~~, qualora l'istogramma fosse più denso a sinistra.

Curve



Il contrast stretching (espansione del contrasto)

Operazione che agendo sull'istogramma permette di ottenere una distribuzione sull'istogramma che copre l'intero range di colori.



La normalizzazione

Operando aritmeticamente sulle immagini possiamo incorrere nei seguenti problemi:

- Un pixel ha valore negativo, risolvibile sostituendo il valore negativo con il minimo (tipicamente 0);
- Un pixel ha valore superiore al massimo, risolvibile sostituendo il valore con il massimo (tipicamente 255);
- Un pixel ha valore decimale, risolvibile tramite arrotondamento o troncamento;

I primi due problemi costituiscono i *problemi di range* di un'immagine.

Tramite normalizzazione si trasforma un range. Per esempio dato un range limitato dai valori A e B, lo si vuole trasformare in un range da 0 a 255:

$$V_{nuovo} = 255 * \frac{v_{vecchio} - MIN_{osservato}}{MAX_{osservato} - MIN_{osservato}}$$

L'equalizzazione

Si parla di immagine equalizzata quando il contributo di ogni differente tonalità di grigio è pressappoco eguale, in tal caso l'istogramma dell'immagine sarà uniforme o appiattito. L'equalizzazione non migliora sempre l'immagine.

Un algoritmo di equalizzazione è il seguente:

- 1) Sia r_k un livello di grigio e n_k il numero di pixel dell'immagine MN di quel livello, si definisce:

$$p_r(r_k) = \frac{n_k}{MN} \text{ con } (k = 0, 1, 2, 3, 4 \dots L - 1) (L = \text{numero di livelli})$$

- 2) I nuovi valori di grigio dell'istogramma saranno così definiti:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j)$$

- 3) Si arrotondano i valori ottenuti;

Operazioni sulle immagini e tipologie

Le operazioni sulle immagini alterano i valori dei pixel di un'immagine, ottenendo così un'immagine finale differente da quella iniziale. Tali operazioni sono egualmente attuabili su immagini a scala di grigio o a colori, per quest'ultime si dovrà applicare l'operatore su ogni canale.

Queste operazioni, definite come elaborazioni nel dominio spaziale posso essere espresse come:

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

Dove:

- f rappresenta l'immagine d'ingresso;
- g rappresenta l'immagine d'uscita;
- T è l'operatore dell'elaborazione su f definito in un intorno di (x, y) ;

A seconda della dimensione dell'intorno di (x, y) si caratterizzano i seguenti operatori:

- *Operatore puntuale*: quando l'intorno coincide con il pixel stesso;
- *Operatore locale*: quando l'intorno coincide corrisponde a una piccola regione quadrata (*finestra*) incentrata sul pixel;
- *Operatore globale*: quando l'intorno coincide con l'intera f ;

Operatori puntuali

Si definisce operatore puntuale, un operatore che preso in input il valore di un pixel ne restituisce uno cambiato che dipende esclusivamente dal valore del pixel in ingresso. Gli usi tipici di un operatore puntuale sono i seguenti:

- Aggiunta o sottrazione di una costante a tutti i pixel (per compensare una sotto/sovra esposizione);
- Inversione della scala dei colori (per ottenere il negativo);
- Espansione del contrasto;
- Modifica dell'istogramma (equalizzazione);

In un operatore puntuale $f(x, y)$ e $g(x, y)$ sono appartenenti allo stesso campo di definizione (es. entrambi da 0 a 255).

Una LUT (look up table) è un grafico che mostra la curva dell'operatore. Si sono studiati i seguenti operatori puntuali:

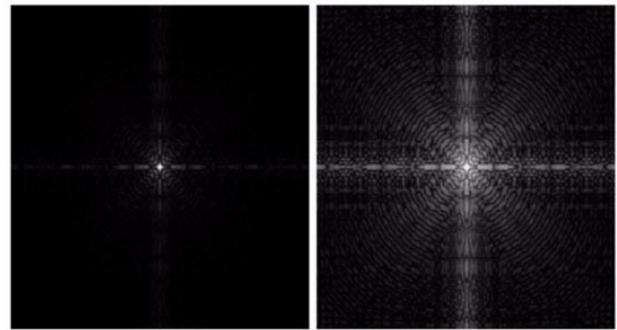
- *Negativo*: ogni pixel $g(x, y)$ assume il colore complementare del rispettivo $f(x, y)$: $g(x, y) = 255 - f(x, y)$. Nella LUT la curva corrisponde a una bisettrice inversa.



- **Logaritmo:** consiste nel comprimere la gamma dinamica, permettendo di visualizzazione di immagini caratterizzate da escursioni di intensità molto ampie. Ovvero rende più visibili le differenze di intensità che non sono individuabili a occhio nudo. Questa trasformazione logaritmica viene espressa come:

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

$$c = \frac{255}{\log(255)}$$



Dove c è una costante utilizzata per normalizzare il risultato in un determinato range, spesso 0-255.

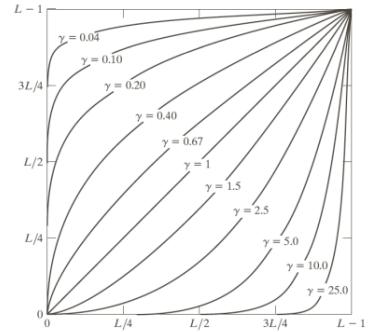
- **Potenza (Gamma):** la trasformazione di potenza può essere espressa come:

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

$$c = \frac{255}{255^\gamma}$$

La costante c viene utilizzata per normalizzare il risultato in un determinato range, spesso 0-255. Tale trasformazione ha il seguente comportamento:

- Per $\gamma < 1$: Si hanno effetti analoghi all'operatore logaritmo;
- Per $\gamma > 1$: Si hanno effetti opposti all'operatore logaritmo;
- Per $\gamma = 1$: Si ha l'effetto identità;



- **Binarizzazione:** tale trasformazione produce un'immagine composta da due livelli, bianco e nero; si ottiene scegliendo una soglia T e mettendo a nero tutti i pixel il cui valore è inferiore a T e a bianco tutti gli altri. Tale trasformazione comporta un istogramma a cui unici bin popolati sono gli estremi: 0 e 255.



- **Aumento e diminuzione del contrasto:** tali trasformazioni si ottengono rispettivamente:
 - L'aumento è ottenuto sostituendo un pixel con un altro avente un valore più scuro;
 - La diminuzione è ottenuta sostituendo un pixel con un altro avente un valore più chiaro;
- **Solarizzazione:** è una trasformazione avente una curva non monotona;

Operatori lineari e invarianti per traslazione

- **Operatore Lineare:** un operatore $F: V \rightarrow W$ si dice lineare se per ogni coppia di vettori v_1 e v_2 in V e per ogni coppia di scalari a e b si ha che:

$$F(av_1 + bv_2) = aF(v_1) + bF(v_2)$$
 - Se conosco la base di V e il comportamento dell'operatore F su ogni elemento di tale base, posso calcolare il comportamento di F su ogni elemento di V .
 - Per descriverlo basta conoscere il suo comportamento su tutte le immagini impulsive.
- **Operatore invariante per traslazione:** un operatore si dice invariante per traslazione quando il suo comportamento sulle immagini impulsive è sempre il medesimo indipendentemente dalla posizione in cui si trova il pixel.
 - Tutti gli operatori puntuali sono invarianti per traslazione (anche se non sono lineari).
- Se un operatore è sia lineare che invariante per traslazione per descriverlo basta conoscere come si comporta in un solo impulso, la point spread function (PSF) diventa la carta d'identità dell'operatore.
- Spesso un operatore su un'immagine prende il nome di filtro.

Il Kernel

Ad ogni operatore lineare e invariante per traslazione corrisponde una maschera ma vale anche il viceversa: ad ogni maschera corrisponde un operatore. Tale maschera si chiama spesso *kernel* o *maschera di convoluzione*.

La grandezza del kernel può variare fino ad essere infinito, ma per ragioni pratiche vengono usati solo kernel di dimensioni finite. Le dimensioni del kernel influenzano la complessità dell'operatore di filtraggio.

È importante sapere che la complessità dipende anche dal numero di pixel dell'immagine.

La convoluzione e il problema dei bordi

I filtri lineari e invarianti per traslazione vengono chiamati anche filtri convolutivi.

Per indicare l'operazione di convoluzione si usa la notazione: $h = f \otimes g$ e gode delle seguenti proprietà:

- Proprietà commutativa: $f \otimes g = g \otimes f$;
- Proprietà associativa: $(f \otimes g) \otimes h = f \otimes (g \otimes h)$

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

Un problema ricorrente è *come fare la convoluzione e il filtraggio ai bordi*, seguono alcune soluzioni:

- Filtrare solo le zone centrali e lasciare i bordi non calcolati;
- Supporre che tutto intorno all'immagine sia 0;
- Riempire le righe e colonne aggiunte in maniera toroidale;
- Riempire le righe e colonne aggiunte con i valori più vicini;

Operatori locali basici

Si definisce operatore locale, un operatore che preso in input il valore di un pixel ne restituisce uno cambiato che dipende da un intorno limitato centrato sul pixel in ingresso. Gli usi tipici di un operatore locale sono i seguenti:

- Migliorare la qualità delle immagini, come aumentare il contrasto o la nitidezza;
- Estrarre informazioni dalle immagini, come per esempio i bordi;

Tali operazioni si possono pensare come filtri dell'immagine, ottenuto facendo la convoluzione tra l'immagine ed il kernel dell'operatore.

Si sono studiati i seguenti operatori locali:

- *Mediano*: è un filtro non lineare che fornisce in uscita il valore mediano dell'intorno del pixel. Consiste in un effetto di smoothing dell'immagine;
- *Massimo*: è un filtro che fornisce in uscita il valore massimo dell'intorno del pixel. Consiste in uno schiarimento dell'immagine;
- *Minimo*: è un filtro che fornisce in uscita il valore minimo dell'intorno del pixel. Consiste in uno oscuramento dell'immagine;
- *N-box (di media)*: sono filtri definiti da kernel di dimensione $N \times N$ con ogni elemento pari a $\frac{1}{N^2}$ con N generalmente dispari. Consiste in un effetto di sfocatura dell'immagine, una sfocatura molto forte in orizzontale e verticale ma meno in diagonale;
- *N-binomiale*: sono filtri definiti da kernel derivati dalla distribuzione binomiale e, poiché tale distribuzione è una approssimazione discreta della distribuzione gaussiana, sono detti filtri gaussiani. Consiste in un effetto di sfocatura sull'immagine uguale in tutte le direzioni ma meno forte rispetto all'n-box;

Tali filtri vengono detti *energy preserving*, poiché la somma dei loro pesi fa 1.

I filtri appena visti servono anche a ridurre il rumore in un'immagine, più è grande è il kernel e migliore sarà il risultato rischiando di aumentare la sfocatura.

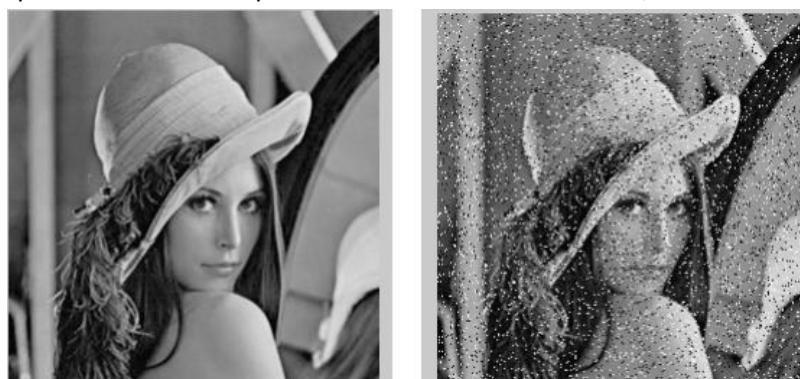
Il rumore

Ci sono due tipo principali di rumore:

- Rumore impulsivo, detto “*sale e pepe*”, caratterizzato dalla frazione dell’immagine modificata (in %):

$$p(z) = \begin{cases} P_a & \text{per } z = a \\ P_b & \text{per } z = b \\ 0 & \text{altrimenti} \end{cases}$$

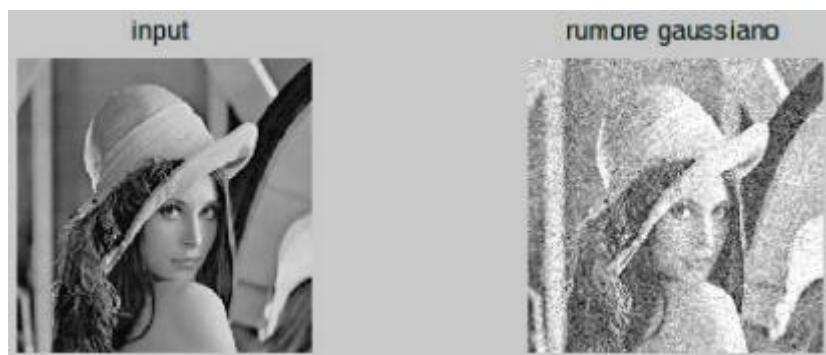
Il filtro ottimale per la rimozione di questo rumore è il filtro mediano;



- Rumore gaussiano bianco, caratterizzato dalla media e dalla varianza:

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

Il filtro ottimale per la rimozione di questo rumore è il filtro di media.



L'estrazione dei bordi

Un altro uso degli operatori locali è estrarre informazioni dalle immagini, una di queste informazioni sono i contorni, o bordi, che sono definiti come delle discontinuità locali della luminanza.

I filtri *edge detector* forniscono immagini in cui sono preservate le variazioni di luminanza ed elimina tutte le altre informazioni.

Alcuni *edge detector* sono basati sulla derivata prima. In un segnale monodimensionale se calcolo da derivata prima, scopro che i lati sono in corrispondenza dei massimi della derivata: è necessario calcolare la derivata della direzione x e poi della direzione y e poi combinarle.

- Sobel_x*: [(-1, -2, -1), (0, 0, 0), (1, 2, 1)], fornisce una matrice con i lati orizzontali (e le componenti orizzontali dei lati obliqui) che hanno valori non nulli;
- Prewitt_x*: [(-1, -1, -1), (0, 0, 0), (1, 1, 1)]], fornisce una matrice con i lati orizzontali (e le componenti orizzontali dei lati obliqui) che hanno valori non nulli;
- Sobel_y*: [(-1, 0, 1), (-2, 0, 2), (-1, 0, 1)], fornisce una matrice dei lati verticali (e le componenti verticali dei lati obliqui) che hanno valori non nulli;
- Prewitt_y*: [(-1, 0, 1), (-1, 0, 1), (-1, 0, 1)], fornisce una matrice dei lati verticali (e le componenti verticali dei lati obliqui) che hanno valori non nulli;

Questi filtri possono essere combinati mediante la formula: $\sqrt{\text{filtro}_x^2 + \text{filtro}_y^2}$

Si ottengono risultati migliori con algoritmi più sofisticati (e non lineari) o con strategie più intelligenti.

Alcuni *edge detector* sono basati sulla derivata seconda. In un segnale monodimensionale se calcolo da derivata seconda, scopro che i lati passano per lo zero.

Il filtro più diffuso di questa tipologia è il *Laplaciano* avente come maschera $[(-1,0,-1), (0,4,0), (-1,0,-1)]$, ma dopo aver applicato tale filtro è necessario che si verifichi la condizione *Zero-crossing*, ovvero rispetto al punto in questione vi deve essere nel suo intorno un valore positivo e un valore negativo.

Filtri di sharpening

Un filtro di sharpening incrementa la nitidezza di un'immagine aumentando il contrasto locale. Possiamo definirla come l'operazione opposta allo sfocamento.

Per ottenere tale effetto si può adottare una maschera derivante dal Laplaciano che rinforza i lati dell'immagine, come effetto collaterale vi è il rinforzamento del rumore:

$$[(-1,0,-1), (0,5,0), (-1,0,-1)]$$

Lezione 10 – Fourier

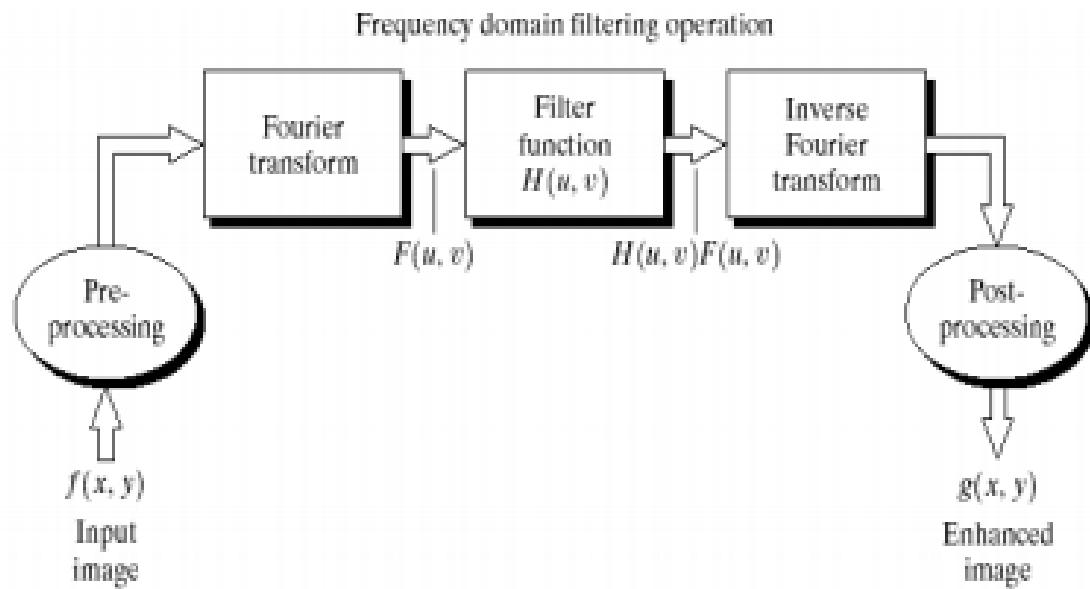
La serie, trasformata, dominio di Fourier

Una *funzione periodica* può essere espressa come *somma di seni e/o coseni di differenti frequenze e ampiezze*, tale somma prende il nome di *serie di Fourier*.

Anche una *funzione non periodica*, sotto certe condizioni, può essere espressa come *integrale di seni e/o coseni*, moltiplicati per opportune funzioni-peso, tale integrale prende il nome di *trasformata di Fourier*.

Sia la *serie di Fourier* che la *trasformata di Fourier* condividono il fatto che una funziona possa essere *ricostruita (recovered)* con un semplice processo di inversione senza perdita di informazione. È quindi possibile operare nel cosiddetto *dominio di Fourier*, detto anche *dominio delle frequenze*, e tornare nel dominio originale della funzione in maniera del tutto naturale.

Inizialmente l'analisi di Fourier trovò applicazione nel campo della diffusione del calore, in cui permise la formulazione e soluzione di equazioni differenziali di alcuni fenomeni fisici, ma con l'avvento della *Fast Fourier Transform (FFT)*, un algoritmo per il calcolo della *Discrete Fourier Transform*, il settore dell'elaborazione digitale dei segnali ha subito una vera e propria rivoluzione.



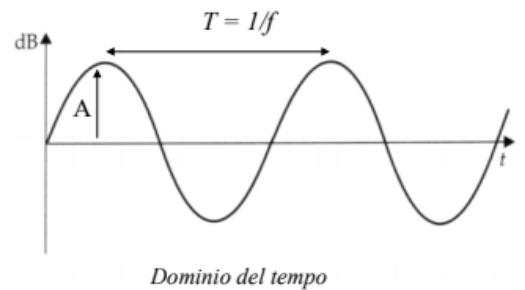
Immagini e Segnali

Un'immagine può essere vista come una *funzione discreta* in due dimensioni in cui i valori rappresentano il livello di grigio di un determinato pixel.

La funzione immagine così definita può essere vista come un segnale, cioè una funzione variabile in un dominio con una propria frequenza, quest'ultima costante o variabile.

In una tipica funzione sinusoidale possiamo distinguere le seguenti componenti:

- *Aampiezza (A)*, espressa in decibel dB;
- *Periodo (T)*, espresso in secondi;
- *Frequenza (f)*, espressa come numero di cicli (onde) al secondo e misurata in Hertz Hz;



La serie di Fourier, trasformata e anti-trasformata di Fourier in 1-D e 2-D

Nel caso unidimensionale, data una funzione variabile nel tempo e periodica $f(x)$, questa funzione può essere rappresentata come somma di infiniti termini sinusoidali, attraverso la *serie di Fourier*:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx))$$

Nonostante oggi si utilizzino altre trasformate in molte applicazioni, la *trasformata di Fourier* mantiene un ruolo cardine nell'elaborazione dei segnali e in particolare nell'image processing. La trasformata di Fourier di una funziona continua $f(x)$ variabile x , o di $f(x, y)$ se bidimensionale, sotto opportune ipotesi di continuità è definita come:

$$1D \Rightarrow F(u) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi ux} dx \rightarrow \text{Dove } e^{-2\pi ux} = \cos(2\pi ux) - i\sin(2\pi ux)$$

$$2D \Rightarrow F(u, v) = \iint_{-\infty}^{+\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

La trasformata sarà composta dalla somma di infiniti termini sinusoidali e cosinusoidali, le variabili u e v rappresentano le variabili frequenza, definite nel piano delle frequenze, che determinano la frequenza della coppia seno-coseno corrispondente.

Data $F(u)$ o $F(u, v)$ è possibile risalire rispettivamente a $f(x)$ o $f(x, y)$ attraverso l'*anti-trasformata* definita dalla seguente formula:

$$1D \Rightarrow f(x) = \int_{-\infty}^{+\infty} F(u) e^{i2\pi ux} dx \rightarrow \text{Dove } e^{2\pi ux} = \cos(2\pi ux) + i\sin(2\pi ux)$$

$$2D \Rightarrow f(x, y) = \iint_{-\infty}^{+\infty} F(u, v) e^{i2\pi(ux+vy)} dx dy$$

La *trasformata* e l'*anti-trasformata* esistono se $f(x)$ o $f(x, y)$ è continua e integrabile e se $F(u)$ o $F(u, v)$ è integrabile, inoltre se $f(x)$ o $f(x, y)$ è reale allora $F(u)$ o $F(u, v)$ è in generale complessa.

Possiamo dire che la trasformata di Fourier riorganizza i dati in un nuovo spazio, lo *spazio delle frequenze*.

Le componenti della trasformata di Fourier

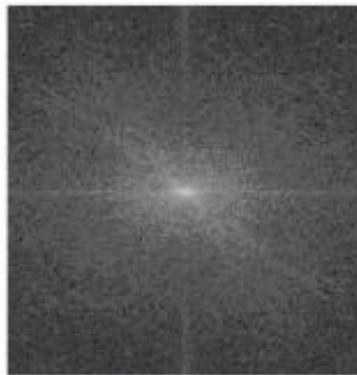
Dato che allora $F(u)$ o $F(u, v)$ è in generale complessa, è possibile definire:

- *Spettro della trasformata/Aampiezza:* $1D \Rightarrow |F(u)| = \sqrt{R^2(u) + I^2(u)}$ $2D \Rightarrow |F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$
- *Angolo di fase:* $1D \Rightarrow \phi(u) = \arctan\left(\frac{I(u)}{R(u)}\right)$ $2D \Rightarrow \phi(u, v) = \arctan\left(\frac{I(u, v)}{R(u, v)}\right)$

- *Potenza Spettrale:* $1D \Rightarrow P(u) = |F(u)|^2 = R^2(u) + I^2(u)$ $2D \Rightarrow P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$



$f(x,y)$



$|F(u,v)|$



$\Phi(u,v)$

La *fase* contiene le informazioni essenziali per la struttura dell'immagine, quelle cioè relative al dove le strutture periodiche evidenziate dalla trasformata siano collocate.

L'*ampiezza*, invece, contiene solo l'informazione relativa al fatto che una certa struttura periodica è presente nell'immagine.

Range dinamico

Quando si visualizza lo spettro di Fourier come immagine di intensità, esso manifesta in genere una dinamica molto più grande di quella riproducibile su un tipico display, per cui solo le parti più luminose dello spettro risulteranno visibili.

Dopo aver attuato una normalizzazione per poter visualizzare lo spettro in un display, sarà necessaria un ulteriore compressione di tipo logaritmico per mettere in risalto le già poche parti luminose. Invece dello spettro si visualizza una funzione di tipo:

$$D(u, v) = c \log(1 + F(u, v))$$

Dove c è una costante di scala, che va scelta opportunamente per far ricadere i valori trasformati nel range voluto.

Vantaggi dell'uso della Trasformata di Fourier

Operando nel *dominio di Fourier*, o *dominio delle frequenze*, è possibile:

- Sopprimere le frequenze indesiderate;
- Ridurre lo spazio dei dati pur limitando la degenerazione del segnale;
- Rigenerare un segnale degradato;

Trasformazione diretta e trasformazione inversa

La trasformazione diretta può essere vista come un processo di analisi: il segnale $f(x)$ viene scomposto nelle sue componenti elementari, che sono nella forma dei vettori di base. I coefficienti della trasformata specificano quanto di ogni componente di base è presente nel segnale.

Nella trasformazione inversa, mediante un processo di sintesi, il segnale viene ricostruito, come somma pesata delle componenti di base: il peso di ogni vettore di base nella ricostruzione del segnale è rappresentato dal corrispondente coefficiente della trasformata.

Il coefficiente della trasformata è una misura della correlazione tra il segnale ed il corrispondente vettore di base. La trasformazione non comporta perdita di informazione: essa fornisce solo una rappresentazione alternativa del segnale originale.

La Discrete Fourier Transform (DFT) in 1-D e 2-D

La *trasformata di Fourier discreta* e l'*anti-trasformata* di $f(x)$ o $f(x, y)$ assumono le seguenti forme:

- *Trasformata 1D* $\Rightarrow F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-\frac{i2\pi ux}{N}}$ per $u = 0 \dots N - 1$
- *Trasformata 2D* $\Rightarrow F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}$ per $u = 0 \dots M - 1$ e $v = 0 \dots N - 1$
- *Anti-trasformata 1D* $\Rightarrow f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{i2\pi ux}{N}}$ per $x = 0 \dots N - 1$
- *Anti-trasformata 2D* $\Rightarrow f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})}$ per $x = 0 \dots M - 1$ e $y = 0 \dots N - 1$

In caso di 2D, le variabili u e v sono gli indici relativi agli assi frequenza discretizzati, mentre M e N sono le dimensioni in pixel dell'immagine.

Inoltre, tutti i valori di $f(x, y)$ contribuiscono alla costruzione di ciascuno dei campioni di $F(u, v)$. Analogamente tutti i valori di $F(u, v)$ contribuiscono durante l'anti-trasformata a ciascun valore della $f(x, y)$. I campioni della $F(u, v)$ sono complessi, per cui ciascuno di essi avrà un modulo e una fase.

Le proprietà della DFT 2-D

La DFT gode di 3 proprietà:

- *Separabilità*: la trasformata di Fourier discreta può essere espressa in forma *separabile*, in particolare vale la seguente espressione (cioè per il 2D è come se venissero applicati 2 passi del 1D):

$$F(u, v) = \frac{1}{M} \sum_{x=0}^{M-1} g(x, v) e^{-\frac{i2\pi ux}{M}} \quad \text{dove } g(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-\frac{i2\pi vy}{N}}$$
- *Traslazione*: nel caso bidimensionale è utile prima di operare sulla trasformata applicare uno shift (traslazione) dell'origine nel punto $P(\frac{M}{2}, \frac{N}{2})$, cioè nel centro della matrice dei coefficienti delle frequenze. In questo modo i dati vengono traslati in maniera tale che $F(0,0)$ risulti il centro del rettangolo delle frequenze definito tra [0, M-1] e [0, N-1]. Uno shift nella $f(x, y)$ non modifica la magnitudo della trasformata. Queste proprietà vengono utilizzate per una migliore visualizzazione dello spettro;
 - È opportuno notare che la trasformata non è sensibile alla traslazione bensì alla rotazione;
- *Valore medio*: il valore della trasformata nell'origine, cioè nel punto $P(0,0)$ è dato da:

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

Il valore della trasformata di Fourier di un'immagine $f(x)$ nell'origine è uguale alla media dei valori di grigio contenuti nell'immagine. $F(0,0)$ prende anche il nome di *componente continua* o *componente DC*.

La Fast Fourier Transform

Nella sua forma classica implementare la trasformata di Fourier richiederebbe un numero di operazioni proporzionali a N^2 (N moltiplicazioni complesse e $N - 1$ addizioni per ciascuno degli N valori di u).

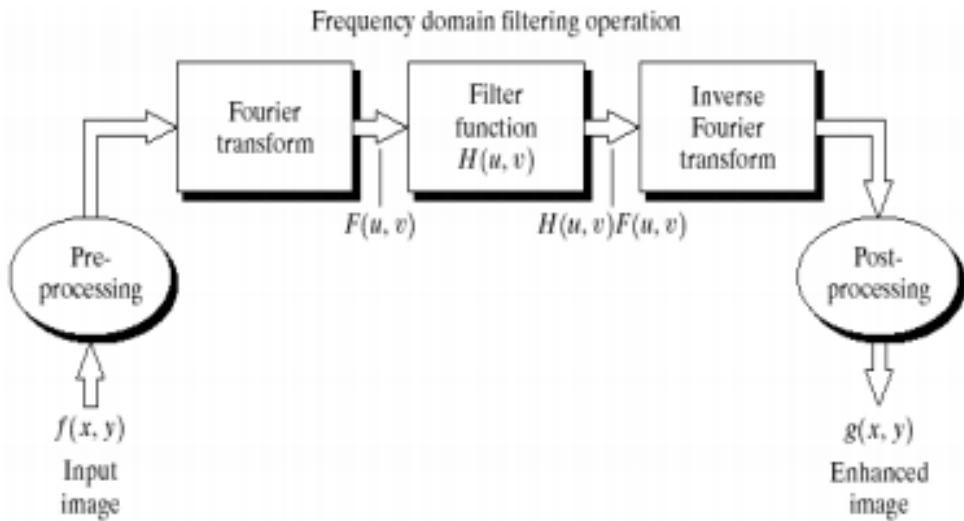
Utilizzando un algoritmo costituito da opportune tecniche di decomposizione è possibile abbassare la complessità a $N \log_2 N$, implementando così la cosiddetta *Fast Fourier Transform (FFT)*.

Filtraggio nel dominio delle frequenze

Esclusi casi banali è normalmente impossibile fare associazioni dirette fra specifiche parti dell'immagine e la sua trasformata a causa di un'ovvia perdita di localizzazione spaziale.

Dato che la frequenza è legata alla velocità di variazione è però possibile associare le basse frequenze alle zone uniformi dell'immagine, mentre quelle alte alle variazioni più o meno brusche e quindi ai bordi o al rumore.

Anche le eventuali presenze di strutture orientate in maniera regolare e netta vengono facilmente individuate da un'analisi visuale dello spettro.



La funzione $H(u, v)$ prende il nome di filtro poiché agisce su alcune frequenze della trasformata lasciando le altre immutate. Molto spesso tale funzione è una funzione reale e ciascuna sua componente moltiplica la sua corrispondente componente reale che quella immaginaria della F . I filtri così definiti prendono il nome di *zero phase shift* poiché non introducono sfasamento.

Il teorema della convoluzione

Il *teorema della convoluzione* afferma che la convoluzione di due segnali nel dominio spaziale equivale all'anti-trasformata del prodotto delle trasformate dei due segnali:

$$g(x, y) = f(x, y) \# h(x, y) = F^{-1} (F(f(x, y)) * F(h(x, y)))$$

Ciò è possibile in quanto il teorema fa corrispondere all'operazione così definita nel dominio spaziale:

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

L'operazione nel dominio delle frequenze:

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

Studiando le complessità si ha che:

- Nel dominio delle frequenze la convoluzione ha una complessità pari a $O(n \log n)$
- Nel dominio spaziale la convoluzione ha una complessità pari a $O(n^2)$

È opportuno notare che se il filtro ha dimensioni confrontabili con quelle dell'immagine è più efficiente effettuare il filtraggio nel dominio delle frequenze, se il filtro ha, invece, dimensioni inferiori a quelle dell'immagine è più opportuno operare nel dominio spaziale.

Come ottenere un filtro a partire da una maschera parziale

- 1) Il filtro H ha la stessa dimensione dell'immagine;
- 2) H deve avere in alto a sinistro i valori della maschera parziale, nel resto sempre il valore 0;
- 3) Si fa lo shift di H ;
- 4) Si calcola da H la trasformata di Fourier;

Filtri di smoothing o passa-basso (low-pass)

I filtri di *smoothing* sono dei filtri *low-pass (passa-basso)* che eliminano il rumore; ciò corrisponde nel dominio delle frequenze ad attenuare le componenti ad alta frequenza.

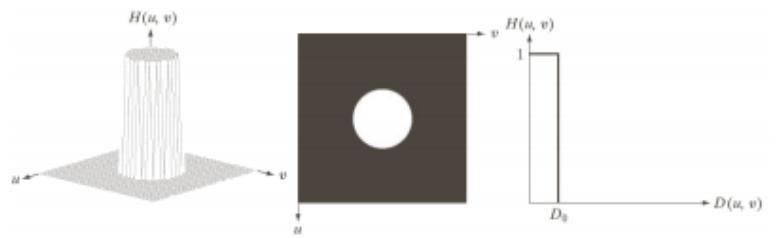
I principali filtri passa-basso sono:

- *Filtri ideali*: tali filtri eliminano totalmente tutte le componenti di frequenza che nel rettangolo delle frequenze distano dall'origine più di D_0 , la cosiddetta *frequenza di taglio*:

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{altrimenti} \end{cases}$$

Dove $D(u, v)$ è la distanza dall'origine ed è data da:

$$D(u, v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$



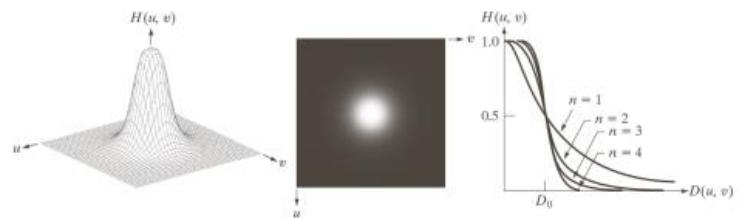
Il termine “ideale” indica che tutte le frequenze all’interno di un cerchio di raggio D_0 sono trasferite senza attenuazione, mentre tutte quelle esterne sono, invece, annullate.

Tali filtri causano un forte fenomeno di sfocatura ad anello detto *ringing*, legato alla funzione $h(x, y)$ dell’anti-trasformata. Gli anelli generati hanno un raggio inversamente proporzionale alla frequenza di taglio.

- *Filtri butterworth*: la funzione di trasferimento di tale filtro di ordine n e frequenza di taglio D_0 è:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

A differenza dei filtri passa-basso ideali, il filtro low-pass di butterworth non determina un taglio netto in frequenza in quanto la sua funzione di



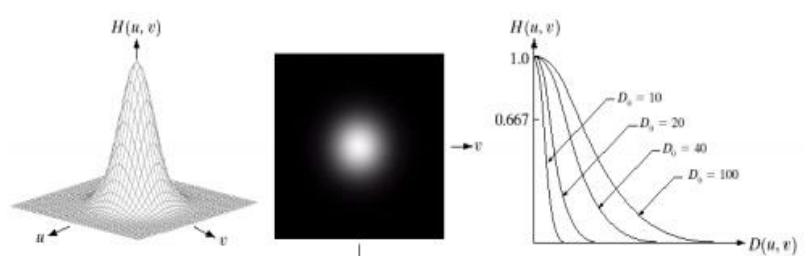
trasferimento non introduce una discontinuità brusca, possiamo dire quindi che viene effettuata una attenuazione graduale.

Al crescere dell’ordine n del filtro l’attenuazione diventa sempre più netta e il filtro sempre più simile a quello ideale, cominciando ad assumere le stesse caratteristiche e gli stessi difetti;

- *Filtri gaussiani*: la funzione di trasferimento di tale filtro di frequenza di taglio D_0 è:

$$H(u, v) = e^{-\frac{D^2(u, v)}{2D_0^2}}$$

Il principale vantaggio di questi filtri è avere come trasformata di Fourier ancora una gaussiana. Presentano caratteristiche simili ai filtri di Butterworth di piccolo ordine, ma con una più semplice realizzazione.



I principali impieghi di un filtro passa-basso sono:

- Il riconoscimento di caratteri;
- La riduzione del rumore;
- L’elaborazione di immagini aeree e satellitari;

Filtri di sharpening o passa-alto (high-pass)

Poiché i bordi e gli altri rapidi cambiamenti nei livelli di grigio sono legati alle componenti ad alta frequenza, è possibile attuare lo sharpening dell’immagine attraverso i filtri *high-pass* (*passa-alto*) che eliminano le componenti a bassa frequenza.

I principali filtri passa-alto sono:

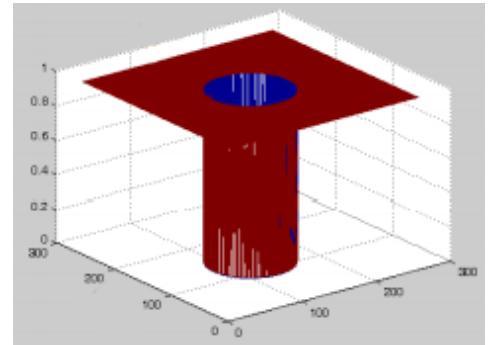
- *Filtri ideali*: tali filtri eliminano totalmente tutte le componenti di frequenza che nel rettangolo delle frequenze distano dall'origine meno di D_0 , la cosiddetta *frequenza di taglio*:

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{altrimenti} \end{cases}$$

Dove $D(u, v)$ è la distanza dall'origine ed è data da:

$$D(v, u) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

Il termine “ideale” indica che tutte le frequenze all'esterno di un cerchio di raggio D_0 sono trasferite senza attenuazione, mentre tutte quelle interne sono, invece, annullate.



Tali filtri causano un forte fenomeno di sfocatura ad anello detto *ringing*, legato alla funzione $h(x, y)$ dell'anti-trasformata. Gli anelli generati hanno un raggio inversamente proporzionale alla frequenza di taglio.

- *Filtri butterworth*: la funzione di trasferimento di tale filtro di ordine n e frequenza di taglio D_0 è:

$$H(u, v) = \frac{1}{1 + \left(\frac{D_0}{D(u, v)}\right)^{2n}}$$

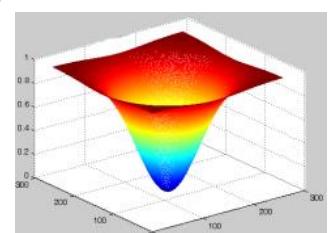
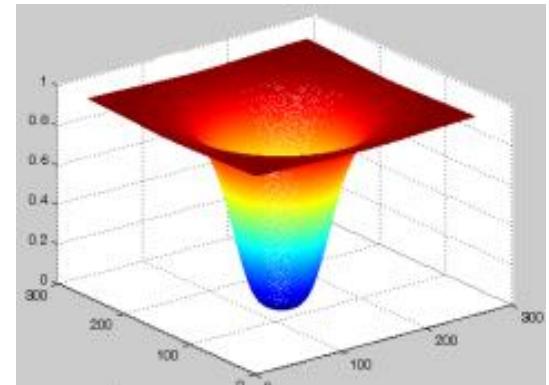
A differenza dei filtri passa-basso ideali, il filtro high-pass di butterworth non determina un taglio netto in frequenza in quanto la sua funzione di trasferimento non introduce una discontinuità brusca, possiamo dire quindi che viene effettuata una attenuazione graduale.

Al crescere dell'ordine n del filtro l'attenuazione diventa sempre più netta e il filtro sempre più simile a quello ideale, cominciando ad assumere le stesse caratteristiche e gli stessi difetti;

- *Filtri gaussiani*: la funzione di trasferimento di tale filtro di frequenza di taglio D_0 è:

$$H(u, v) = 1 - e^{-\frac{-D^2(u, v)}{2D_0^2}}$$

Il principale vantaggio di questi filtri è avere come trasformata di Fourier ancora una gaussiana. Presentano caratteristiche simili ai filtri di Butterworth di piccolo ordine, ma con una più semplice realizzazione. Introduce smoothing.



Filtri di band-reject

Individuato un range di frequenze su cui operare, il filtro *band-reject* costruisce una maschera opportuna per eseguire tale filtraggio.

I principali filtri band-reject sono:

- *Filtri ideali*: tali filtri eliminano totalmente tutte le componenti di frequenza che nel rettangolo delle frequenze non entrano nel range desiderato di centro D_0 , la cosiddetta *frequenza di taglio*:

$$H(u, v) = \begin{cases} 0 & \text{se } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{altrimenti} \end{cases}$$

Dove $D(u, v)$ è la distanza dall'origine ed è data da:

$$D(v, u) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

- *Filtri butterworth*: la funzione di trasferimento di tale filtro di ordine n e frequenza di taglio D_0 è:

$$H(u, v) = \frac{1}{1 + \left(\frac{DW}{D^2 - D_0^2}\right)^{2n}}$$

A differenza dei filtri passa-basso ideali, il filtro high-pass di butterworth non determina un taglio netto in

frequenza in quanto la sua funzione di trasferimento non introduce una discontinuità brusca, possiamo dire quindi che viene effettuata una attenuazione graduale.

Al crescere dell'ordine n del filtro l'attenuazione diventa sempre più netta e il filtro sempre più simile a quello ideale, cominciando ad assumere le stesse caratteristiche e gli stessi difetti;

- *Filtri gaussiani*: la funzione di trasferimento di tale filtro *di frequenza di taglio* D_0 è:

$$H(u, v) = 1 - e^{-\left[\frac{D^2(u, v) - D_0^2}{D \cdot W}\right]^2}$$

Il principale vantaggio di questi filtri è avere come trasformata di Fourier ancora una gaussiana. Presentano caratteristiche simili ai filtri di Butterworth di piccolo ordine, ma con una più semplice realizzazione.

Lezione 11 – Compressione & BitPlanes

Cosa si intende per comprimere?

Con il termine *compressione dati* si indica la *tecnica di elaborazione dei dati* che, attuata a mezzo di opportuni *algoritmi*, permette la riduzione della quantità di bit necessari alla rappresentazione in forma digitale di un'informazione

Riducendo il numero di bit si ha una *riduzione della dimensione del file* ottenendo così anche una *riduzione sia dello spazio necessario alla sua memorizzazione e sia dell'occupazione di banda necessaria in una generica trasmissione digitale*.

I dati ridondanti

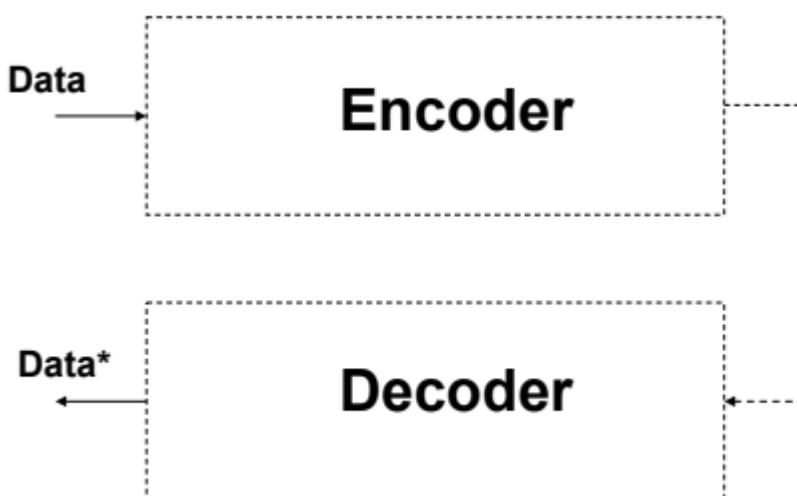
Si definiscono *dati ridondanti* dati irrilevanti (ovvero inutilizzati) o ripetuti. Si definiscono 3 tipologie di ridondanza:

- Ridondanza nella codifica;
- Ridondanza spaziale e temporale: riguardante la correlazione spaziale e/o temporale nelle immagini e video;
- Ridondanza di informazioni percettivamente irrilevanti: riguardante quelle informazioni che vengono ignorate dal sistema sensoriale umano;

Definizione di algoritmo di compressione e classificazione dei metodi di compressione

Si definisce *algoritmo di compressione* una tecnica che elimina la ridondanza di informazione dai dati e consente di risparmiare memoria.

Possiamo quindi generalizzare dicendo che la trasmissione di un dato segue il seguente schema:



Possiamo inoltre classificare i metodi di compressione nel seguente modo:

- Basati sul *tipo di dati*: generico, audio, immagini, video, eccetera;
- Basati sul *tipo di compressione*, che può essere:
 - *Reversibile o lossless*: cioè senza perdita di informazione;
 - *Irreversibile o lossy*: cioè con eventuale perdita di informazione;

Compressione lossless e criterio per buona compressione lossless

Si parla di *compressione lossless* quando i dati possono essere *trasformati* in modo da essere memorizzati con un risparmio di memoria e successivamente ricostruiti perfettamente, senza errori e senza perdita di alcun bit di informazione.

Questo tipo di compressione trova impiego nel ridurre lo spazio occupato da documenti testuali, programmi eseguibili, eccetera.

Il *criterio per una buona compressione di tipo lossless* consiste nel cercare di raggiungere il *limite teorico* per la compressione senza perdita di dati che viene fornito dal *primo teorema di Shannon*.

Tale teorema afferma che *per una sorgente discreta e a memoria zero, il bitrate minimo è pari all'entropia della sorgente*, ovvero il numero minimo di bit necessari a rappresentare dei dati senza perdere informazione è pari a: $N * E$, dove:

- N rappresenta il numero di caratteri;
- E è l'*entropia*, definita come *la quantità media di informazione associata alla singola generazione di un simbolo di una sequenza di dati S*:

$$E = - \sum_{i \in S} f_i * \log_2(f_i) \text{ dove } f_i \text{ (frequenza di un carattere)} = \frac{\text{occorenze}_i}{N}$$

- Più è grande l'*incertezza maggiore* è l'*entropia*, il valore massimo di entropia si ha quando i simboli della sequenza sono equiprobabili;

Compressione lossless: Huffman

L'algoritmo per la codifica di Huffman è un *algoritmo greedy* che permette di ottenere un *dizionario* carattere-codifica per una compressione lossless quasi ottimale dei dati, cioè pari al limite di Shannon con un eccesso di al più qualche bit.

Questa codifica ha due proprietà caratteristiche:

- Si tratta di una codifica a lunghezza variabile che associa ai simboli meno frequenti codici più lunghi mentre ai simboli più frequenti codici più corti;
- Si tratta di una codifica in cui nessun codice è prefisso di altri codici.



È necessario considerare un *costo aggiuntivo* dato che si deve memorizzare il dizionario e se i caratteri sono tanti ciò può risultare costoso.

Compressione lossless: Run-length

Il metodo di compressione *run-length* risulta efficiente solo in presenza di sequenze di valori ripetuti e consiste nel comprimere tali sequenze, dette *run*, sottoforma di coppie *run-length*, in cui ciascuna coppia individua l'inizio di una nuova sequenza di un preciso valore e la lunghezza della sequenza.

La codifica run-length è molto usata nelle immagini, le quali possono avere ripetizioni di intensità lunga le righe o colonne: è attualmente impiegato in JPEG, BMP, M-JPEG, eccetera.

Si voglia codificare "00000111001011101110101111111":

0000011100101110111010111111 → (0,5)(1,3)(0,2)(1,1)(0,1)(1,3)(0,1)(1,3))(0,1)(11)(0,1)(1,7)

Ora bisognerebbe codificare le coppie attraverso un'opportuna tabella di codifica. Questa.

Compressione lossless: Differenziale

Il metodo di compressione *differenziale* risulta efficiente solo in presenza di sequenze i cui valori variano lentamente e consiste nel comprimere tali sequenze ricordandosi solo il valore iniziale e le differenze successive.

Si voglia codificare “134, 137, 135, 128, 130, 134, 112”, ricorderò solo il 134 e le differenze:

$$-3 (134 - 137); +2 (137 - 135) \dots$$

Il concetto base è che le differenze di valori vicini risultano più piccole, e quindi si risparmia spazio salvando queste invece che i valori effettivi.

Compressione lossy e criterio per buona compressione lossy

Si parla di *compressione lossy* quando i dati possono essere *trasformati* in modo da essere memorizzati con un risparmio di memoria ma con una perdita irreversibile di informazione.

Il *criterio per una buona compressione di tipo lossy* ha due modalità:

- Fissata la massima distorsione accettabile, l'algoritmo di compressione deve trovare la rappresentazione con il più basso numero di bit;
- Fissato il massimo numero di bit accettabile, bisogna trovare il miglior algoritmo di compressione che a parità di numero di bit fornisce la minima distorsione.

Generalmente l'idea principale di una compressione lossy per le immagini è: se non è percettivamente rilevante, buttalo via.

Compressione lossy: Requantization

Il metodo di compressione *requantization* è alquanto semplice in quanto si tratta di una riduzione del numero di livelli in modo da risparmiare bit per pixel: vengono scartati *n pixel meno significativi* per canale.



Compressione lossy: JPEG (Joint Photographic Experts Group)

La codifica *JPEG* è abbastanza complessa poiché viene eseguita in 3 passi, i quali costituiti da sotto-passi:

1) Il *pre-procesing*:

- Trasformazione dei colori*: si tratta di passare dallo spazio di colore *RGB* a quello YC_bC_r , questo è necessario per il passaggio successivo.
- Sotto-campionamento della crominanza*: dato che l'occhio umano è più sensibile alla luminanza che alla crominanza si decidere di mantenere tutti i valori della luminanza e prendere, invece, 1 valore ogni 4 per il C_b e C_r . Questo passaggio comporta una perdita delle informazioni ed è perciò irreversibile.
- Partizione dell'immagine*: per approfittare al meglio della ridondanza (che è localmente maggiore che sull'intera immagine) e per semplificare i calcoli, JPEG procede dividendo l'immagine in quadranti *8x8 da 64 pixel* non sovrapposti;
 - Quadranti diversi subiranno una elaborazione differente e indipendente dagli altri, da qui nasce la problematica della *quadrettatura* caratteristica delle immagini compresse in JPEG.

2) Trasformazione:

- a. Discrete Cosine Trasform: viene applicata la DCT a ogni quadrante *8x8 da 64 pixel* al fine di decorrelare al massimo di dati permettendo così maggiori rapporti di compressione nella fase di codifica;
- Prima della applicazione della DCT ai 64 pixel di ciascun blocco viene sottratta una quantità pari a 2^{n-1} , dove 2^n rappresenta il numero massimo di livelli di grigio dell'immagine. Tale processo prende il nome di *shift dei livelli di grigio*. In un'immagine a 8 bit il valore di grigio 128 diventerebbe 0;
 - Ogni immagine è la somma pesata di 64 immagini impulsive, tutte nere tranne per un pixel di valore 1, dove i *pesi* rappresentano l'effettivo livello di luminosità di ogni pixel;
 - Tali immagini impulsive rappresentano una base, detta "*base impulsiva*", che non è unica;
 - La trasformata del coseno esprime l'immagine in un'altra base.*
 - I coefficienti* risultanti dalla DCT andranno moltiplicati alle basi DCT per riottenere il blocco di partenza;
 - Il *coefficiente DC* è il primo valore in alto a sinistra;
 - Tutti gli altri 63 valori sono detti *coefficienti AC*;

La trasformata e l'anti-trasformata sono:

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

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

con $C(h) = \begin{cases} \frac{1}{\sqrt{2}} & \text{se } h = 0 \\ 1 & \text{altrimenti} \end{cases}$

- b. Quantizzazione: un ulteriore passo lossy che consiste nel ridurre i livelli. Si ottiene col formalismo
- $$F_{quantizzato} = \text{round}\left(\frac{F}{Q}\right)$$

3) Codifica:

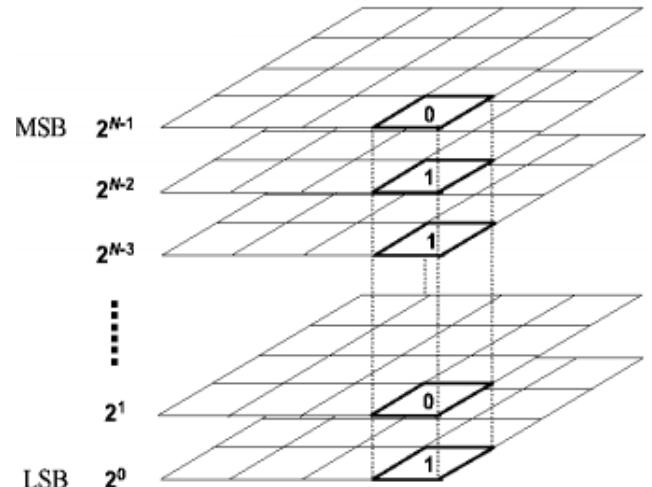
- Codifica dei coefficienti DC: tale codifica avviene col metodo differenziale;
- Codifica dei coefficienti AC: viene creato un vettore di 64 e riempito seguendo uno schema a serpentina detto zig-zag al fine di ottenere una lunga sequenza finale di zeri;
- Entropy Coding: si tratta di applicare huffman sulla sequenza ottenuta;

I bit-plane

Un'immagine con una *profondità di colore* di N bit può essere rappresentata da N *piani di bit*, ognuno dei quali può essere visto come una singola immagine binaria. In particolare si può indurre un ordine che va dal *Most Significant Bit (MSB)*, il bit più significativo, fino al *Less Significant Bit (LSB)*, il bit meno significativo.

Quindi, si definisce *bit-plane* di un'*immagine digitale a N bit* l'insieme di N immagini binarie, o piani, in cui l'immagine i -esima è costituita dai valori dell' i -esimo bit della codifica scelta.

Verranno studiate due codifiche: *binario puro* e *gray code*.



Codifica binario puro, caratteristiche e problematiche

La codifica *binario puro* consiste nel rappresentare i valori dei pixel nella loro normale codifica binaria. Questa semplice codifica gode delle seguenti caratteristiche:

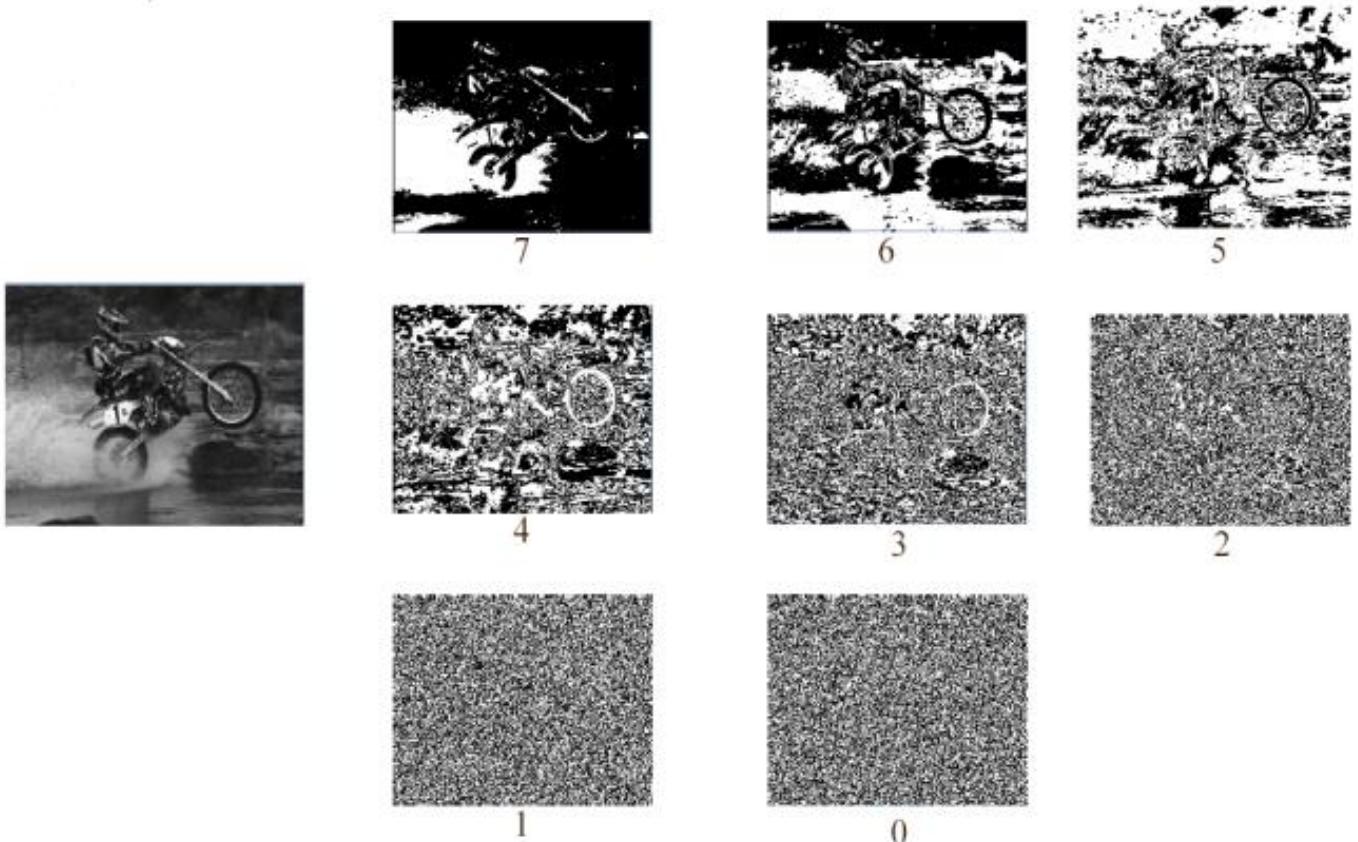
- I piani di bit più significativi contengono maggiori informazioni sulle geometrie dell'immagine, infatti dal punto di vista visuale tendono ad avvicinarsi all'immagine originale;

- I piani di bit meno significativi forniscono, invece, dettagli sempre più piccoli e di minore importanza, in tali piano sono più evidenti il rumore dell'immagine ed eventuali errori di acquisizione;

Queste caratteristiche ci permettono di:

- Attenuare il rumore in un'immagine eliminando i piani meno significativi. Per eliminazione si intende l'azzeramento del piano.
- Rimuovere specifici valori compresi fra un certo range:
 - Azzerando uno specifico piano x si elimineranno tutti i valori nel range $2^{x+1} > \text{valori} \geq 2^x$

Esempio di bit-plane di un'immagine a 8bit:



La codifica *binario puro* soffre di una problematica per cui *una piccola variazione può ripercuotersi su tutti i piani*: per esempio nel passaggio da 127 (0111 1111) a 128 (1000 0000) la variazione di 0 e 1 si ripercuote su tutti gli 8 piani.

Codifica Gray (Gray code) e confronto con il binario puro

La codifica *gray* a m bit $g_{m-1} \dots g_1 g_0$ di un numero in binario puro $a_{m-1} \dots a_1 a_0$ può essere calcolato con la formula:

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

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

Per codificare un i -esimo bit, eccetto per il più significativo che resta invariato, si effettua uno *XOR* fra l' i -esimo bit e il successivo bit.

Il codice Gray gode della seguente proprietà:

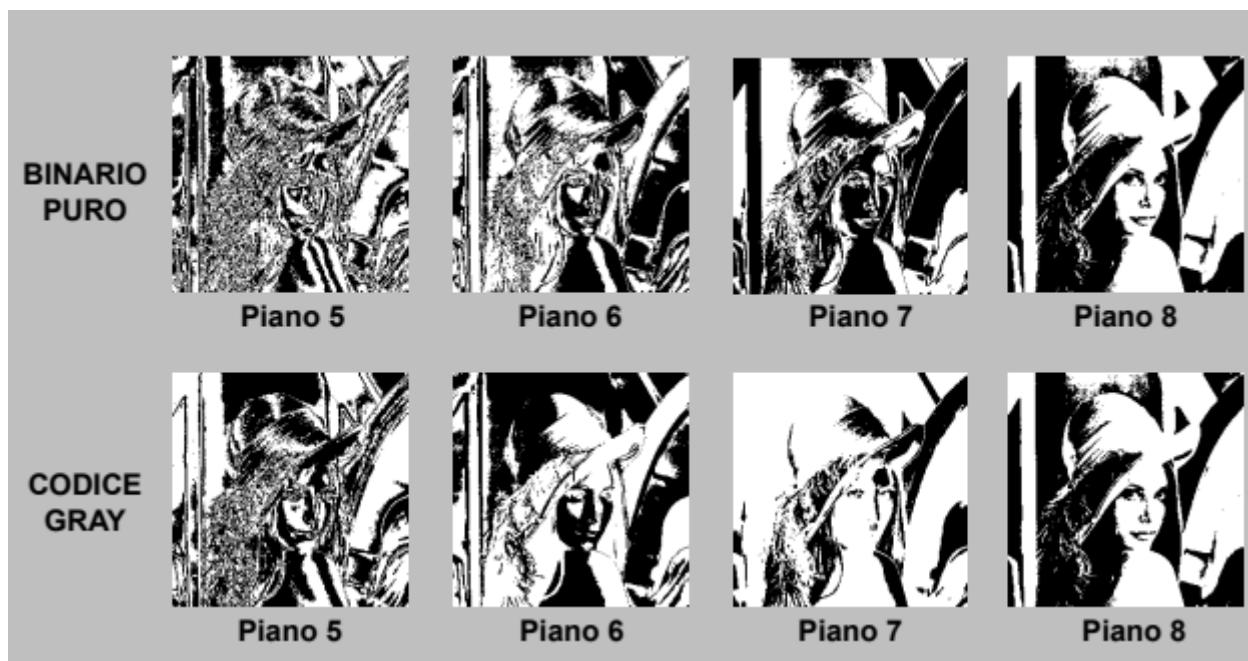
- La codifica di un dato numero differisce da quello del numero successivo per un solo bit;*

I bit-plane di un'immagine codificata in gray code *differiscono* da quelli della stessa immagine codificata in binario puro:

- I bit-plane delle immagini in codice Gray risultano fra loro più *coerenti* se confrontati con i rispettivi in binario puro, questo perché all'aumentare dell'intensità del pixel di 1 varierà sempre solo un piano (ciò grazie alla proprietà di cui gode il gray code);

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- Il numero di *transizioni bianco-nero* nel singolo piano risulta inferiori se si usa la codifica Gray;



Queste differenze comportano una *minore entropia*, ovvero una *maggior ridondanza*, che favoriscono l'impiego delle immagini codificate in gray per la compressione.

A causa del *diverso significato attribuito ai bit a seconda della codifica* alcune proprietà del binario puro non valgono per la codifica gray e viceversa:

- L'*azzeramento di un piano di bit* in Gray code comporta l'eliminazione di un *range di valori diversi* rispetto a quello del binario puro;
- Nonostante i dettagli minori e il rumore dell'immagine continuino a concentrarsi nei piani con bit meno significativi, al contrario del binario puro l'eliminazione di questi piano potrebbe introdurre artefatti;