SLIDE 2

Leggi della percezione visiva: Legge della vicinanza, legge della chiusura, legge della somiglianza, legge della continuità, legge della buona forma

Inpainting (vedi le line dietro le cose)

contrasto simultaneo tra colori (se ho due colori vicini, avrò l'illusione di vedere delle sfumature di transizione)

Un'immagine è rappresentata da una funzione che è la luce incidente ad xy per la luce riflessa (i 0/+inf) (r 0/1)

Immagini vettoriali formule matematiche e leggere, per disegno tecnico, **raster** quelle che usiamo

Pixel ovvero PIcture ELement, le rgb sono 3 matrici in scala di grigio sovrapposte (non confondersi con le immagini a falsi toni).

SLIDE 3

I 4 vicini sono su giù dx e sx, gli 8 vicini è **l'intorno** di un pixel

Forward mapping (v,w) in /(x,y) out/ T matrice affine **(x,y,1)=(v,w,1)*T** si fa scorrere l'input e si crea l'out

Il forward mapping lascia buchi nell'immagine

Inverse mapping, si parte dalla matrice finale e con la formula inversa si calcola il pixel **(v, w, 1) =(x,y,1)*** T^{-1}

Si possono combinare le trasformazioni combinando le matrici

Per lo **zooming** Ci potrebbero essere pixel comunque mancanti, si usa **l'interpolazione** quindi stima i dati

REPLICATION, assegna ad ogni pixel il valore di quello vicino (immagine pixelosa)

BILINEARE, si usano i pixel più vicini per calcolare l'intensità di x,y e abbiamo **v(x,y)=ax+by+cxy+d** i coefficienti si ottengono con 4 eq un 4 incognite dei pixel vicini, l'immagine sfoca

Bicubico, usa i sedici pixel più vicini, sfoca di meno ed è il migliore ma più pesante

Di solito o si tagliano i bordi o si interpola con meno valori

Lo **zooming out** viene detto **decimazione**

Metodo 1, ogni 4 pixel se ne sceglie uno, se dimezziamo le 2 dimensioni dell'immagine, quindi **lunghezza/2** e **larghezza/2** danno un quarto dell'immagine

Metodo 2, dei 4 pixel se ne fa la **media**

Qualità degli algoritmi di interpolazione:

MSE (mean square error) più è basso meglio è, stima l'errore.

PSNR (peak signal to noiseRatio) misura la qualità di un'immagine compressa, più è alto meglio è per calcolarlo serve l'immagine da valutare e una ideale (più alto meglio è).

$$PSNR = 10\log_{10}\left(\frac{S^2}{MSE}\right) \quad MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I'(x,y) - I(x,y)\right]^2$$

S = il massimo valore che può assumere un pixel (solitamente 255).

M, N = dimensione delle matrici

I = matrice ideale

I' = matrice su cui calcolare il PSNR

<u>SLIDE 4</u>

Quando la luce colpisce un oggetto parte è assorbita e parte è riflessa, noi con il sensore catturiamo l'immagine riflessa

CCD (sensore) (charged coupled device) se colpito da fotone assume una carica positiva (ha un tetto massimo altrimenti avremo delle immagini sovraesposte).

Dopo che le cariche sono acquisite da una matrice vengono **shiftate** verso un bus colonna per colonna e poi salvate in memoria

CFA (color filter array) ogni cella memorizza un colore e non una terna, si farà **l'interpolazione** per quelli mancanti

Bayer Pattern (migliore disposizione dei sensori per fotocamere) verdi sulle diagonali e un pixel rosso e uno blu per ogni griglia 2x2, un'immagine in Bayer pattern è in formato **raw** (spesso proprietario). Ovviamente il verde

Dall'immagine scattata vanno interpolati i pixel mancanti con il **demosaicking**, possiamo usare la **nearest-neighbor interpolation** detta anche **replication**

Color interpolation **Bilinear**, in red non si fa nulla, in blue e green si prende il valore dai 4 vicini, questo quando siamo sul pixel red, quando siamo sul pixel verde in red e blu prendiamo i 2 valori più vicini, quando siamo su un pixel blu su red e green prendiamo il valore dai 4 pixel più vicini.

<u>SLIDE 5</u>

Si dice **risoluzione** il numero di pixel per unità di misura spesso si misura in **Mpxl** (megapixel) o **Dpi** (dot per inch).

La risoluzione va contestualizzata come quella di apparecchiatura di ripresa o di resa.

Immagini con una determinata risoluzione andrebbero visualizzati con la stessa risoluzione per una resa ottima

<u>SLIDE 6</u>

L'occhio ci interessa molto per la **retina** che è formata da **coni** e **bastoncelli**

I coni sono fortemente sensibili al colore (sono quelli grandi)

I bastoncelli sono molto sensibili alla **luce** e poco al colore (sono superiori in numero rispetto ai coni)

La **fovea** è la zona di maggior popolazione di coni e bastoncelli

Il **pinhole** è il principio per la quale da un buco possiamo catturare un'immagine su una "lastra" in maniera ideale non accade perché possiamo avere più fonti luminose

Una **lente sottile** è definita come 2 parti speculari

1) raggi paralleli all'asse della lente sottile vengono concentrati in un unico punto detto **fuoco**, posto a distanza f dalla lente

2) i raggi che si dipartono dal fuoco vengono ritrasmessi tutti paralleli all'asse della lente

3) una lente ha 2 fuochi equidistanti da essa

Equazione della lente sottile:

$$1/u+1v = 1/f$$

-u distanza oggetto dalla lente

-v immagine dell'oggetto tramite la lente

La relazione **f = um/m+1** è utile se si vuole fissare un fuoco per garantire una **magnificazione fissata**

Se faccio una foto con oggetto distante u, e sensore distante v se volessi fissare un fuoco per poter avere lo stesso effetto su un oggetto distante u' allora calcolo **m** (**fattore di magnificazione**) = **v/u** e dopo

F = u' *(v/u) /(v/u) + 1=…

Focale piccola=grande campo;

focale grande =piccolo campo se riduci l'area di sensori riduci il campo

<u>Possiamo combinare più lenti</u>

<u>SLIDE 7</u>

Per discretizzare un segnale continuo dobbiamo **campionarlo**

Un campionamento troppo basso fa perdere i dettagli dell'informazione, inoltre potrebbe far apparire nell'immagine dettagli non presenti nell'originale (**aliasing**)

Per scegliere il giusto valore di campionamento si ricorre al **teorema di Shannon**, tale teorema si basa sulla misura della **frequenza di Nyquist**.

Si definisce **Nyquist rate** la più alta frequenza in un segnale continuo e limitato.

Nella pratica la frequenza di Nyquist si calcola suddividendo in intervalli sempre più piccoli il segnale fino a quando ogni intervallo è costante. Il numero di intervalli **N** si dice frequenza di Nyquist.

Teorema di Shannon: Se si raccolgono campioni con frequenza più del doppio della frequenza di Nyquist (2N nel nostro caso) il segnale può essere ricostruito **fedelmente** in ogni suo punto.

Sia data un'immagine NxN e sia il minimo livello di dettaglio x, la frequenza di Nyquist viene calcolata come N/x, per ottenere un'immagine fedele dopo il campionamento dovremo usare 2N+1 campioni, i pixel saranno ottenuti per interpolazione.

Se **sottocampioniamo** (prendiamo un numero di campioni < 2N+1, perdiamo molti dettagli)

Aliasing: le alte frequenze sono "mascherate" da basse frequenze, provocando artefatti nell'immagine, esso viene introdotto quando si impone che il segnale sia limitato per essere campionato. L'aliasing può essere ridotto applicando una funzione di smussamento sul segnale originario prima del campionamento (**antialiasing**)

Alcuni effetti di aliasing sono il Mòire e il wagon wheel.

Quantizzazione: consiste nell'arrotondare i valori campionati per mantenerli in un centro range.

Procedimento standard: se vogliamo quantizzare in un range reale [a, b] su n livelli, si fissano n+1 numeri in [a, b]:

t0=a < t1 < t2 < … < tn < tn+1=b

Il numero x in [a,b] verrà assegnato al livello di quantizzazione k se risulta: tk<= x < tk+1.

Per rappresentare N livelli in memoria saranno necessari $\log_2(N)$ bit.

La quantizzazione commerciale non è uniforme e logaritmica ciò permette di assegnare più livelli nell' area dei toni scuri e meno livelli nella area dei toni chiari.

Quantizzazione uniforme

-range in ingresso: 0, …, N-1;

-range in uscita: 0, …, K-1, K<=N;

L: livello di ingresso rappresentato da un intero

L': livello in uscita

L' = (L*K)/N

Quantizzazione non uniforme

- range in ingresso: 0, …, N-1;

-range in uscita: 0, …, K-1, K<=N;

L' = f(L, N, K)

La funzione f(L, N, K) definisce lo **schema di riquantizzazione**

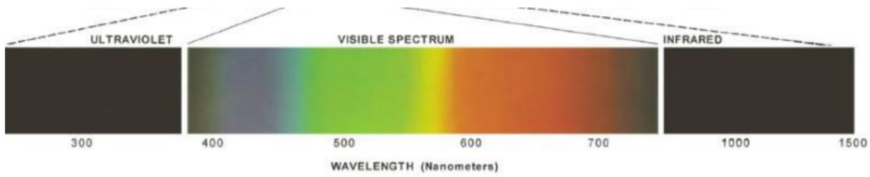
Quantizzazione logaritmica

f(L,N,K) = (log2 (L)*K)/log2 (N)

Nel caso più comune N=256, log(N)=8 e K=8.

SLIDE 8

la luce può essere decomposta in onde luminose di tipo differente, il nostro occhio ne percepisce solo una piccola parte dagli ultravioletti (400 nanometri) agli infrarossi (700 nanometri)



Le bande di colore non sono tutte della stessa grandezza e degradano in quelle limitrofe.

Per descrivere la luce bastano i seguenti valori:

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

L'occhio umano è più sensibile alle tonalità di verde poiché essere ricoprono uno spettro maggiore rispetto alle altre.

Sintesi Additiva: qualsiasi luce di spettro complesso può essere ottenuta come "somma" di tre luci monocromatiche agenti simultaneamente (RGB).

Diagramma cromatico CIE (tristimolo o XYZ) (non percettivamente uniforme)

(X, Y, Z)

-x è la quantità di rosso

-y è la quantità di verde

- z è la quantità di blu ottenuta come **$z = 1 - (x + y)$**

per passare dalle coordinate tridimensionali XYZ alle coordinate xy abbiamo:

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

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

La rappresentazione grafica al variare di x e y da origine al **diagramma cromatico CIE**

Il diagramma cromatico CIE, ha sui bordi tutti i colori puri, Il punto di uguale energia è il bianco (centro) unendo due colori con una linea, tutti i colori nella linea sono quelli ottenibili mischiando i due colori. Unendo un colore con il bianco si ottengono tutti le tonalità di quel colore

Color gamut: unendo R G e B si ottiene un triangolo che contiene tutti i colori che si possono produrre, che, non sono tutti quelli dello spettro

Uniformità percettiva nello spazio di colore

Dati due colori C_1 e C_2, consideriamone le distanze ΔC, rispettivamente, dal colore $C_3 = C_1 + \Delta C$ e dal colore $C_4 = C_2 + \Delta C$. Supponendo che le due distanze siano quantitativamente uguali, in un sistema percettivamente uniforme i due colori C_3 e C_4 sono percepiti come ugualmente distanti da C_1 e C_2.

CIE L*a*b* (percettivamente uniforme

Questo spazio di colore nasce data la mancanza di uniforme percettività del tristimolo (XYZ).

La metrica **CIELAB** si ottiene dalla seguente formula:

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

I parametri che lo determinano sono: L*, a*, b*.

L* rappresenta la luminanza

a* e b* rappresentano la crominanza.

L*, a* e b* sono le trasformazioni dei tre valori di **tristimolo** X, Y e Z dello spazio colore CIE XYZ.

Nello spazio CIELAB le differenze di colore sono definite come distanza tra due punti dello spazio L *a *b *

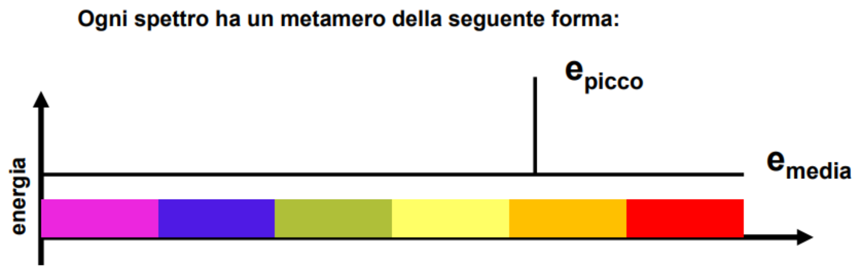
Lo spazio CIELAB è rappresentato da una sfera il cui raggio è lungo |60|.

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

Spettri diversi possono produrre colori eguali: coppie di spettri con questa reciproca proprietà si chiamano **metameri**.

Tra i vari metameri di un dato spettro se ne può sempre individuare uno assai importante che è alla base del modello dei colori detto "del pittore".

Modello del pittore



La lunghezza d'onda in cui si ha il picco è responsabile del "colore percepito" (HUE).

Il rapporto (epicco-emedia) / (epicco+emedia) è la SATURAZIONE, cioè quanto il colore è puro. Meno luce bianca equivale ad un maggiore valore del rapporto. emedia è proporzionale al contenuto energetico della radiazione: essa può essere considerata una misura della "luminosità" di una radiazione.

Spazio HSV

H = hue (colore), copre tutti i colori ordinati in sequenza

S = saturazione, da un minimo (centro) pari al bianco puro ad un massimo (periferia) colore puro.

V = valore o luminosità, da un minimo (nessuna energia emessa) ad un massimo.Lo spazio può essere rappresentato tramite: un cono, un doppio cono, una piramide a base triangolare o una doppia piramide. Tutti i colori hanno la stessa distanza dal bianco e i colori complementari sono opposti (a distanza di 180°).

Il solido dei colori

Munsell è stato il primo a pensare ad una rappresentazione **tridimensionale** dei colori, il solido dei colori è discreto ma non continuo, poiché si rifà alla realtà.

Sintesi additiva

RGB: Composizione additiva dei colori partendo dai tre colori primari RGB. Dalla loro sovrapposizione si ottiene il bianco (W), dalla sovrapposizione di due luci si ottengono giallo (Y), magenta (M) e ciano (C).

Combinandoli tra di loro i tre colori RGB, **NON** si ottengono tutti i colori visibili.

Sintesi sottrattiva

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

Sovrapponendo tutti e tre i filtri viene assorbita tutta la radiazione visibile per cui si ottiene il Nero (K).

I colori sottrattivi sono usati nei sistemi passivi, nei quali la luce viene assorbita selettivamente alle diverse lunghezze d'onda, riflettendone solo alcune che comporranno i colori desiderati.

RGB

Il modello RGB è graficamente descritto come un **cubo**.

I contributi del RED, GREEN e BLUE sono assunti indipendenti l'uno dall'altro (e quindi rappresentanti da direzioni perpendicolari tra loro). Ogni colore è un punto contenuto dentro il cubo. La retta che congiunge nero e bianco è la **retta dei grigi**.

Il modello CMY è il complementare dell'RGB, a volte viene aggiunta una quarta coordinata K (black).

Colori sicuri per il web

I colori sicuri per il web sono **216**, essi sono: (00, 33, 66, 99, CC, FF), tutte le possibili terne di questi numeri creano tutti i colori utilizzabili sul web

Rappresentazioni luminanza-crominanza

Gli spazi colore, nei quali una componente è la **luminosità** e le altre due componenti sono legate alla **crominanza**, vengono chiamate **rappresentazioni luminanza-crominanza**. La luminanza fornisce una versione a scala di grigi dell'immagine mentre la crominanza fornisce le informazioni che trasformano l'immagine in scala di grigi in un'immagine a colori.

Poiché l'occhio umano è più sensibile alla luminanza, conviene "spendere" molti bit per registrare la luminanza e risparmiare sulle crominanze

YUV

Lo spazio YUV viene spesso utilizzato per la codifica di immagini, tenendo separate la luminanza dalla crominanza. Con YUV ci si riferisce NON ad uno specifico spazio, ma ad una famiglia di spazi con le caratteristiche sopraccitate

Da RGB a YUV:

La luminanza si può ottenere come combinazione lineare dei tre canali RGB

$$\mathbf{Y = 0.3}R\mathbf{ + 0.6}G\mathbf{+ 0.1B}$$

Il termine crominanza è definito come la differenza tra il colore e un bianco di riferimento alla stessa luminanza opportunamente pesato:

$$\mathbf{U} = B - Y$$

$$\mathbf{V} = R - Y$$

Quando **R=G=B**, U e V valgono 0 e si ottengono solo grigi (nessuna crominanza)

I tre canali RGB non danno eguale contributo alla luminanza.

Da YUV a YCbCr

Per passare da YUV a YCbCr, si shiftano i valori di un bit, per normalizzare i risultati e inserirli in un range 0,255

Y = 0.3R + 0.6G+ 0.1B C

C_b = U/2 + 114.75

C_r = V/1.6 + 111,57

Questo sistema di coordinate viene largamente utilizzato dagli standard di **compressione** (Es. JPEG).

Colori e memoria

Poiché conservare un’immagine in memoria è molto dispendioso (24 bit per pixel), si utilizzano delle **tabelle indicizzate** (palette, LUT, look up table) dove figurano i colori che hanno maggiore presenza nell’immagine per abbattere i costi, solitamente, le palette hanno dimensione standard di **256** colori. Se nell’immagine true color sono presenti meno di 256 colori essi vengono memorizzati tutti nella palette, altrimenti vengono memorizzati solo i 256 colori più significativi (e presenti).

Re-Indexing

Consiste nel modificare la palette, ordinando i colori, per migliorare la qualità dell’immagine, diminuendo l’**entropia**.

<u>**SLIDE 9**</u>

Istogramma: è la distribuzione delle frequenze dei toni di grigio in un’immagine. In una matrice [MxN], H(k) è il numero di pixel di tonalità k presenti in essa.

Istogrammi di immagini “diverse” possono essere uguali in quanto esso non tiene conto della distribuzione spaziale.

Contrast stretching: Serve a dilatare un istogramma i cui valori sono concentrati solo su un intervallo limitato su tutto il range. Dopo lo stretching l’istogramma avrà la forma di un “pettine” a causa dei bit mancanti.

Normalizzazione: Serve a riportare nel range 0,255 i pixel che soffrono di problemi di range.

Soluzione 1: quando il pixel è sotto lo zero lo settiamo a 0 (nero) e quando è sopra 255 lo settiamo a 255 (bianco).

Soluzione 2: Trasformiamo ciascun valore secondo tale equazione:

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

Min e max sono il valore minimo e quello massimo assunto dall'istogramma. Quest'equazione è usata per "spalmare" tutti i valori sull'istogramma.

Si parla di immagine equalizzata quando il contributo di ogni differente tonalità di grigio è pressappoco eguale. Si parla anche di "istogramma" uniforme o appiattito. (*non sempre un'immagine equalizzata è migliore).*

Algoritmo di equalizzazione

$\mathbf{r_k}$ = un livello di grigio;

$\mathbf{n_k}$ = numero di pixel nell'immagine [NxM] di quel livello di grigio;

M, N = dimensioni dell'immagine.

Possiamo definire la **distribuzione di intensità** per ogni pixel:

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

I nuovi valori di grigio dell'istogramma sono così definiti:

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

L = livello di intensità, dato da $\mathbf{2^{n\_bit\_per\_pixel}}$

L'istogramma di un'immagine equalizzata è composto da valori ordinati (e a gradini).

Operazioni sulle immagini

Le elaborazioni nel dominio spaziale:

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

g(x,y) = immagine in uscita;

f(x,y) = immagine in entrata;

T = operatore su f definito in un intorno di (x,y).

La dimensione dell'intorno di (x,y) definisce il **carattere della elaborazione**:

-**puntuale** (l'intorno coincide con il pixel stesso);

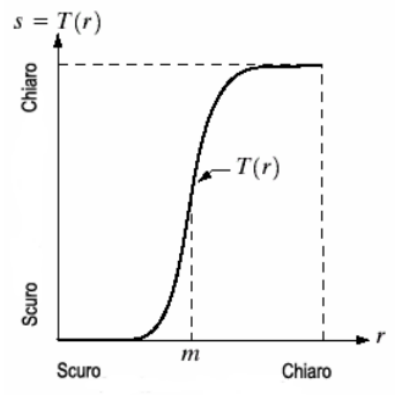
- **locale** (per esempio una piccola regione quadrata centrata sul pixel);

-**globale** (l'intorno coincide con l'intera f).

Operatori puntuali

Si dice 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.

LUT



Questo tipo di grafico si chiama **Look Up Table** (**LUT**).

Negativo: consiste nell'associare al valore f(x,y) del pixel il valore 255-f(x,y);

Incupimento: sposta i valori della LUT da destra verso sinistra;

Schiarimento: sposta i valori della LUT da sinistra verso destra;

Trasformazione logaritmica:

consente di comprimere la gamma dinamica per visualizzare l'immagine in una scala di grigi usuale:

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

c: costante positive per normalizzare il risultato tra 0 e 255;

L'uno nella formula serve ad evitare di calcolare logaritmo di 0.

Trasformazione di potenza:

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

dove **c** e **γ** sono costanti positive. La costante c'è scelta di volta in volta in modo da normalizzare i valori di s nell'intervallo [0, 255]. Come vedremo, per valori di γ minori di 1 la trasformazione ha effetti analoghi alla trasformazione logaritmica (espansione della dinamica per bassi valori di f, compressione della dinamica per alti valori di f), mentre per valori di γ maggiori di 1 la trasformazione ha esattamente gli effetti opposti.

Più la gamma aumenta più l'immagine si incupisce.

Binarizzazione: Produce una immagine che ha solo due livelli: nero e bianco. Si ottiene scegliendo una soglia T e mettendo a nero tutti i pixel il cui valore è minore a T e a bianco tutti gli altri.

Variazioni di contrasto: Aumentare il contrasto, significa rendere più evidenti le differenze di colore, scurendo o schiarendo il pixel.

Solarizzazione: ottenuta da curve non monotone.

SLIDE 10

Dato un vettore di lunghezza N, esso può essere scomposto nella sua base canonica, essendo le immagini matrici, lo stesso ragionamento può essere applicato anche ad esse.

Operatori locali

Sono usati per migliorare la qualità delle immagini o per estrarre delle informazioni dall'immagine. Si possono pensare come filtraggi dell'immagine ottenuti per convoluzione.

Operatori Lineari

Un operatore **F: V -> W**, si dice lineare se accade che per ogni a,b reali:

$$\mathbf{F(a\ v1 + b\ v2) = a\ F(v1\ ) + b\ F(v2)}$$

Ogni elemento della matrice ha un comportamento diverso.

Un operatore si dice invariante per traslazione (shift invariant) 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

Definizioni

-Se F è **lineare** per descriverlo basta conoscere il comportamento su tutte le immagini impulsive;

-Se F è **shift invariant** si comporta allo stesso modo su tutti gli impulsi, indipendentemente dalla loro posizione;

-Se F è sia **lineare** che **shift invariant** per descriverlo basta conoscere come si comporta su un solo impulso. La "risposta all'impulso" o "point spread function" di F è la carta di indentità di tale operatore.

Ad un operatore lineare e shift invariante corrisponde una **maschera** e vale anche il viceversa

La "risposta all'impulso" o PSF definisce completamente un operatore lineare e invariante per traslazioni F. Spesso un operatore su una immagine prende il nome di "filtro". La matrice che descrive la risposta all'impulso si chiama anche **kernel** o **maschera dell'operatore**. Essa è detta anche **maschera di convoluzione di F,** La grandezza del kernel può variare fino ad essere infinita.

Convoluzione: si indica con la notazione: h = f x g, essa è commutativa e associativa.

Quando applichiamo un kernel ad una sottomatrice dell'immagine su cui stiamo operando, il risultato andrà memorizzato nel pixel centrale.

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

Problema dei bordi:

Dopo aver applicato un kernel per convoluzione, i bordi rimarranno non calcolati, possibili soluzioni sono:

-tagliare i bordi;

- Supporre che tutto intorno all'immagine ci sia 0 e applicare il kernel;

- Assumere una topologia toroidale (effetto pacman), non è ottima in caso di immagini non omogenee;

-replicare i bordi della matrice per poter applicare il kernel.

Esempi di operatori locali

Mediano: È un filtro non lineare che fornisce in uscita il valore mediano dell'intorno del pixel. (ordina gli n elementi e restituisce n/2), provoca una sfocatura.

Minimo e massimo: dato un intorno di pixel, seleziona rispettivamente o il più piccolo o il più grande e lo replica in ogni pixel dell'intorno. Il minimo incupisce l'immagine mentre il massimo la schiarisce.

N-box (o di media): Sono definiti da kernel N x N con ogni elemento pari a 1/N2 . Si sceglie generalmente un valore N dispari. Hanno l'effetto di sfocare le immagini.

N-binomiale: Sono filtri di smussamento, detti anche filtri gaussiani, Hanno il pregio di smussare egualmente in tutte le direzioni. Smussano meno vigorosamente degli n-box.

Conservazione dell'energia: si dice che un filtro "conserva l'energia" se la somma dei suoi pesi fa 1. Tutti i filtri di smoothing visti prima sono "energy preserving": la somma dei valori totali della luminanza nella immagine non cambia.

Noise cleaning: I filtri appena visti servono anche a ridurre il rumore in una immagine. In questo caso, più è grande il kernel e migliore sarà il risultato anche se si rischia di aumentare la sfocatura

Il rumore

Rumore impulsivo (sale e pepe): I pixel dell'immagine diventano o bianchi (255) o neri (0). Per questo tipo di rumore il filtro mediano funziona meglio in fase di smoothing.

Rumore gaussiano: questo rumore, causa una variazione del pixel irreversibile, nonostante questo, a differenza del sale e pepe, ci consente di distinguere comunque i dettagli. Il filtro migliore per eliminare questo rumore è il filtro di media.

<u>Se in un'immagine il rumore è troppo diffuso, è preferibile applicare applicare 2 volte un filtro piccolo anziché una volta un filtro più grande.</u>

Media vs mediano: il filtro media tende a creare dei livelli di grigio prima non esistenti e non attenua solo il rumore ma anche tutte le alte frequenze spaziali in maniera indiscriminata dando origine ad immagini sfocate

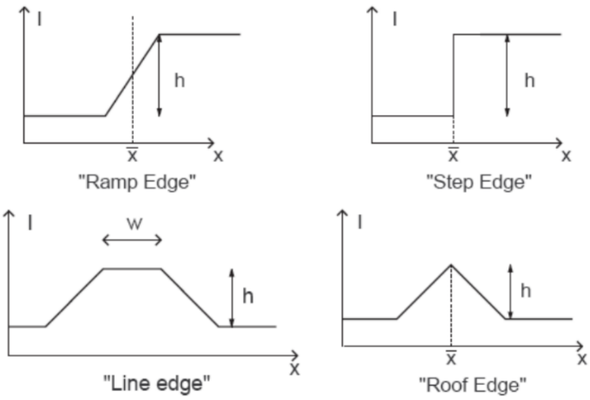
Esistono altri filtri **non lineari** molto importanti:

> **Outlier**: il valore del pixel centrale viene confrontato con il valore della media dei suoi 8 vicini. Se il valore assoluto della differenza è maggiore di una certa soglia, allora il punto viene sostituito dal valore medio, altrimenti non viene modificato.
>
> **Olimpico**: da un dato intorno si scartano i valori massimo e minimo e sul resto si fa la media.

Estrazione di contorni

Gli operatori locali ci aiutano ad estrarre i contorni da una immagine, I contorni sono definiti come delle discontinuità locali della luminanza. Gli **edge detector** forniscono immagini in cui sono preservate le variazioni di luminanza ed eliminate tutte le altre informazioni.

Esempi di lato in 1D



Se ho un segnale monodimensionale e calcolo la **derivata prima**, scopro che i lati sono in corrispondenza dei massimi della derivata.

Kernel notevoli

Lati orizzontali

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

lati verticali

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

Le due matrici possono essere combinate insieme mediante la seguente formula:

sqrt($Sobel_x$^2 + $Sobel_y$^2)

la matrice ottenuta ha valori non nulli per i pixel «di lato». Se si fissa una soglia adeguata, si può ottenere una matrice binaria che per ogni pixel ci dica se è o non è di lato.

Edge detector basati sulla derivata seconda

Il filtro più diffuso per calcolare la derivata seconda è detto **Laplaciano**, ed è definito dalla maschera:

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

Zero-crossing

Dopo aver applicato l'operatore Laplaciano è necessario che si verifichi la condizione di Zero-crossing. Cioè, deve sempre accadere che rispetto al punto in questione ci sia nel suo intorno un valore positivo e un valore negativo.

Filtri di sharpening

Sono filtri il cui scopo è quello di incrementare la nitidezza di una immagine aumentando il contrasto locale. Questa è una operazione opposta allo sfocamento. Per ottenere tale effetto si può adottare una maschera che, derivata dal Laplaciano, "rinforza" i lati presenti nell'immagine.

-1	0	-1
0	5	0
-1	0	-1

Nello sharpening, la somma delle componenti non è mai zero ma sempre positiva.