Appunti di Elaborazione di Immagini e Visione Artificiale

A.A. 2022-2023

Matteo Brunello

Indice

1 Introduzione

1.1 Il sistema visivo umano

Le principali componenti dell'occhio umano sono:

- **Cornea** e **sclera**: la cornea è un tessuto duro e trasparente che copre la superficie anteriore dell'occhio, mentre la sclera è una membrana opaca che racchiude il resto dell'occhio che non è coperto dalla cornea.
- **Coroide**: tessuto ricco di vasi sanguigni per il trasporto di nutrienti alle cellule dell'occhio. L'**iride** è parte della coroide e permette di controllare la quantità di luce che entra nell'occhio per mezzo di contrazioni/espansioni, la sua funzione è quella di mettere a fuoco gli oggetti del mondo reale.
- **Lente**: la lente dell'occhio è un tessuto fibroso da cui passa la luce, questo tessuto assorbe relativamente poca luce visibile e molta luce non visibile.
- **Retina**: membrana che ricopre l'interno dell'occhio su cui viene riflessa l'immagine che si sta vedendo. Essa è costituita da:
  - **Coni**: recettori molto sensibili al colore. Sono localizzati in una zona centrale della retina detta *fovea*. C'è ne sono tra i 6 e 7 milioni. La vista di questi recettori è detta *fotopica*.
  - **Bastoncelli**: molto più numerosi dei coni, tra i 75 e 150 milioni. Molto sensibili all'intensità luminosa, servono a creare una visione generale dell'immagine (utile in condizioni di scarsa luminosità), detta anche visione *scotopica*.

Il numero di bastoncelli e coni decresce a 0 a circa 20 gradi di distanza (a sinistra) della fovea, chiamato *blind spot* cioè il punto in cui il nervo ottico è connesso alla retina. I coni sono più densi al centro della retina, decrescendo via via che ci si sposta (in gradi) verso destra e sinistra. D'altra parte, il punto con meno bastoncelli è la fovea, mentre crescono esponenzialmente per raggiungere il massimo a circa 20 gradi e per poi decrescere mano a mano. Tendenzialmente ciò riprende il fatto che i bastoncelli sono più presenti in numero per garantire una migliore visione periferica, mentre i coni per garantire una fedeltà migliore dell'immagine che si sta direttamente guardando.

La concentrazione della sensibilità dei colori dei coni è la seguente:

- 65% sensibili al giallo-arancione.
- 33% sensibili al verde.
- 2% sensibili al blu.

Il colore percepito è la somma di questi colori primari. Inoltre, la concentrazione di coni ripartita in questo modo non permette di percepire tutti i colori nello spettro della luce visibile.

1.2 Fondamenti di immagini digitali

Un'immagine bidimensionale è una rappresentazione nel piano di una sena o di un fenomeno fisico che ha luogo nel mondo reale. Formalmente possiamo descriverla come una funzione $f(x, y)$ il cui valore è una quantità scalare positiva il cui significato fisico dipende dalla sorgente dell'immagine. La funzione $f(x, y)$ può essere caratterizzata dal prodotto di due componenti: la quantità di *illuminazione* $i(x, y)$ della scena vista e la quantità di *riflettenza* $r(x, y)$ degli oggetti visti nella scena. Ovviamente, $i(x, y)$ è determinata dalla sorgente di illuminazione, mentre $r(x, y)$ è determinata dalle caratteristiche degli elementi che sono presenti nella scena. Possiamo definire diverse tipologie di immagini:

- **Visibili**: possono essere percepite da un sistema visivo umano oppure ottico.
- **Non direttamente visibili**: create da distribuzioni bi-dimensionali di quantità fisiche quali ad esempio temperatura, pressione o campo elettrico.
- **Definite Analiticamente**: definite da funzioni continue a due variabili oppure da funzioni discrete a due indici.

Le immagini possono essere ulteriormente categorizzate in immagini:

- **Analogiche**: caratterizzate dall'evoluzione *continua* dei valori x, y nello spazio, e dei valori di $f(x, y)$ che definiscono il colore/intesità luminosa.
- **Digitali**: sono ottenute discretizzando i valori di f e dei valori di x e y nello spazio. Il *campionamento* (sampling), è il processo di discretizzazione spaziale (dei valori x e y) e la *quantizzazione* (quantization), è la digitalizzazione del valore di f all'intero più vicino.

Le immagini salvate come una serie di linee (o righe) di pixels sono dette immagini *raster* (aka *bitmap format*). L'acquisizione di un immagine digitale prevede un processo di campionamento della grandezza fisica di riferimento (in quanto grandezza fisica continua). Questo processo può essere visto concettualmente come la superimposizione di una griglia bi-dimensionale. La grandezza delle singole celle determina direttamente la qualità dell'immagine. La scelta della grandezza delle celle determina la qualità dell'immagine in un certo senso. Più esse sono grosse, più si accentua il fenomeno in cui le immagini diventano *pixelate*. Questa grandezza è determinata direttamente dal sensore di acquisizione.

Per rappresentare un'immagine digitale, è necessario di fatto salvare in una matrice i valori (quantizzati) di $f(x, y)$. Il numero di bit che si scelgono per rappresentare questi valori determina il *color depth*, rappresentato in *bit per*

pixel (bpp). In caso $Im(f) = \{0, 1, \ldots, L-1\}$, saranno necessari $log_2(L)$ bpp per rappresentarne i valori. Se abbiamo immagini catturate in *grayscale*, di solito si utilizza 1 byte (8 bpp). Molto comuni invece sono immagini catturate in RGB (chiamati anche *additive primaries*, in caso siano in CMY sono chiamate *subtractive primaries*), in questo caso si utilizza un byte per canale, per cui ogni pixel richiederà 24 bit. Il numero totale di colori rappresentabile con 24 bit è (2^8 per ogni componente spettrale) $2^8 \cdot 2^8 \cdot 2^8 = 2^{24}$ colori possibili (circa 16.000.000).

> In generale, per calcolare il numero di bit per pixel, facciamo il logaritmo in base 2 del numero *massimo* di valori rappresentabile.

Siccome le imagini non sono altro che la rappresentazione di una grandezza fisica, vien da sè che è possibile fare diverse operazioni su di esse. Prime tra tutte sono le operazioni logiche quali and o or. Altre operazioni sono la somma, differenza, moltiplicazione e divisione. Il risultato è definito applicando l'operazione $(\cdot)$ nel modo seguente $h(x, y) = f(x, y) \cdot g(x, y)$.

Un caso di applicazione di queste operazioni è ad esempio la riduzione del rumore per mezzo della media (si veda esempio sulle slides). Altre tipologie di operazioni sono le **trasformazioni affini**, che sono in grado di mantenere line e parallelismi. Appartententi a questa famiglia di trasformazioni sono le traslazioni, rotazioni, ridimensionamenti e ritagli.

Altro concetto molto importante è la **risoluzione** di un'immagine, che è in funzione della *grandezza* dell'immagine ($W \times H$) e della sua dimensione *spaziale*. La risoluzione di un'immagine viene tipicamente espressa in *dots per inch* (dpi). In generale, per calcolare la risoluzione si fa $resolution = \frac{size}{spatialdimension}$.

Come citato in precedenza, il campionamento del segnale analogico periodico (cioè che cambia nel tempo) determina in modo diretto quanto questo segnale viene rappresentato fedelmente a livello digitale. Tanto più frequentemente si acquisiscono i campioni di un segnale, tanto più questo sarà rappresentato fedelmente, valendo invece l'inverso se campionato troppo infrequentemente. Il teorema di **Nyquist-Shannon** ci da la certezza teorica che un segnale che contiene frequenze più piccole di f_c Hertz, può essere ricostruito completamente da campioni presi a intervalli non più grandi di $frac12f_c$ secondi.

2 Colore

Nonostante la sensazione del colore percepito sia un fenomeno fisiopsicologico, dal punto di vista formale non è nient'altro che una particolare lunghezza d'onda della luce. Il colore percepito di un oggetto, è il risultato della luce assorbita e riflessa dello stesso. Ad esempio, se percepiamo un oggetto verde, significa che questo rifletterà le lunghezze corrispondenti al verde, assorbendo invece tutte le altre.

Ci sono diverse quantità di base che vengono utilizzate per descrivere le componenti delle luce cromatica:

- **Radianza**: quantità totale di energia *emessa* dalla sorgente luminosa (misurata in Watts).
- **Luminanza**: quantità di energia emessa dalla sorgente *percepita* da un osservatore (misurata in Lumen).
- **Luminosità**: intensità della componente *acromatica*, è soggettiva e non formalmente definita.
- **Lunghezza d'onda dominante** (hue): lunghezza d'onda della componente del colore più forte.
- **Saturazione**: rapporto tra la potenza della lunghezza d'onda dominante e del valore medio dello spettro (luce bianca).
- **Cromaticità** (chroma): Hue + Saturazione.

2.1 Spazi colore

Lo scopo di un modello colore (chiamato anche spazio colore o sistema colore), è principalmente quello di standardizzare la specifica dei colori. Essenzialmente, un modello colori non è nient'altro che una specifica di un sistema di coordinate e un sottospazio all'interno di esso in cui ogni colore viene rappresentato da un singolo punto.

2.1.1 Modello RGB

Il modello più semplice è il modello RGB. Il sistema di coordinate è dato dalle 3 componenti principali del colore: rosso, verde e blu. Il sottospazio in cui sono definiti tutti i colori è il cubo unitario in cui i colori RGB primari sono ai 3 vertici, mentre negli altri 3 vertici ci sono il ciano, magenta e giallo. Nei rimanenti 2 vertici, ci sono il nero (che è situato all'origine) e il bianco (situato invece nel vertice più lontano dall'origine). Sulla diagonale principale (dal nero al bianco) del cubo sono situate tutte le possibili tonalità di grigio.

Implicitamente ogni valore all'interno del cubo è normalizzato nel range $[0, 1]$. Il numero di bits che vengono utilizzati per rappresentare le singole componenti di colore è detto *pixel depth*. Tipicamente, il valore di color depth è circa 8 bit, per cui è possibile rappresentare $2^24 = 16.777.216$ colori diversi con tale modello.

2.1.2 Modello HSI

I modelli HS* sono basati sulle caratteristiche della luce che nell'occhio umano definiscono direttamente un colore. Quando un essere umano vede un colore, lo descrive in termini delle sue caratteristiche di:

- **Hue** (o lunghezza d'onda dominante): definisce direttamente il colore *puro*.
- **Saturazione**: definisce quanto puro è il colore in termini di quanto questo viene *diluito* dalla componente di luce bianca.
- **Luminosità**: definisce quanto è luminoso il colore.

Siccome la luminosità è definita solo soggettivamente, il modello HSI introduce la componente acromatica di *intensità* del colore, che è invece definibile e

direttamente collegata alla sensazione di luminosità. Dal punto di vista formale, questo modello si ottiene essenzialmente *ruotando* il cubo del modello RGB in modo tale da rendere verticale la diagonale. In questo modo, considerando tutti i punti ad una determinata altezza otterremo un piano. Questo piano, detto *piano cromatico*, conterrà quindi tutti i possibili colori all'intensità luminosa specificata dall'altezza. È immediato notare che la saturazione dei colori aumenta in base alla distanza dal centro di questo piano (punto d'intersezione con la retta che rappresenta i valori di intensità). D'altra parte, se considerassimo un piano "verticale" definito ad esempio dai punti bianco, nero e ciano, su questo piano ci saranno tutti i valori di colori con lo stesso valore di hue (ciano in questo caso). Questo perchè i valori all'interno di tale piano verticale sono ottenuti come combinazione lineare dei valori dei vertici.

Più nel dettaglio, dato un punto nello spazio HSI, il valore di saturazione viene dato dalla lunghezza del vettore giacente sul piano perpendicolare all'asse dell'intensità, mentre il valore di hue viene determinato dall'angolo tra tale vettore e l'asse del rosso (0 gradi = rosso, 120 gradi = verde, 240 gradi = blu). Il valore di hue, è quindi compreso tra $[0; 360]$. Questo tipo di modelli funziona particolarmente bene perchè divide la componente cromatica (HS) dalla componente acromatica, luminosa (I). Grazie a questa stretta relazione tra i due modelli di colore possiamo convertire direttamente un valore di RGB in uno HSI e viceversa.

Dal punto di vista formale:

- $I = (r_\gamma + g_\gamma + b_\gamma)/3$
- $S = 1 - min(r_\gamma + g_\gamma + b_\gamma)/I$, altrimenti 0 se $I = 0$

2.1.3 Modello HSV

Il modello HSV (Hue, Saturation, Value) è molto simile al modello HSI. Questo modello si ottiene trasformando gli angoli del cubo in modo tale da essere portati allo stesso livello del bianco ($V = 1$), ottenendo così uno spazio a forma di piramide.

Dal punto di vista formale, siano:

- R: vertice del piano dell'esagono rappresentante il rosso.
- O: l'origine (centro) dell'esagono.
- P: il colore da rappresentare.
- $\gamma = (r_\gamma, g_\gamma, b_\gamma)$

Allora, possiamo calcolare i valori:

- $hue_{360}(\gamma) = 360 \cdot \frac{RP}{6OR}$
- $value(\gamma) = max(r_\gamma, g_\gamma, b_\gamma)$
- $chroma(\gamma) = max(r_\gamma, g_\gamma, b_\gamma) - min(r_\gamma, g_\gamma, b_\gamma)$
- $saturation(\gamma) = \frac{chroma(\gamma)}{value(\gamma)}$ (in realtà questo valore è definito come 0 in caso il denominatore sia 0 per evitare evidenti problemi)

2.1.4 Modello HSL

Anche in questo caso il modello HSL (Hue, Saturation, Lightness) è simile ai precedenti, solamente che i vertici del cubo vengono proiettati sul piano centrale (quello ad altezza $L = 0.5$), ottenendo di fatto una *doppia piramide*. Il vertice superiore della piramide corrisponde ad un valore di luminsity pari a 1 (bianco), mentre quello superiore pari a 0 (nero).

Come già detto, i valori di H e S sono calcolati nello stesso modo come nel modello HSV, mentre il valore di L viene calcolato nel modo seguente

$$lightness(\gamma) = (max(r_\gamma, g_\gamma, b_\gamma) + min(r_\gamma, g_\gamma, b_\gamma))/2$$

2.2 Color Image Processing

Esistono diverse tecniche di processamento delle immagini che riguardano direttamente il colore.

2.2.1 Intensity Slicing

Consiste nel trasformare un'immagine a scala di grigi in un'immagine a colori mappando ogni range di intensità ad un colore specifico. Il razionale sull'impiego di questa tecnica è che l'occhio umano è in grado di percepire e differenziare meglio i colori che diverse intensità di grigio, per cui potrebbe essere utile per analizzare particolari dettagli di immagini.

2.2.2 False Color Transformation

Un'altro tipo di trasformazione consiste invece nel mappare i valori di rosso/-verde e giallo scambiandoli di ordine oppure variandone le intensità. Questa trasformazione è facilmente implementabile per mezzo di una trasformazione lineare.

3 Image processing types

Esistono diverse tipologie di processamenti di immagini. Innanzitutto, un'immagine può essere processata in due domini differenti:

- Dominio spaziale: direttamente sui pixel dell'immagine.
- Dominio delle frequenze: rappresenta il contenuto delle frequenze spaziali, cioè *quanto* varianto le insensità dei vari pixel in direzione orizzontale/verticale.

Le tecniche di image processing che avvengono nel dominio delle frequenze sono l'analogo delle tecniche di processamento del suono, solamente che il segnale si evolve nello spazio anzichè nel tempo, per questo motivo si parla di *frequenze spaziali*.

Le tipologie di tecniche per il processamento delle immagini possono essere categorizzate in modo più generale.

3.1 Image Enhancement

L'obiettivo principlae dell'enhancement è di preparare l'immagine ad un'ulteriore processing in modo che questo processing sia più effettivo. Un esempio potrebbe essere quello di applicare un'operazione di rimozione del rumore per migliorare un task di riconoscimento di oggetti.

La valutazione dei risultati di questo tipo di processamento è puramente *soggettiva*, cioè non esistono metriche oggettive per valutare la qualità del risultato. In questo senso, possiamo definire l'image enhancement come un processo che migliora la qualità **percepita da un umano** di un'immagine. Ne deriva naturalmente che le tecniche utilizzate per migliorare la qualità di immagini sono fortemente *application-driven* (*es.* algoritmi usati nel campo medico potrebbero essere totalmente inutili in ambito astronomico).

3.2 Image Restoration

L'obiettivo della restoration è quello di migliorare la qualità dell'immagine, possibilmente ripristinandone parti o caratteristiche perse. La restoration viene applicata quando si ha una conoscenza (parziale o totale) sulla *funzione di degradazione*.

Contrariamente all'enhancement, la restoration è invece un procedimento il cui risultato può essere valutato *oggettivamente*.

3.3 Compression

La compressione ha invece l'obiettivo di ridurre lo spazio richiesto per salvare un'immagine o il tempo richiesto per trasmettere tale immagine in un medium di trasmissione. Ci sono due tipologie principali di compressione:

- **Lossy**: l'immagine decompressa potrebbe avere differenze rispetto all'originale.
- **Lossless**: l'immagine decompressa è identica all'immagine originale.

Alcuni esempi di lossy compression possono esssere:

- **Subsampling**: in cui solo un sottinsieme dei pixel viene salvato. In questo caso, il processo di decompressione dovrà far fronte alla mancanza di pixel impiengando tecniche di replicazione o di interpolazione. In ogni caso l'immagine decompressa non sarà identica all'originale.
- **Quantizzazione**: in cui si decresce il livello di quantizzione, riducendo così il numero di possibili livelli di intensità, per cui saranno necessari meno bits per codificare il valore di un *color channel* (o livello di grigio). All'estremo di questa tecnica possiamo ottenere immagini a due livello di colore (bianco e nero), dette immagini *binarizzate*.

- **Sampling e Quantizzazione**: l'idea è quella di modificare adattivamente sia il livello di quantizzazione che il livello di sampling. Questi due livelli vengono fatti variare in base ai livelli di dettaglio in una particolare area dell'immagine. In questa maniera, il numero corretto di campioni e di livelli d'intensità sarà utilizzato per rappresentare i pixel in quella determinata area.

3.4 Segmentation

La segmentazione è un processo di computer vision che consiste nel partizionare un'immagine per individuare le componenti *semanticamente* utili ad un determinato task. Tipicamente dopo un task di segmentazione è necessario:

- Rappresentare le parti segmentate (utilizzando la sua area interna, oppure i suoi bordi).
- Descrivere le parti segmentate (in termini di area, perimetro, lunghezza dei bordi).

4 Image Enhancement

Le tipologie di image enhancement si suddividono in tipologie di trasformazione che possono essere:

- **Puntuali**: trasformazione da pixel a pixel, rappresentabili come una mappa $(x_t, y_t) = f(x, y)$.
- **Locali**: trasformazione da insieme di pixel a pixel. L'insieme di pixel è rappresentato da una *funzione di neighborhood* $I(x, y)$, per cui la trasformazione è rappresentata dalla mappa $(x_t, y_t) = f(I(x, y))$.
- **Globali**: trasforma un'intera immagine in un singolo punto.

4.1 Intensity Tranformation

Una tipologia di image enhancement è l'intensity tranformation. Abbiamo visto questa trasformazione in particolare per le immagini rappresentate in scala di grigi. Una trasformazione puntuale T di questo tipo mappa ogni valore di intensità in un'altro valore. Questa funzione può essere implementata attraverso una tabella di conversione chiamata LUT (*Look Up Table*). Queste tabelle vengono rappresentate per mezzo di un grafico bidimensionale che esprime i nuovi valori di T in corrispondenza dei vari valori di grigio.

Una trasformazione T molto importante è la *funzione gamma*, che implementa la cosiddetta *gamma correction*. Tale operazione permettere di espandere/comprimere i livelli di grigio scuri/chiari. La funzione gamma è la seguente:

$$s = c \cdot r^{\gamma}$$

dove:

- c è una costante pre-definita.
- r è il livello di intensità di grigio in input.
- $\gamma > 0$ è l'iperparametro della funzione.

Valori di $\gamma > 1$ determinano un'amplificazione dei livelli di grigio scuri, attenuando il livello di quelli chiari. D'altra parte, con $0 < \gamma < 1$ si ottiene un'amplificazione dei livelli di grigio chiari, attenuando quelli scuri.

Altre traformazioni che sono state viste di questo tipo sono:

- **Ladder**: grafo a *"scaletta"* che riduce il numero di livelli di intensità, creando dei contorni falsi. Questa trasformazione ha il problema che introduce un tipo di compressione irreversibile perchè si perde informazione data dalle linee verticali della funzione (i "salti").
- **Ramp**: grafo che corrisponde ad una funzione a tratti collegata da una rapida ascesa tra i valori minimi e massimi di intensità in un intervallo compreso tra r_1 e r_2. Questa trasformazioe essenzialmente serve ad ridurre il contrasto dei pixel che hanno intensità troppo bassa ($< r_1$) o troppo alta ($> r_2$). L'operazione è detta *contrast stretching*.
- **Binarization**: come suggerisce il nome, trasforma i valori di intensità in intensità massima (1) e minima (0). Il grafico di tale LUT è una funzione a tratti.

Fin'ora abbiamo partlato di LUT per trasformazioni di immagini a scala di grigi. Tipicamente, nei casi di immagini a colori si impiegano le *pseudocolor* LUT. Essenzialmente sono tabelle che hanno una LUT per ogni canale di colore.

4.2 Bit-Plane Slicing

La tecnica di bit-plane slicing consiste nel rappresentare un'immagine (a scala di grigi) inizialmente in binario e poi creare un'immagine binarizzata per ogni posizione di bit di ogni pixel. Ad esempio, se ci concentriamo su un dato pixel con valore 010, l'operazione produrrà 3 piani in cui lo stesso pixel avrà valore 0 nel primo, 1 nel secondo e infine 0 nel terzo piano.

4.3 Histogram Processing

È una tecnica che ricade nelle tecniche di *global processing*. Innanzitutto, l'istogramma dei livelli di grigio di un'immagine è una funzione discreta h definita come

$$h(r_k) = n_k$$

dove:

- r_k: è k-esimo livello di grigio ($r_k \in [0; 255]$ nel caso siano 8bpp).
- n_k: è il numero (*frequenza*) di pixel nell'immagine con intensità r_k.

Tipicamente, per far si che i valori siano tutti tra 0 e 1, si utilizza una versione normalizzata dell'istrogramma:

$$p(r_k) = \frac{n_k}{M \cdot N}$$

dove M sono il numero di pixel per riga e N il numero di pixel per colonna.

La versione normalizzata di un istogramma definisce la probabilità che un qualsiasi pixel abbia uno specifico valore di grigio r_k. La sua rappresentazione grafica dà una descrizione globale dell'immagine. In questo senso possiamo anche definire la *frequenza cumulativa* (o Cumulative Density Function)

$$F(r_k) = \sum_{i=0}^{k} p(r_i)$$

Gli istogrammi riassumono quindi la distribuzione dell'intensità di pixel nell'immagine. Ad esempio, istogrammi di immagini molto scure, avranno una distribuzione *skewed* verso valori bassi di r_k. Altri casi interessanti sono gli istogrammi delle immagini a basso contrasto (che appariranno come una distribuzione maggiormente concentrata verso i valori centrali), e quelle a basso contrasto (con una distribuzione molto omogenea su tutti i possibili valori).

4.3.1 Histogram Equalization

Come suggerisce il nome, il metodo della histogram equalization serve ad aumentare il *contrasto* globale dell'immagine. Lo scopo di questa tecnica è quindi quello di modificare le intensità dei pixel in modo che l'istogramma risultante sia il più possibile uniforme. Di conseguenza, i valori dei pixel dell'immagine saranno quindi distribuiti più uniformemente, sfruttando l'intero range dei valori disponibili.

La trasformazione $s_k = T(r_k)$ per fare histogram equalization sarà una mappa monotonicamente crescente da $L-1$ a $L-1$. Questo perchè r_k rappresenta una singola intesità di pixel, che verrà mappata ad un altra intensità in modo da equalizzare l'istogramma. Siano $p_r(r_k)$ e $p_s(s_k)$ le rispettive distribuzioni delle intensità r_k e s_k, allora sfruttando un risultato della teoria delle probabilità, possiamo trasformare la distribuzione p_r in p_s nel modo seguente

$$p_s(s_k) = p_r(r_k) \left| \frac{dr}{ds} \right|$$

Se consideriamo la funzione

$$T(r) = s = (L-1) \int_0^r p_r(z) dz$$

otteniamo, applicando la formula precedente che la risultante PDF sarà

$$p_s(s) = \frac{1}{L-1}$$

L'istogramma equalizzata sarà quindi ottenuto applicando la seguente trasformazione

$$T(r_k) = (L-1)\frac{H(r_k)}{MN}$$

dove:

- $H(r_k) = \sum_{i=0}^{k} h(r_i)$ è *l'istogramma cumulativo* (come la distribuzione cumulativa ma non sulle *probabilià* ma sulle *frequenze*).
- M numero di colonne e N numero di righe.
- $L-1$ livelli di intensità (255 per immagini a 8bpp).

Un problema di questa tecnica è che queste trasformazioni potrebbero mappare più valori di intensità sullo stesso valore, di fatto causando una perdita di dettagli nell'immagine originale.

4.3.2 Histogram Specification

L'histogram specification consiste nel mappare i valori di intensità in modo da cambiare la forma dell'istogramma, senza limitarsi solamente ad una forma uniforme come nell'equalization.

Essenzialmente vogliamo trsformare l'istogramma in un istogramma con distribuzione $p_z(z)$. Possiamo seguire lo stesso ragionamento dell'*histogram equalization* e ottenere quindi la trasformazione $G(z)$ come

$$G(z) = s = (L-1)\int_0^z p_z(v)dv$$

Si hanno quindi una trasformazione T da r a s discussa precedentemente e una trasformazione G da z a s. Quello che si vuole ottenere è però una trasformazione da r a z. Per ottenerla basta fare l'inversa G^{-1}, per cui la mappa per ottenere z a partire da r sarà definita nel modo seguente

$$z = G^{-1}(T(r))$$

Per fare histogram specification quindi abbiamo 3 passi definiti:

1. Utilizza l'immagine originale per ottenere $T(r)$.
2. Utilizza la distribuzione desiderata $p_z(s)$ per calcolare $G(z)$, invertendola successivamente.
3. Per ogni pixel, ottieni la sua traformazione s applicando $T(r)$, poi applica G^{-1} al risultato s, ottenendo z.

4.4 Filtri Spaziali

Un'altra tecnica di image enhancement consiste nell'applicare dei filtri spaziali. I filtri spaziali permettono di fare un processing *locale* all'immagine. L'idea è quella di definire una maschera bidimensionale che determina *"quanto"* la singola intensità deve essere amplificata o ridotta, e successivamente sovraimporre tale

maschera all'immagine scorrendo via via tutte le possibili posizioni. Questa operazione è detta *convoluzione* ed è riassumibile (nel caso bidimensionale e discreto) dalla seguente formula

$$y(n,m) = \sum_{k=-a/2}^{a/2} \sum_{l=-b/2}^{b/2} x(n-k, m-l)h(k,l)$$

dove:

- $h(k,l)$ è la *maschera* definita nella regione $(-a,a)$, $(-b,b)$.
- $x(n,m)$ è l'intensità del pixel alla posizione (n,m).

Ovviamente bisogna tenere conto dei pixel ai bordi dell'immagine, poichè la maschera potrebbe "*uscire*" al di fuori dell'immagine. In questi casi si effettua un'operazione di *0-padding*, cioè si estende l'immagine opportunamente di valori pari a 0.

In generale i valori di h sono normalizzati a 1, alternativamente è necessario dividere la formula per la somma di tutti i pesi della maschera. I valori indicano se il tipo del filtro è passa-basso o passa-alto. Valori = 1 lasciano passare il segnale inalterato, mentre = 0 lo filtrano.

Possiamo vedere questo tipo di processamento essenzialmente come una media pesata specificata dalla maschera (o filtro). Dal punto di vista operativo, i filtri passa basso vanno a rimuovere i dettagli dell'immagine, evidenziando aree grandi nell'immagine.

4.4.1 Ordered Statistics Filters

Un'altra tipologia di filtri è quella basata sulle statistiche ordinate come la *mediana*. Un filtro mediano funziona andando a fare una statistica dei pixel definiti nella finestra del filtro, ordinandoli in ordine ascendente e prendendo il valore centrale come l'output del fitro. Questo tipo di filtri è particolarmente utile per rimuovere del rumore di tipo *salt&pepper*.

4.4.2 Filtri Passa Alto

Vengono utilizzati in task come *sharpening* o *contour extraction*. Questa tipologia di filtri viene ottenuta andando a simulare un operatore di derivazione. Sono stati visti essenzialmente 2 operatori basati su questa tecnica: il Laplaciano e il Jacobiano.

4.4.3 Operatore Laplaciano

Il primo operatore visto è il *Laplaciano*. Questo operatore permette di fare *contour extraction*, cioè evidenzia i contoni (o bordi) degli oggetti presenti nelle immagini. L'idea nasce dal fatto che la derivata seconda è proporzionale alle variazioni di intensità nell'immagine. Essa è difatti $\neq 0$ solo in prossimità di

cambiamenti dei valori di intensità. Seguendo questa intuizione, si introduce il Laplaciano, un operatore che approssima la derivata seconda di una funzione

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

nel caso delle immagini si utilizza il filtro Laplaciano isotropico (cioè un filtro *invariante rispetto alle rotazioni*).

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Notiamo essenzialmente che la definizione di questa derivata non è nient'altro che una somma pesata per cui possimo definire un kernel (maschera) che potrà essere combinata eventualmente con l'immagine facendone un'operazione di convoluzione. Il kernel risultante sarà:

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

Si noti che il kernel precedente non è nient'altro che la somma dei due kernel calcolati su una specifica direzione

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Una volta definito il Laplaciano, possiamo utilizzarlo per fare image sharpening.

$$g(x,y) = f(x,y) + c \cdot \nabla^2 f(x,y)$$

in cui g è l'immagine risultato dell'operazione, mentre f è l'immagine originale. c è un coefficiente che può essere 1 oppure -1, tipicamente non si scelgono valori grandi poichè aumenterebbero il rumore nell'immagine. L'evidenza che l'operatore faccia sharpening è data dal fatto che la derivata seconda assume valori più grandi in prossimità in cui ci sono bruschi cambiamenti di intensità. L'operazione rinforzerà quindi i bordi degli oggetti nell'immagine, oscurando (rendendo nero) lo sfondo. Il fatto che si aggiunge il Laplaciano può essere visto intuitivamente come l'aggiunta di un segnale che ha valori alti su porzioni in cui cambia bruscamente l'immagine, mentre valori bassi in tutte le altre porzioni.

4.4.4 Operatore Gradiente

Il gradiente è invece la derivata prima di una funzione a più variabili (nel nostro caso parliamo di funzioni bivariate). Non è nien'altro che un *vettore* a due componenti che punta nella direzione che ha l'intensità massima.

$$\nabla f = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j}$$

Per ottenere una singola componente numerica si considera il modulo M del gradiente

$$M(x,y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Spesso però per motivi di linearità si considera l'approssimazione del modulo seguente:

$$M(x,y) = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$

Tra gli operatori che vanno a calcolare il gradiente abbiamo l'operatore di *Roberts*, che calcola la derivata sulla direzione *diagonale*. Ad esempio, la maschera che implementa l'operatore di Roberts per calcolare il gradiente (rispetto alla diagonale in cui ci sono valori diversi da 0) è il seguente

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

È importante sottolineare che il pixel che viene considerato ogni volta durante l'operazione di convoluzioe viene "allineato" alla componente $(0,0)$ della maschera. Questo perchè non esiste un centro ben definito essendo la maschera 4×4. Proprio per il fatto che utilizzare maschere di dimensioni pari provoca questo problema si preferisce utilizzare l'operatore di *Sobel*, che è invece definito da una maschera 3×3. Gli operatori di Sobel non si concentrano sulle componenti diagonali ma sulle direzioni verticali e orizzontali. Ad esempio, la seguente maschera implementa l'operatore di *Sobel* che enfatizza le linee orizzontali

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

La somma dell'operatore per le linee verticali e quello per le linee orizzontali ci da il gradiente.

Un'altro operatore che implementa il gradiente è l'operatore di *Prewitt*, essenzialmente questo operatore dà meno importanza al pixel centrare rispetto all'operatore di *Sobel*. L'operatore di Prewitt per le linee orizzontali è implementato nel modo seguente (quello per le linee verticali ne è semplicemente una rotazione di 90 gradi in senso antiorario)

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Anche in questo caso, per ottenere il gradiente si fa la somma delle due derivate direzionali (prima si calcola poi si fa la somma).

Per riassumere, sia il Gradiente che il Laplaciano servono a evidenziare i contorni (sono sensibili ad essi), ma è importante sottolineare varie differenze tra i due. In generale, il Laplaciano:

- Mantiene i dettagli;
- È più sensibile al rumore (più evidente in zone lisce);
- Viene utilizzato per fare *sharpening*.

Mentre il Gradiente ha una risposta più marcata in aree in cui ci sono transizioni di grigio molto significative, in cui i bordi sono particolarmente più spessi, per cui è preferibile per fare *edge detection*.

4.5 Contour Enhancement (Local Histogram Statistics)

Un'altro metodo di enhancement a livello spaziale è il cosiddetto contour enhancement. L'obiettivo ad esempio potrebbe essere quello di migliorare delle aree scure dell'immagine lasciando invariate quelle più chiare. L'idea alla base è quella di utilizzare delle statistiche *locali*, definite in un'opportuna finestra di dimensioni $N \times M$. Ad esempio, la media sarà

$$\mu = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{x=0}^{M-1} f(x, y)$$

ed eventualmente la varianza locale

$$\sigma^2 = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{x=0}^{M-1} [f(x, y) - \mu]^2$$

A questo punto, il processing andrà a considerare pixel per pixel, considerando opportunamente per ognuno di esso queste statistiche. Il processing verrà effettuato solo se i seguenti vincoli sono soddisfatti:

- Se $\mu_s \leq k_0 \cdot \mu_G$ (dove μ_s è il valore medio *locale*, μ_G la media globale e $0 \leq k_0 \leq 1$ un opportuno valore fissato), in modo tale da considerare solamente i valori al di sotto di una determinata soglia di threshold, ignorando quelli al di sopra della stessa;
- Se $\sigma_S \leq k_2 \sigma_G$, in modo da considerare solo i pixel che non sono già in una zona ad altro constrasto;
- Se $k_1 \sigma_G \leq \sigma_S$ (ovviamente $k_1 < k_2$) in modo tale da evitare di aumentare il contrasto in zone uniformi.

Mettendo insieme tutti questi vincoli, si ottiene la seguente formula riassuntiva per l'aumento locale di contrasto:

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } \mu_s \leq k_0 \mu_G \text{ and } k_1 \sigma_G \leq \sigma_S \leq k_2 \sigma_G \\ f(x, y) & \text{otherwise} \end{cases}$$

Questo tipo di processamento può essere applicato per fare *contour enhancement*, in generale, zone che hanno una varianza locale molto alta si traducono in un'alta variabilità delle intensità di pixel (all'interno della finestra). Le zone che contengono bordi sono spesso tali da avere un'alta varianza, siccome contengono

spesso molti pixel di varie intensità. D'altra parte, zone di *background* avranno un numero più ristretto di intensità, risultando quindi in zone con varianza più piccola.

5 La trasformata di Fourier

5.1 Introduzione

L'idea alla base dello studio dei segnali periodici di Fourier è che ogni funzione **periodica** può essere scritta come la **somma** di *seni* e *coseni*, di differenti *ampiezze* e *frequenze* (in caso solo un tipo di funzione venga utilizzato - seno o coseno - allora si utilizza anche la *fase*). Questa somma viene chiama **serie di Fourier**.

Nel caso invece delle funzioni **non-periodiche**, si possono rappresentare come un *integrale* di *seni* e *coseni* pesati (da un'opportuna funzione). Questa rappresentazione viene invece chiamata **trasformata di Fourier**.

> *La rappresentazione nel **dominio di Fourier** è* equivalente *all'originale.*

5.2 La serie di Fourier

Come già citato in precedenza, ogni funzione periodica di periodo T (o equivalentemente di frequenza $\omega = \frac{2\pi}{T}$) può essere espresso come

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega nt} \quad \text{con } n \in \mathbb{Z}$$

dove i coefficienti calcolati con la seguente formula

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega nt} dt$$

i coefficienti vanno ad indicare *quanto* la frequenza fondamentale ωn è presente all'interno del segnale $f(t)$. In altri termini, questo integrale va a calcolare la similarità tra il segnale *"sonda"* $f(t)$ e la frequenza fondamentale $e^{j\omega nt}$.

> **Formula di Eulero**: $e^{j\omega} = cos(\omega) + jsin(\omega)$

Si noti che T indica la *risoluzione* della pulsazione in frequenza. Tanto più T è grande, tanto più sarà grande.

5.3 La trasformata di Fourier

Per fare da *"ponte"* tra la serie e la trasformata di Fourier, è necessario introdurre un concetto fondamentale: la distribuzione *delta di Dirac*. Tale funzione (più correttamente *distribuzione*) è caratterizzata da un impulso di ampiezza infinita

e di una durata infinitamente piccola (puntuale), ed è definita matematicamente nel modo seguente

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

inoltre, essendo una distribuzione deve valere che

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

ha una proprietà molto importante detta *shifting property* (oppure *sampling property*) tale per cui, presa una funzione $f(t)$ e moltiplicata per la *delta di Dirac*, ne ritorna il valore al punto 0 (in generale, nel punto in cui è centrata la funzione δ). Tale proprietà generale è riassumibile nel modo seguente

$$\int_{-\infty}^{\infty} f(t)\delta(t-u)dt = f(t-u)$$

L'intuizione è che siccome la distribuzione δ è 0 ovunque tranne che intorno ai valori $t-u$, l'integrale anche è nullo per cui si possono considerare solo gli estremi di integrazione in un intorno infinitesimamente piccolo di $t-u$. In questo intorno la funzione $f(t-u)$ non cambia, per cui è costante e può esser portata fuori dall'integrale. A questo punto, l'integrale della distribuzione δ è 1, per cui ne deriva la dimostrazione.

È importante notare che questa proprietà vale anche nel discreto, creando di fatto una connessione tra la trasformata e la serie che verrà discussa più avanti.

Avendo introdotto tutto il necessario, per ottenere la trasformata dalla serie facciamo tendere (nella definizione di serie di Fourier) $T \to \infty$. Focalizzandosi inizialmente solo sul calcolo dei coefficienti c_n, notiamo in primo luogo (dalla definizione data precedentemente), che l'estremo di integrazione diventa da $-\infty$ a $+\infty$. Inoltre, la variabile discreta n, siccome è divisa per T, indica delle frequenze arbitrariamente vicine, per cui $\frac{n}{T}$ viene sostituito da una variabile continua u. Infine, il termine moltiplicativo $\frac{1}{T}$ diventa per definizione il *differenziale* dell'integrale dt. Otteniamo quindi la cosiddetta **trasformata** di Fourier, riportata in seguito

$$F(u) = \int_{+\infty}^{-\infty} f(t)e^{-j2\pi ut}dt$$

Si noti che la funzione $F(u)$ è in funzione della *frequenza*, ed equivale alla rappresentazione di f nel dominio delle *frequenze*. È possibile anche definire la trasformata inversa, semplicemente applicando lo stesso processo al limite alla definizione della *serie*, e sostituendo le definizioni dei coefficienti con la definizione di trasformata

$$f(t) = \int_{+\infty}^{-\infty} F(u)e^{j2\pi ut}du$$

tra la trasformata e l'inversa cambia solo il segno dell'esponenziale, per cui si dice che la trasformata rispetta la proprietà di **simmetria**.

Un'altra proprietà importante della trasformata di Fourier è che nonostante individui tutte le frequenze presenti in un segnale, le informazioni di tipo *spaziale* (o *temporale* nel caso di segnali nel tempo) vengono perse. Questo è dimostrabile considerando la trasformata di un segnale qualunque traslato poiché il risultato (in modulo) non cambia rispetto al segnale non traslato (dettagli matematici sule slides).

5.3.1 Trasformate importanti

5.3.1.1 Funzione Rettangolo Una delle poche funzioni di cui è necessaria una certa dimestichezza con la sua trasformata è la funzione *sinc*. Per introdurla, prendiamo un segnale a finestra (potrebbe, per esempio, essere una riga di pixel) definita nel modo seguente

$$f(t) = \begin{cases} A & \text{for } -a \leq t \leq a \\ 0 & \text{otherwise} \end{cases}$$

La trasformata F di tale funzione è la cosiddetta *funzione sinc*:

$$F(u) = \int_{-a}^{+a} Ae^{-j2\pi ut}dt = \left[\frac{A}{-2j\pi u}e^{-2\pi ut}\right]_{-a}^{a} = \frac{sin(2a\pi u)}{2a\pi u} = a \cdot A \cdot sinc(2au)$$

> *Graficamente, la funzione sinc ha valore* 1 *in* $u = 0$*, e oscilla con* $u \to \infty$ *avvicinandoci asintoticamente all'asse* x*.*

Ovviamente, possiamo anche definire la trasformata per segnali bi-dimensionali, per cui la trasformata di un segnale rettangolare bi-dimensionale (ad esempio una porzione d'immagine) in questo caso sarebbe una funzione *sinc* bi-dimensionale.

5.3.1.2 Delta di Dirac Vale la pena considerare anche la trasformata della funzione δ introdotta precedentemente, poichè ha diverse caratteristiche degne di nota.

$$F(u) = \int_{+\infty}^{-\infty} \delta(t)e^{-j2\pi ut}dt = e^{-j2\pi u0} = 1$$

il fatto che la trasformata sia la funzione costante 1, indica che tutte le frequenze sono contenute all'interno di un implulso. In generale, vale che

$$F(u) = \int_{+\infty}^{-\infty} \delta(t - t_0)e^{-j2\pi ut}dt = e^{-j2\pi ut_0}$$

ciò significa che la trasformata della funzione delta di Dirac individua la frequenza fondamentale multipla di t_0.

Inoltre, grazie alla relazione della trasformata inversa otteniamo che

$$\delta(t) = \int_{+\infty}^{-\infty} e^{j2\pi ut}du$$

5.3.1.3 Segnale Monocromatico Consideriamo il segnale monocromatico $f(t) = Acos(2\pi u_0 t)$ e calcoliamone la trasformata

$$F(u) = \int_{+\infty}^{-\infty} A\ cos(2\pi u_0 t) e^{-j2\pi ut} dt = A\left(\delta(u-u_0) + \delta(u+u_0)\right)$$

ciò significa che la trasformata sarà composta da due *delta* di Dirac centrate nei punti u_0 e $-u_0$. Questo è ragionevole poichè la trasformata è sempre simmetrica rispetto all'asse y, per cui la funzione *delta* individua la frequenza u_0, che è effettivamente l'unica frequenza contenuta nella funzione originale.

5.4 Teorema del campionamento

Per campionare un segnale possiamo utilizzare la proprietà vista in precedenza della funzione δ di Dirac. L'idea è quella di definire una serie di funzioni δ equispaziate di un certo ΔT (chiamato *periodo di campionamento*). Si costruisce quindi il seguente *treno di impulsi*

$$s(t) = \sum_{n=-\infty}^{+\infty} \delta(t - n\Delta T)$$

la cui trasformata sarà

$$S(u) = \frac{1}{\Delta T} \sum_{n=-\infty}^{+\infty} \delta(u - \frac{n}{\Delta T})$$

La trasformata è ancora un *treno di impulsi*, equispaziati di $\frac{n}{\Delta T}$. Si noti che tanto più gli impulsi sono vicini nella funzione iniziale (quindi valori ΔT piccoli che corrispondono ad una frequenza di campionamento alta), tanto più gli impulsi nella trasformata saranno distanti.

Intruitivamente, quindi, se cerchiamo di avvicinare le varie delta tra loro nel dominio spaziale (o temporale), le distanziamo nel dominio delle frequenze. Questo è ragionevole perché la trasformata andrà a coprire frequenze più alte.

5.5 Teorema della Convoluzione

Precedentemente abbiamo trattato la convoluzione come operazione di media mobile nel discreto, ma è possibile definirla anche nel continuo. La convoluzione di due funzioni h e f è così definita

$$f(t) \star h(t) = \int_{-\infty}^{\infty} f(\tau) h(t-\tau) d\tau$$

Teorema 1 (Di Convoluzione)**.** *La convoluzione di due funzioni h e f è pari al prodotto delle loro rispettive trasformate H e F, formalmente*

$$f(t) \star h(t) = F(u) \cdot H(u)$$

La dimostrazione del teorema è banale, poichè basta applicare la definizione di trasformata all'operazione di convoluzione. Calcoliamo innanzitutto la trasformata di $h(t-\tau)$ per semplificare i calcoli successivamente

$$\begin{aligned}\mathscr{F}[h(t-\tau)] &= \int h(t-\tau)e^{-j2\pi ut}dt \\ &= \int h(v)e^{-j2\pi u(v+\tau)}dv \ (v = t-\tau) \\ &= e^{-j2\pi u\tau}\int h(v)e^{-j2\pi v}dv \\ &= H(u)\cdot e^{-j2\pi u\tau}\end{aligned}$$

passiamo ora a calcolare la trasformata della convoluzione, ottenendo quindi la dimostrazione. Il passaggio principale consiste nel cambiare l'ordine degli integrali, sostituendo successivamente le definizioni che compaiono

$$\begin{aligned}\mathscr{F}[f\star h] &= \int\left(\int f(\tau)h(t-\tau)d\tau\right)e^{-j2\pi ut}dt \\ &= \int f(\tau)\left(\int h(t-\tau)e^{-j2\pi ut}dt\right)d\tau \\ &= \int f(\tau)H(u)e^{-j2\pi u\tau}d\tau \\ &= H(u)\int f(\tau)e^{-j2\pi u\tau}d\tau \\ &= F(u)\cdot H(u) \quad \blacksquare\end{aligned}$$

da questo teorema risulta evidente il perché h venga anche chiamato *filtro*. Vale quindi che tutte le operazioni che abbiamo definito nel dominio dello spazio per mezzo di un'operazione di convoluzione, possono essere fatte anche nel dominio delle frequenze, semplicemente per mezzo di una moltiplicazione.