

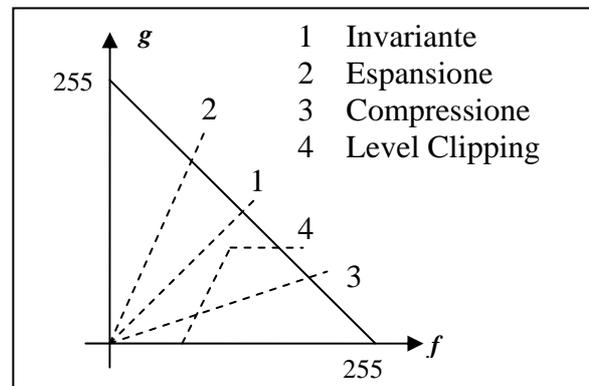
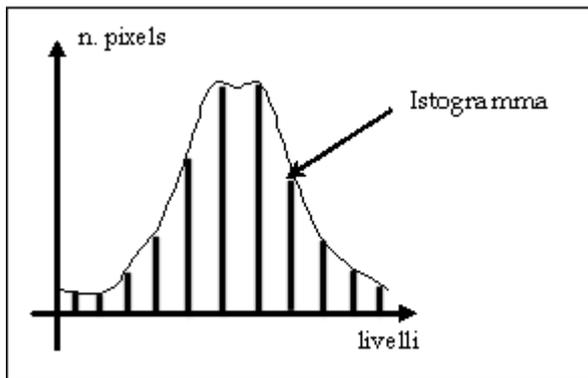
Corso di programmazione – Test Finale

Fra le tante tecniche possibili di *Image Processing*, ne esiste una famiglia chiamata Point Operation, le quali tengono conto solo dell'intensità luminosa (per semplicità, d'ora in poi tratteremo solo immagini bianco e nero, in cui l'intensità luminosa corrisponde al valore in toni di grigio dei singoli pixel dell'immagine, dove un valore uguale a 0, corrisponde al colore nero, il valore massimo rappresentabile, ad esempio 255, corrisponde al bianco ed i valori intermedia ai toni di grigio) e non della posizione del pixel nell'immagine. Queste tecniche agiscono allo stesso modo in tutti i punti dell'immagine.

Uno strumento importante per questo genere di manipolazioni è l'istogramma, cioè un grafico che riporta per ogni livello di grigio, quanti pixel hanno quella tonalità: si tratta quindi di un grafico monodimensionale (qualunque sia la dimensionalità dell'immagine) che ha in ascissa l'intensità di colore dei pixel e in ordinata il numero di pixel che hanno quel determinato valore di intensità. Per un'immagine a toni di grigio ad 8-bit ci sono 256 possibili valori d'intensità e quindi l'istogramma mostrerà 256 punti in ascissa con il relativo numero di pixel con quella intensità in ordinata.

L'istogramma può anche essere eseguito su immagini a colori ed in questo caso si ottengono tre diversi istogrammi che corrispondono ai tre canali R, G, B.

Se questo grafico è normalizzato tra zero ed uno, si ottiene la densità di probabilità dei livelli di grigio per quell'immagine.



Equalizzazione dell'istogramma

Si tratta di una tecnica che mira a modificare la forma dell'istogramma ridistribuendo i valori dei livelli di grigio in modo che l'istogramma sia quanto più uniforme possibile, per migliorare la distribuzione dei vari livelli spandendoli su tutto l'intervallo dei possibili valori, migliorando ad esempio il contrasto dell'immagine.

Matematicamente, questa tecnica consiste nell'applicare a tutti i punti dell'immagine una funzione

$$g = T[f(m, n)]$$

la quale realizza una trasformazione sui livelli di intensità luminosa (nel caso, come detto, di immagini bianco e nero equivale ai toni di grigio) che "remappi" i

valori in modo da cambiarne la distribuzione sull'istogramma (ad esempio per rendere tale distribuzione più uniforme).

In pratica, per ogni pixel, la funzione g cambia il valore del livello di grigio.

La trasformazione più semplice che si può applicare all'istogramma è quella di assegnare alla funzione g una trasformazione lineare:

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

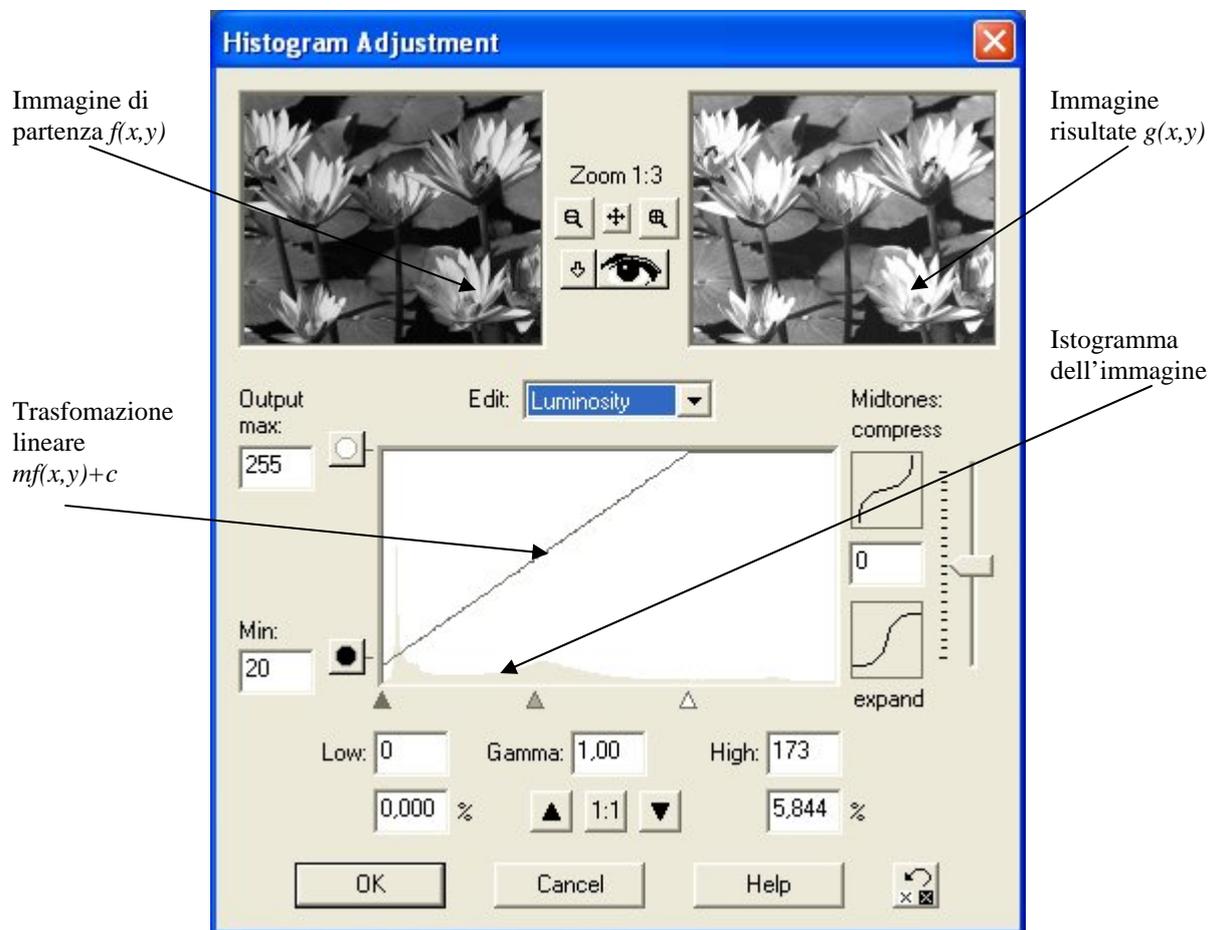
$$0 \leq g(x, y) \leq 255$$

dove

$f(x, y)$ è il valore dell'intensità luminosa nel punto (x, y)

$g(x, y)$ è il valore risultante dall'applicazione della trasformazione, normalizzato tra 0 e 255

m e c sono i parametri della trasformazione, corrispondenti ai parametri della retta che realizza la trasformazione



Test n.1

Sulla base di quanto detto in precedenza, produrre una classe che implementi il calcolo dell'istogramma per un'immagine monocromatica a 256 livelli di grigio.

Sarà fornita una classe, debitamente documentata, che si occupa di aprire un file contenente l'immagine in formato PGM (Portable Gray Map) e di organizzare l'immagine in memoria in un'apposita struttura dati e funzioni membro in grado

di produrre le dimensioni dell'immagine, il valore del tono di grigio di un singolo pixel e modificarlo, così come salvare l'immagine in un file di output.

Il candidato non dovrà occuparsi dei dettagli dell'implementazione della suddetta classe, dovrà solo utilizzarla nel suo progetto, basandosi sulle specifiche fornite nella documentazione.

La classe che si richiede dovrà contenere una funzione (il costruttore o una qualsiasi funzione membro) che prende come parametro un oggetto della classe immagine di cui sopra e ne crea l'istogramma, che consisterà in una struttura dati (un semplice vettore di 256 posizioni, ad esempio) alla quale accedere SOLO tramite un'altra funzione che ritorna (per copia) i valori dell'istogramma.

Test n.2

Estendere la classe per la creazione dell'istogramma, di cui sopra, aggiungendo una funzione che applichi ai valori dei pixel la trasformazione lineare

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

$$0 \leq g(x, y) \leq 255$$

Dove i valori dei parametri m e c sono passati alla funzione, così come un oggetto di tipo immagine che conterrà l'immagine risultato dell'applicazione della trasformata, ovvero l'immagine risultante.

Aggiunta per il lavoro in gruppo (max. 2 persone)

Estendere la classe aggiungendo una funzione che scrive sul disco un'immagine che contiene il grafico dell'istogramma normalizzato, calcolato al punto n.1 del test.

L'immagine che conterrà l'istogramma dovrà avere dimensioni 256 x 256 e sarà così formata: in ogni colonna saranno presenti un numero di pixel neri (partendo dall'ultima riga) proporzionale alla densità di probabilità del corrispondente livello di grigio:

$$\text{numero_pixel_neri_colonna_n} = p(\text{tono_di_grigio_n}) * 256$$

I restanti pixel saranno bianchi.

Ricordo che il colore nero, in immagini a 256 toni di grigio con un byte di profondità di colore (il nostro caso), corrisponde al valore 0, mentre il colore bianco al valore 255, cioè il massimo numero rappresentabile con un byte.

È richiesta la consegna di:

- Il codice C++ che implementa la classe dell'istogramma.
- La documentazione della classe di cui sopra, consistente in un file di testo che descrive il funzionamento delle funzioni membro (nel minore dei casi tre) richieste sopra, in modo che un ipotetico utilizzatore della classe sia in grado di usarla (descrizione dello scopo della funzione, se necessario accenno al suo funzionamento, valori di input richiesti e dati di output prodotti) sul genere di quella fornita in allegato per la classe immagine.
- Un programma di esempio che illustri il funzionamento della classe istogramma di cui al test n.1 (nel minore dei casi un main con apertura del file immagine, creazione dell'istogramma e visualizzazione o salvataggio dei valori dell'istogramma) e per quanti vorranno sostenere il test a coppie anche l'aggiunta di cui sopra.
- Un secondo programma di esempio che illustri il funzionamento della trasformazione di cui al test n.2 (è sufficiente l'apertura del file, la creazione

dell'istogramma, l'applicazione della trasformata ed il salvataggio dell'immagine elaborata.)

Un'immagine di esempio, da utilizzare come dati per il proprio programma verrà fornita in allegato al testo della prova.