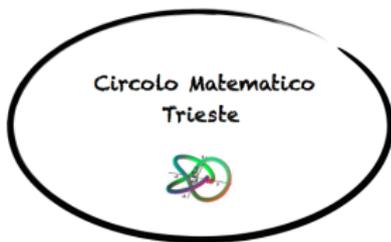


Matematica e immagini digitali

Marino Zennaro
Dipartimento di Matematica e Geoscienze, Università di Trieste
23 marzo 2018



*Le immagini digitali sono memorizzate sotto forma di una o tre tabelle numeriche rettangolari (**matrici**), a seconda che si tratti rispettivamente di un'immagine in scala di grigi o a colori. Le dimensioni delle tabelle (**numero di righe m e di colonne n**) dipendono dalla risoluzione dell'immagine, ossia dal numero di pixel che la compongono.*

Una modifica dei numeri contenuti nelle tabelle comporta una modifica corrispondente dell'immagine da esse rappresentata. Sulla base di ciò, formule matematiche più o meno complicate vengono comunemente utilizzate per operare sulle immagini una svariata gamma di trasformazioni.

In questa presentazione utilizziamo delle formule abbastanza semplici che ci permettono di ottenere alcune trasformazioni di base, che possono alle volte risultare anche divertenti.

Esempio 1.1 (scala di grigi)

$i \setminus j$	1	2	3	4	5	6	7	8	9	10
1	255	227	0	171	143	115	87	255	31	3
2	255	0	0	0	143	115	255	255	255	3
3	0	0	199	0	0	255	255	59	255	255
4	0	227	199	171	0	255	87	59	31	255
5	0	227	199	171	0	255	87	59	31	255
6	0	227	199	171	0	255	87	59	31	255
7	0	227	199	171	0	255	87	59	31	255
8	0	0	199	0	0	255	255	59	255	255
9	255	0	0	0	143	115	255	255	255	3
10	255	227	0	171	143	115	87	255	31	3

tabella numerica F ($m = 10$ righe e $n = 10$ colonne)

codifica a 8 bit ($2^8 = 256$ tonalità): da 0 (nero) a 255 (bianco)

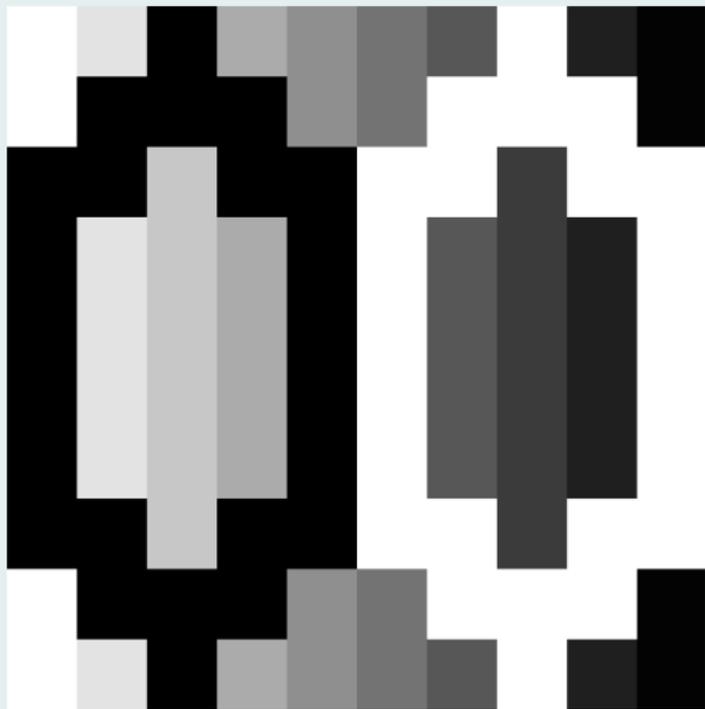


immagine corrispondente (ingrandita 60 volte)

Esempio 1.2 (colori)

$i \setminus j$	1	2	3	4	5	6	7	8	9	10
1	255	227	0	171	143	115	87	255	31	3
	255	255	50	221	193	165	137	255	81	53
	255	255	50	221	193	65	37	205	0	0
2	255	0	0	0	143	115	255	255	255	3
	255	50	50	50	193	165	255	255	255	50
	255	50	50	50	193	65	205	205	205	0
3	0	0	199	0	0	255	255	59	255	255
	50	50	249	50	50	255	255	109	255	255
	50	50	249	50	50	205	205	9	205	205
4	0	227	199	171	0	255	87	59	31	255
	50	255	249	221	50	255	137	109	81	255
	50	255	249	221	50	205	37	9	0	205
5	0	227	199	171	0	255	87	59	31	255
	50	255	249	221	50	255	137	109	81	255
	50	255	249	221	50	205	37	9	0	205
6	0	227	199	171	0	255	87	59	31	255
	0	177	149	121	0	205	37	9	0	205
	50	255	249	221	50	205	37	9	0	205
7	0	227	199	171	0	255	87	59	31	255
	0	177	149	121	0	205	37	9	0	205
	50	255	249	221	50	205	37	9	0	205
8	0	0	199	0	0	255	255	59	255	255
	0	0	149	0	0	205	205	9	205	205
	50	50	249	50	50	205	205	9	205	205
9	255	0	0	0	143	115	255	255	255	3
	205	0	0	0	93	65	205	205	205	0
	255	50	50	50	193	65	205	205	205	0
10	255	227	0	171	143	115	87	255	31	3
	205	177	0	121	93	65	37	205	0	0
	255	255	50	221	193	65	37	205	0	0

tabella numerica "tripla" (F_1, F_2, F_3) ($m = 10$ e $n = 10$)
 codifica a 8 bit: tre numeri tra 0 e 255 per determinare colore e tonalità

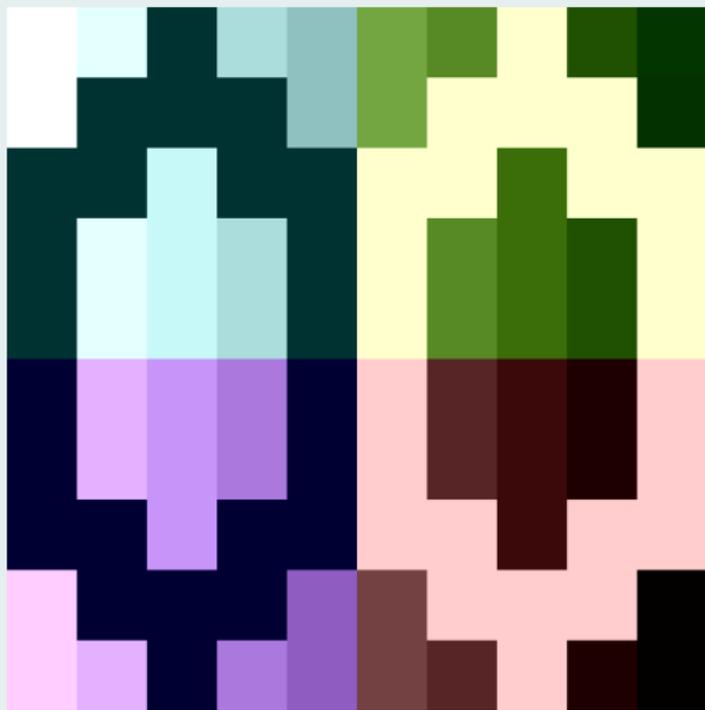


immagine corrispondente (ingrandita 60 volte)

Strumenti utilizzati e procedura generale

Per effettuare i calcoli necessari ad elaborare le immagini noi utilizziamo l'ambiente di programmazione **OCTAVE** (versione "public domain" del più noto **MATLAB**), liberamente scaricabile dal sito

<https://www.gnu.org/software/octave/>

Data un'immagine (generalmente in formato "bmp"), individuata dal file che chiamiamo *immagine.bmp*, utilizziamo il comando

```
> F=imread("immagine.bmp");
```

per acquisirla sotto forma di una **matrice** F a m righe e n colonne (caso in scala di grigi) o di una **terna di matrici** $F = (F_1, F_2, F_3)$ (caso a colori).

Denotiamo con $f_{i,j}$ il valore numerico di **posto** (i, j) nella **matrice** F e con $f_{i,j,k}$ quello di **posto** (i, j) nella **matrice** F_k della terna $F = (F_1, F_2, F_3)$. La numerazione (i, j) si esegue in ogni matrice da sinistra verso destra e dall'alto verso il basso. Abbiamo dunque

$$F = \|f_{i,j}\| \quad \text{oppure} \quad F = (F_1, F_2, F_3) = \|f_{i,j,k}\|$$

Eseguiamo poi le operazioni volute su F ed otteniamo una nuova matrice (o terna di matrici), che chiamiamo \tilde{G} .

Gli elementi $\tilde{g}_{i,j,k}$ di \tilde{G} non sono più necessariamente numeri interi e non sono nemmeno sempre appartenenti all'intervallo $[0, 255]$.

Dobbiamo quindi arrotondarli all'intero più vicino tramite la funzione

$$y = \text{round}(x) \left(= \text{parte intera di } (x + 0.5) \right)$$

e poi ridurli dentro l'intervallo $[0, 255]$ tramite la funzione

$$z = \min\{255; \max\{0; y\}\}$$

Queste due operazioni si eseguono su tutti gli elementi di \tilde{G} con il comando

```
> G=uint8( $\tilde{G}$ );
```

Infine creiamo il file *immagine_new.bmp* che individua la nuova immagine tramite il comando

```
> imwrite(G, 'immagine_new.bmp', 'bmp');
```

D'ora in poi eseguiremo sempre le varie operazioni su una sola matrice F , dato che la stessa procedura va semplicemente ripetuta tre volte se la si deve eseguire su una terna di matrici (F_1, F_2, F_3) .

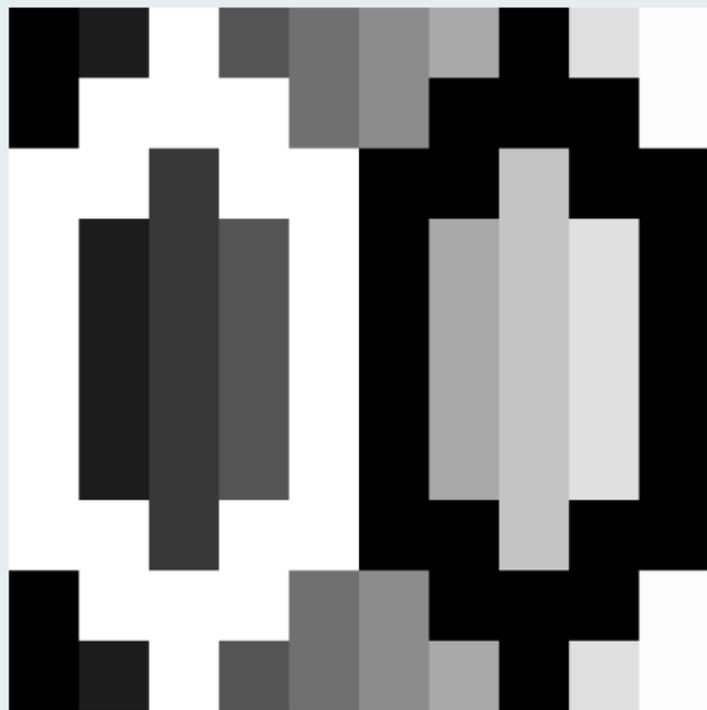
La trasformazione più semplice: l'immagine "negativa"

L'immagine "negativa" si ottiene sostituendo i vari colori con i loro "opposti". Ad esempio, il bianco al posto del nero ed il nero al posto del bianco.

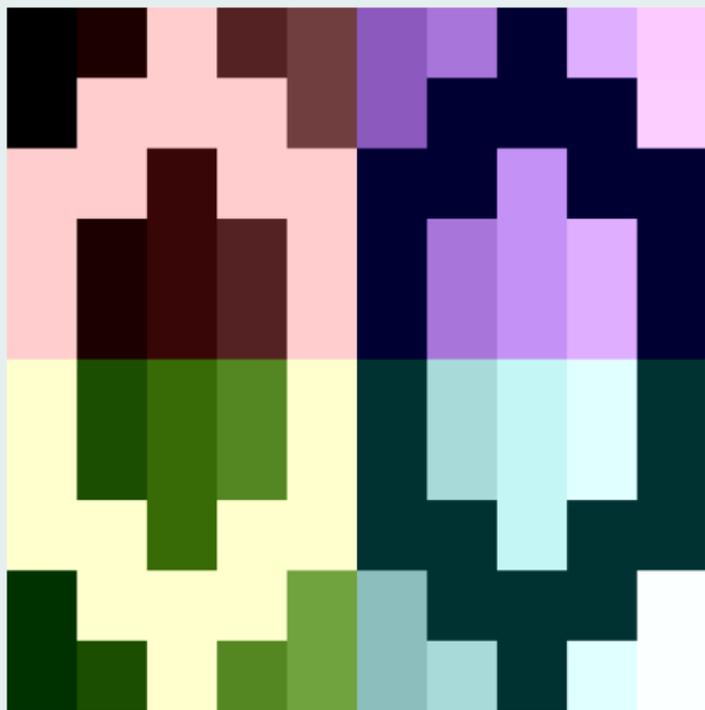
L'operazione da eseguire sugli elementi di F per ottenere G è semplicemente

$$g_{i,j} = 255 - f_{i,j}$$

N.B.: i numeri $g_{i,j}$ sono già interi dell'intervallo $[0, 255]$.



“negativo” dell’immagine in scala di grigi dell’Esempio 1.1



"negativo" dell'immagine a colori dell'Esempio 1.2

Esempio 2



immagine originale



immagine "negativa"

“Schiarire” e “scurire” le immagini

Possiamo “schiarire” l’immagine in maniera uniforme in modo altrettanto semplice.

Decidiamo di quanto schiarirla scegliendo un opportuno parametro intero $\delta > 0$ (e comunque ≤ 255).

L’operazione da eseguire sugli elementi di F per ottenere \tilde{G} è

$$\tilde{g}_{i,j} = f_{i,j} + \delta$$

Alcuni dei numeri $\tilde{g}_{i,j}$ possono risultare > 255 .

Dobbiamo quindi utilizzare la funzione `uint8` per ottenere la matrice G con elementi interi contenuti in $[0, 255]$.

N.B.: per $\delta = 255$ otteniamo un’immagine completamente bianca.



immagine dell'Esempio 2 schiarita con $\delta = 100$

Analogamente possiamo “scurire” l'immagine definendo

$$\tilde{g}_{i,j} = f_{i,j} - \delta$$

In questo caso alcuni dei numeri $\tilde{g}_{i,j}$ possono risultare < 0 .
Dobbiamo quindi utilizzare nuovamente la funzione `uint8` per ottenere la matrice G con elementi interi contenuti in $[0, 255]$.

N.B.: per $\delta = 255$ otteniamo un'immagine completamente nera.



immagine dell'Esempio 2 scurita con $\delta = 100$

“Rimodulazione” dei colori

Partiamo dalla semplice osservazione che

$$255 - z = -(z - 127.5) + 127.5$$

L'applicazione della formula di destra equivale a “spostare l'origine” dell'asse z da 0 a 127.5 (= 255/2) e poi a considerare il “valore opposto” di $z' = z - 127.5$ in questo nuovo sistema di riferimento.

Nasce quindi l'idea di considerare un generico “fattore di rimodulazione” α e di generalizzare la formula precedente per definire gli elementi di una nuova matrice \tilde{G} tramite la formula

$$\tilde{g}_{i,j} = \alpha(f_{i,j} - 127.5) + 127.5$$

I numeri $\tilde{g}_{i,j}$ non sono necessariamente numeri interi e, se $\alpha > 1$, alcuni di loro possono uscire dall'intervallo $[0, 255]$.

Dobbiamo quindi utilizzare la funzione `uint8` per ottenere la matrice G con elementi interi contenuti in $[0, 255]$.

Se si sceglie $\alpha > 1$ si opera una *dilatazione* delle tonalità dei vari colori presenti nell'immagine, che tendono tutte a spostarsi verso quelle estreme (ad esempio, bianco e nero nelle immagini in scala di grigi).
In particolare, per $\alpha \geq 255$ si ottengono matrici G con elementi tutti uguali a 0 o a 255, quindi immagini con le sole tonalità di colore estreme.

Se si sceglie $\alpha = 1$ non si effettua alcuna modifica alla matrice F e l'immagine rimane invariata.

Se si sceglie invece $0 < \alpha < 1$ si opera una *contrazione* delle tonalità, che tendono tutte a spostarsi verso le due colorazioni grigie medie, corrispondenti a 127 o 128.
In particolare, per $\alpha < 1/127.5$ si ottengono matrici G con elementi tutti uguali a 127 o a 128, quindi immagini completamente grigie.

Per $\alpha < 0$ si ottengono i medesimi effetti corrispondenti a $-\alpha$, ma si inverte il "segno" dei colori.



immagine dell'Esempio 2 rimodulata con $\alpha = 2$



immagine dell'Esempio 2 rimodulata con $\alpha = 255$



immagine dell'Esempio 2 rimodulata con $\alpha = 0.5$

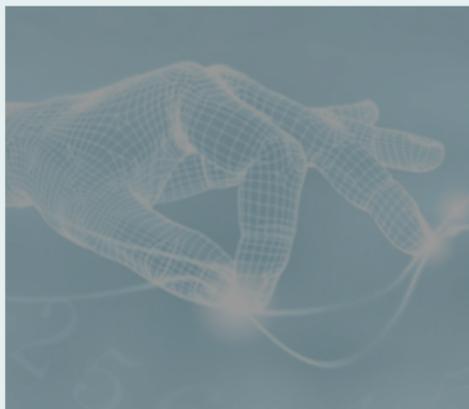


immagine dell'Esempio 2 rimodulata con $\alpha = 0.2$

OSSERVAZIONE IMPORTANTE

Tutte le trasformazioni viste, esclusa la sola “negazione”, NON sono completamente reversibili.

Schiarire un'immagine con un parametro δ e poi scurirla con lo stesso parametro NON restituisce necessariamente l'immagine di partenza.

Analogamente, rimodulare un'immagine con un fattore α e poi rimodularla nuovamente con il fattore inverso α^{-1} NON restituisce necessariamente l'immagine di partenza.

La causa di questo fatto risiede nell'utilizzo della funzione `uint8` al termine di ogni operazione, la quale riporta tutti i numeri a degli interi compresi nell'intervallo $[0, 255]$.



immagine originale



immagine prima schiarita e poi scurita con parametro $\delta = 100$



immagine prima scurita e poi schiarita con parametro $\delta = 100$

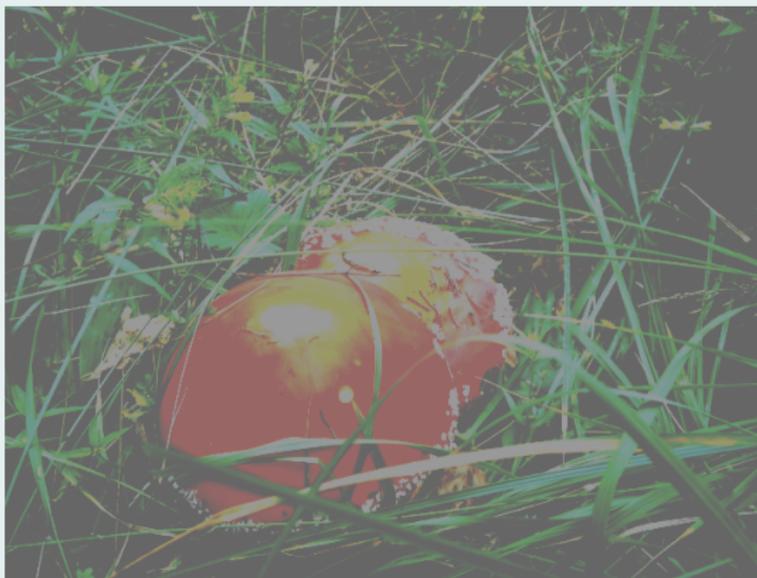


immagine con tonalità di colore prima dilatate del fattore $\alpha = 5$ e poi contratte del fattore inverso $\alpha^{-1} = 0.2$



immagine con tonalità di colore prima contratte del fattore $\alpha = 0.01$ e poi dilatate del fattore inverso $\alpha^{-1} = 100$

*In realtà la contrazione della tonalità dei colori è un'operazione che si riesce ad "invertire in modo approssimato" abbastanza bene fino ad un fattore $\alpha = 0.05$, in corrispondenza al quale l'immagine a sua volta rimodulata con il fattore inverso $\alpha^{-1} = 20$ presenta uno **scostamento uniforme relativo massimo inferiore al 10%** rispetto all'immagine originale (del tutto accettabile).*



Ingrandimenti e interpolazione bilineare

Le immagini degli Esempi 1.1 e 1.2 sono in realtà ingrandimenti delle effettive immagini rappresentate dalle rispettive tabelle numeriche (che avrebbero dimensione 10×10 , quindi molto piccole!).

Essi sono ottenuti sostituendo ciascun pixel con un quadrato di 60×60 pixel dello stesso colore e tonalità. Le immagini così ingrandite hanno dimensione 600×600 .

Esse presentano però una “sgranatura” piuttosto evidente.

*Una delle più semplici tecniche utilizzate per eliminare la sgranatura negli ingrandimenti consiste nel distanziare i pixel dell'immagine originale secondo il fattore di ingrandimento scelto K e, successivamente, nel colorare ciascun pixel intermedio che si è venuto a creare con colore e tonalità corrispondenti al valore numerico ottenuto per **interpolazione bilineare** dei valori numerici dei quattro pixel originali che lo circondano.*

La tabella numerica F (*matrice* $m \times n$) relativa all'immagine da ingrandire viene interpretata come rappresentativa del grafico di una funzione $f(y, x)$ delle variabili y e x , definita sul rettangolo $[1, m] \times [1, n]$ e di cui si conoscono i valori sulle coppie di interi $(i, j) \in [1, m] \times [1, n]$.

In altre parole, il valore numerico che compare al posto (i, j) nella matrice F si suppone essere

$$f_{i,j} = f(i, j)$$

Siccome le righe della matrice si numerano dall'alto verso il basso, fissiamo coerentemente il verso dell'asse y dall'alto verso il basso, contrariamente a quanto si fa comunemente per le coordinate cartesiane del piano.

Interpolare la funzione f in un punto (y, x) diverso da una coppia di interi (i, j) significa assegnarle in tale punto un valore sulla base dei valori noti $f_{i,j}$.

Ovviamente si cerca di fare in modo che il valore approssimato di $f(y, x)$ così trovato sia “vicino” a quello esatto.

Nel caso delle funzioni rappresentative di immagini il valore esatto è quello che si sarebbe ottenuto se si fosse memorizzata l'immagine stessa con una risoluzione maggiore, tale da far corrispondere un pixel al punto di coordinate (y, x) .

Come già anticipato, determiniamo quattro punti (i, j) , $(i + 1, j)$, $(i, j + 1)$, $(i + 1, j + 1)$ tali che (y, x) appartenga al quadrato di lato unitario da essi determinato.

Si ha che $i = \lfloor y \rfloor$ e $j = \lfloor x \rfloor$, per cui $y = i + \alpha$ e $x = j + \beta$ con $\alpha, \beta \in [0, 1)$.

	j			x	$j + 1$
i	$f_{i,j}$				$f_{i,j+1}$
y				?	
$i + 1$	$f_{i+1,j}$				$f_{i+1,j+1}$

esempio con $(y, x) = (i + 0.4, j + 0.8)$ e fattore di ingrandimento $K = 5$

Interpoliamo linearmente i valori $f_{i,j}$ e $f_{i,j+1}$ lungo la riga i -esima calcolando la loro **combinazione lineare convessa**

$$\tilde{f}(i, x) = (1 - \beta)f_{i,j} + \beta f_{i,j+1}$$

e, analogamente, i valori $f_{i+1,j}$ e $f_{i+1,j+1}$ lungo la riga $(i + 1)$ -esima calcolando

$$\tilde{f}(i + 1, x) = (1 - \beta)f_{i+1,j} + \beta f_{i+1,j+1}$$

	j			x	$j + 1$
i	$f_{i,j}$			$\tilde{f}(i, x)$	$f_{i,j+1}$
y				?	
$i + 1$	$f_{i+1,j}$			$\tilde{f}(i + 1, x)$	$f_{i+1,j+1}$

Poi interpoliamo linearmente i due valori così ottenuti lungo la colonna $(j + \beta)$ – esima definendo

$$\begin{aligned}\tilde{f}(y, x) &= (1 - \alpha)\tilde{f}(i, x) + \alpha\tilde{f}(i + 1, x) = \\ &= (1 - \alpha)(1 - \beta)f_{i,j} + (1 - \alpha)\beta f_{i,j+1} + \\ &\quad + \alpha(1 - \beta)f_{i+1,j} + \alpha\beta f_{i+1,j+1}\end{aligned}$$

Infine, siccome $\tilde{f}(y, x)$ in generale non è un valore intero, lo arrotondiamo all'intero più vicino e definiamo quindi

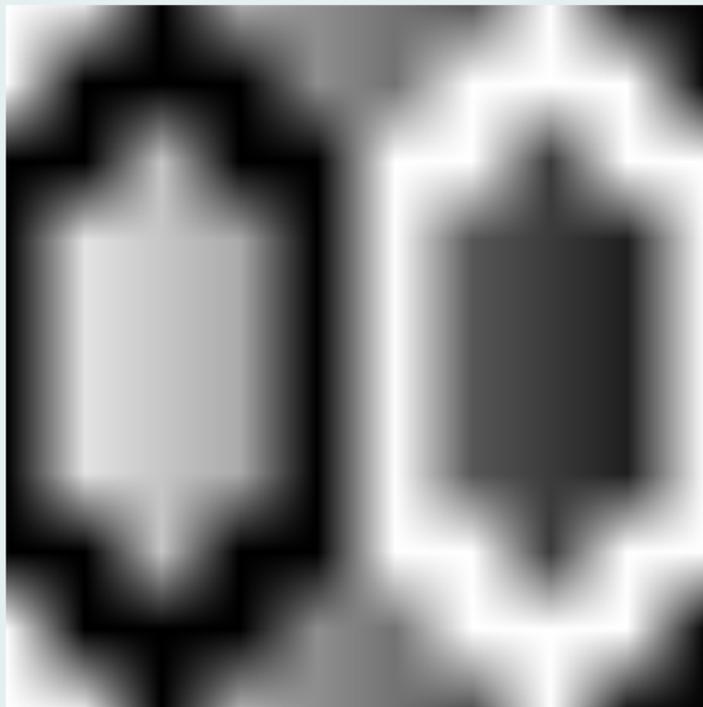
$$f(y, x) = \text{round}(\tilde{f}(y, x))$$

Per lo stesso motivo, arrotondiamo anche i valori sulle righe i -esima e $(i + 1)$ -esima definendo

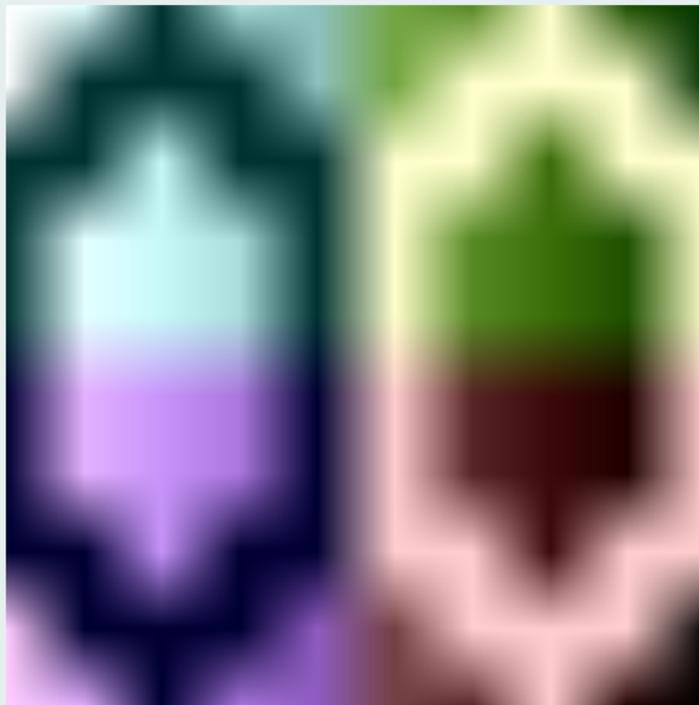
$$f(i, x) = \text{round}(\tilde{f}(i, x)) \quad \text{e} \quad f(i + 1, x) = \text{round}(\tilde{f}(i + 1, x))$$

Dato che $\tilde{f}(y, x)$ è “simmetrica” in α e β , si arriva al medesimo valore di $f(y, x)$ anche se si interpola prima sulle colonne j -esima e $(j + 1)$ -esima e poi sulla riga $(i + \alpha)$ -esima, producendo in maniera analoga i valori intermedi $f(y, j)$ e $f(y, j + 1)$.

	j				x	$j + 1$
i	$f_{i,j}$				$f(i, x)$	$f_{i,j+1}$
y	$f(y, j)$				$f(y, x)$	$f(y, j+1)$
$i + 1$	$f_{i+1,j}$				$f(i + 1, x)$	$f_{i+1,j+1}$



applicazione all'immagine in scala di grigi dell'Esempio 1.1 con fattore di ingrandimento $K = 60$



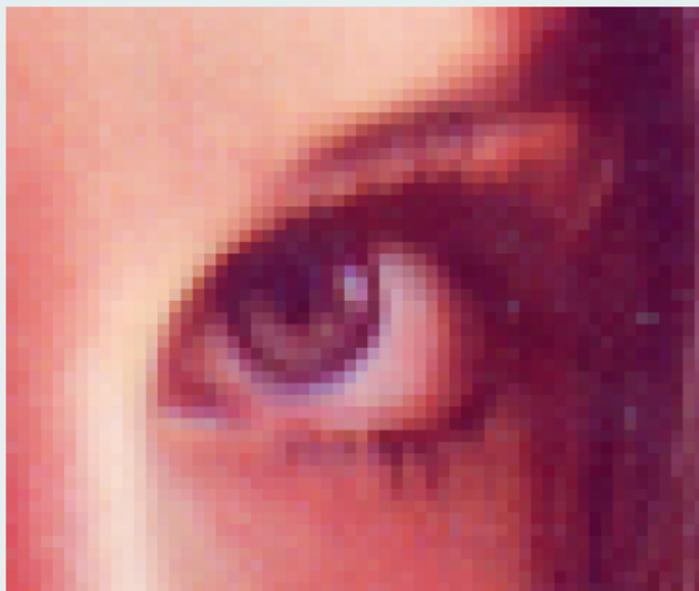
applicazione all'immagine a colori dell'Esempio 1.2 con fattore di ingrandimento $K = 60$

Esempio 4

Vediamo ora come si differenziano le due tecniche, quella del semplice ingrandimento di ciascun pixel e quella dell'interpolazione bilineare, su un'immagine ottenuta come dettaglio ritagliato da una fotografia.



immagine a grandezza originale (dimensione 50×60 pixel)



*immagine ottenuta ingrandendo ciascun pixel con fattore $K = 10$
(dimensione 500×600 pixel)*

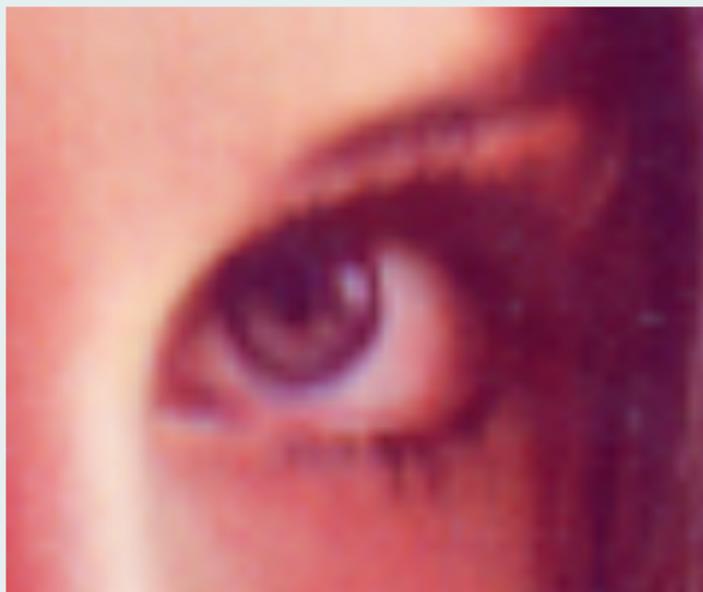


immagine ottenuta mediante interpolazione bilineare con fattore di ingrandimento $K = 10$ (dimensione 500×600 pixel)

Consideriamo ora il problema dell'individuazione dei contorni degli oggetti contenuti in un'immagine. Vogliamo, in altre parole, dare all'immagine le caratteristiche di un disegno eseguito con la matita nera, nel quale si evidenziano solamente i contorni e si trascurano le piccole variazioni di tonalità dei colori.

L'osservazione base che permette di realizzare quanto voluto è che i contorni corrispondono in linea di massima a quelle parti dell'immagine in cui si presentano brusche variazioni delle tonalità dei colori in almeno una delle due direzioni orizzontale e verticale.

*Considerando nuovamente i colori come funzioni del tipo $f(y, x)$, i punti dei contorni sono dunque quei punti (i, j) per i quali almeno una delle due **derivate parziali** $\frac{\partial f}{\partial y}$ e $\frac{\partial f}{\partial x}$ risulta "grande" in valore assoluto.*

La derivata parziale di f rispetto a y nel punto (y_0, x_0) è data da

$$\frac{\partial f}{\partial y}(y_0, x_0) = \lim_{\Delta y \rightarrow 0} \frac{f(y_0 + \Delta y, x_0) - f(y_0, x_0)}{\Delta y}$$

Analogamente, per la derivata parziale di f rispetto a x , si ha

$$\frac{\partial f}{\partial x}(y_0, x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(y_0, x_0 + \Delta x) - f(y_0, x_0)}{\Delta x}$$

In realtà le derivate delle funzioni che modellano le immagini presentano spesso delle discontinuità in corrispondenza dei punti di contorno degli oggetti e, quindi, è più coerente considerare solamente le **derivate sinistre**, convenendo di ricercare i salti nelle tonalità dei colori procedendo dall'alto verso il basso e da sinistra verso destra.

Consideriamo pertanto

$$\frac{\partial f}{\partial y}(y_0, x_0)_- = \lim_{\substack{\Delta y \rightarrow 0 \\ \Delta y < 0}} \frac{f(y_0 + \Delta y, x_0) - f(y_0, x_0)}{\Delta y}$$

$$\frac{\partial f}{\partial x}(y_0, x_0)_- = \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta x < 0}} \frac{f(y_0, x_0 + \Delta x) - f(y_0, x_0)}{\Delta x}$$

Algoritmo di individuazione dei contorni

Approssimiamo $\frac{\partial f}{\partial y}$ e $\frac{\partial f}{\partial x}$ nei punti $(i, j) \in [1, m] \times [1, n]$ come segue:

$$\frac{\partial f}{\partial y}(i, j)_- \approx f(i, j) - f(i, j-1) = f_{i,j} - f_{i,j-1} \quad (\text{con } \Delta y = -1)$$

$$\frac{\partial f}{\partial x}(i, j)_- \approx f(i, j) - f(i-1, j) = f_{i,j} - f_{i-1,j} \quad (\text{con } \Delta x = -1)$$

Fissiamo una **soglia** σ oltre la quale considerare il valore delle derivate (in valore assoluto) indicativo della presenza di un contorno e definiamo la matrice C i cui elementi sono

$$c_{i,j} = \begin{cases} 0 & \text{se } |f_{i,j} - f_{i-1,j}| > \sigma \vee |f_{i,j} - f_{i,j-1}| > \sigma \\ 255 & \text{altrimenti} \end{cases}$$

In particolare, poniamo $C(1, 1) = 255$ (assenza di variazione di tonalità in entrambe le direzioni).

Se scegliamo una soglia $\sigma \leq -1$ otteniamo chiaramente una matrice C di elementi tutti uguali a 0 (tranne $C(1,1) = 255$), quindi un'immagine tutta nera.

Se invece scegliamo una soglia $\sigma \geq 255$ otteniamo chiaramente una matrice C di elementi tutti uguali a 255, quindi un'immagine tutta bianca.

Ha quindi senso scegliere $0 \leq \sigma \leq 254$.

In tale intervallo, tanto più piccolo si sceglie σ , tanto più numerosi sono i punti interpretati come facenti parte del contorno di qualche oggetto.

In generale non è facile individuare il valore di σ che determini esattamente gli effettivi contorni dei vari oggetti contenuti nell'immagine.

Per un'immagine a colori si può ripetere la medesima procedura su tutte e tre le componenti della terna (F_1, F_2, F_3) che la rappresenta.

Si ottiene così una specie di disegno eseguito con dei pastelli colorati in cui, ancora una volta, si tende ad evidenziare solamente i contorni e si trascurano le piccole variazioni di tonalità dei colori.

Esempio 5.1 (scala di grigi)

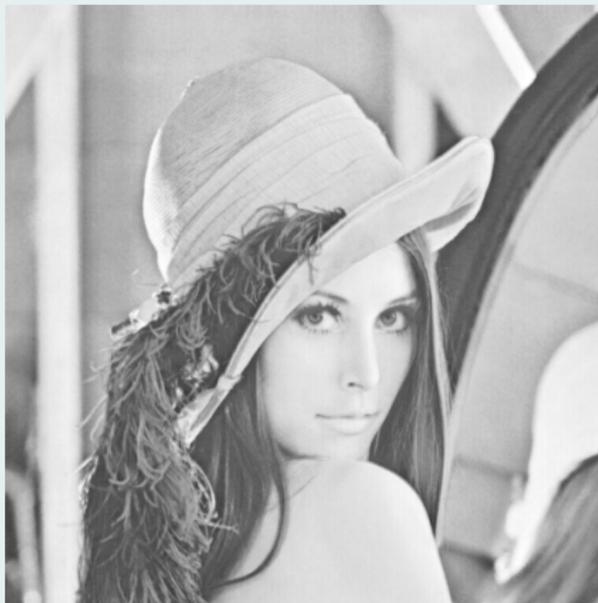


immagine originale (ritratto di Lena Söderberg, molto utilizzato già da qualche decennio per testare gli algoritmi di elaborazione delle immagini)



immagine trattata con soglia $\sigma = 5$



immagine trattata con soglia $\sigma = 10$



immagine trattata con soglia $\sigma = 15$



immagine trattata con soglia $\sigma = 20$

Esempio 5.2 (a colori)



immagine originale

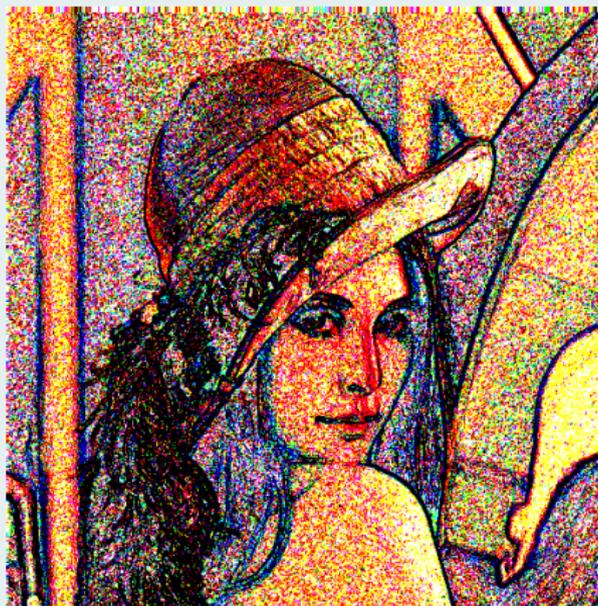


immagine trattata con soglia $\sigma = 5$

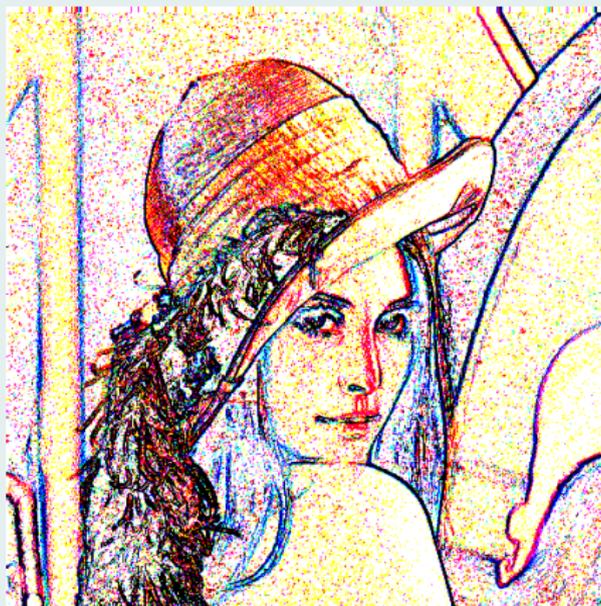


immagine trattata con soglia $\sigma = 10$



immagine trattata con soglia $\sigma = 15$



immagine trattata con soglia $\sigma = 20$

Un'applicazione curiosa: le "immagini derivate"

Definiamo ora una matrice Df_x in cui assegniamo a ciascun elemento di posto (i, j) il valore della derivata parziale sinistra $\frac{\partial f}{\partial x}(i, j)_-$ della funzione $f(y, x)$ che rappresenta un'immagine data (di matrice F).

Nel concreto utilizziamo l'approssimazione che abbiamo già visto, adottando però per un "fattore di amplificazione" $\alpha \geq 1$ al fine di dare più o meno vivacità all'effetto visivo.

Inoltre, per tener conto del fatto che le derivate sono sia positive che negative, facciamo corrispondere la derivata nulla (cioè l'assenza di variazione di tonalità) al valore medio 127.5.

Quindi, per $i = 1, \dots, m$, poniamo

$$Df_{x_{i,1}} = 128$$

e inoltre, per $j = 2, \dots, n$,

$$Df_{x_{i,j}} = \text{uint8}(\alpha(f_{i,j} - f_{i,j-1}) + 127.5)$$

Osserviamo che, per ogni $\alpha \geq 128$, $Df_{x_{i,j}}$ assume sempre lo stesso valore tra uno dei seguenti tre valori: 0, 128, 255. Ha senso pertanto considerare $1 \leq \alpha \leq 128$.

La procedura proposta consiste nel misurare le variazioni di tonalità dei colori scannerizzando l'immagine da sinistra verso destra (lungo le righe della matrice rappresentativa F).

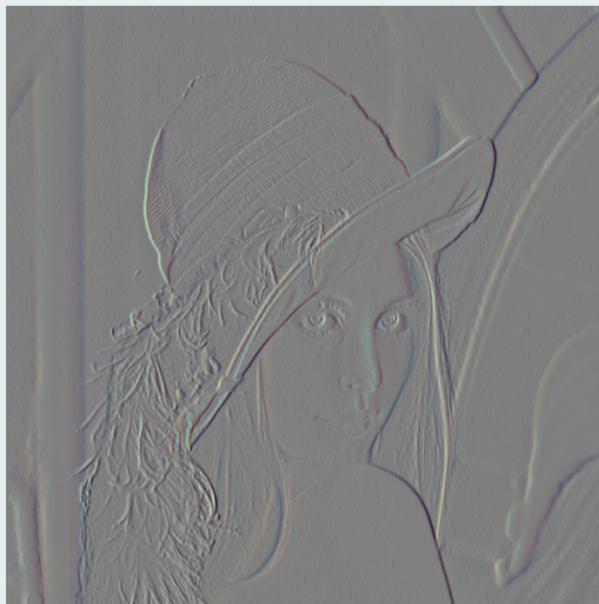
Ovviamente possiamo ripeterla scannerizzando l'immagine dall'alto verso il basso (lungo le colonne di F), definendo una matrice DF_y in cui si assegni a ciascun elemento di posto (i, j) il valore della derivata parziale sinistra $\frac{\partial f}{\partial y}(i, j)_-$.

Nel concreto, analogamente a quanto fatto prima, per $j = 1, \dots, m$, poniamo

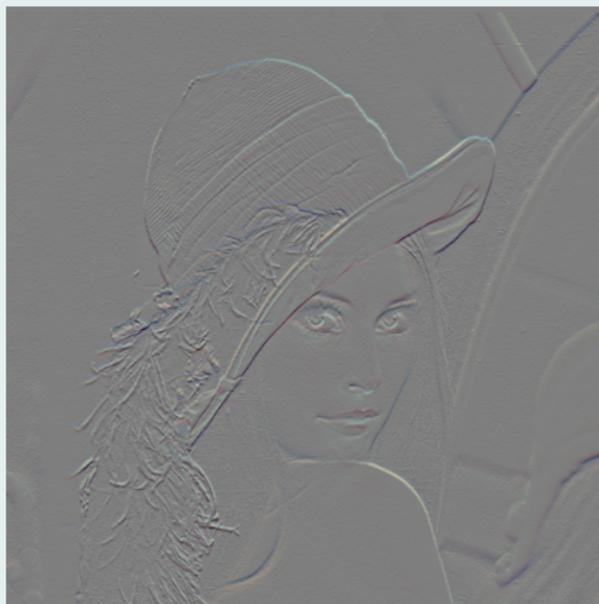
$$Df_{y_{1,j}} = 128$$

e inoltre, per $i = 2, \dots, n$,

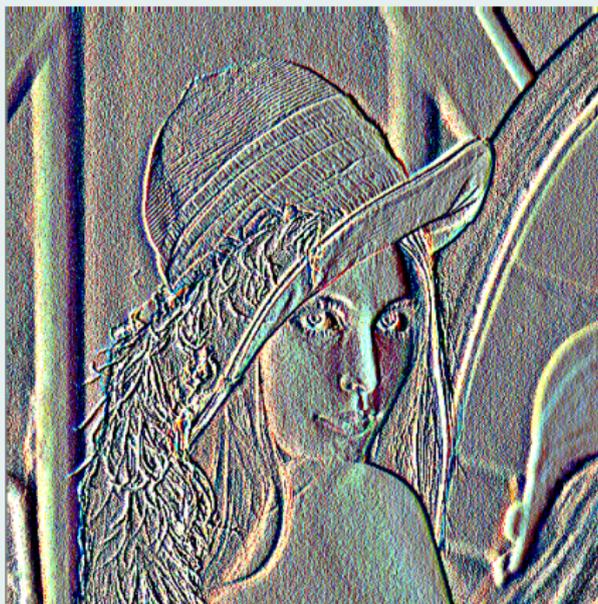
$$Df_{y_{i,j}} = \text{uint8}(\alpha(f_{i,j} - f_{i-1,j}) + 127.5)$$



“derivata lungo le righe” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 1$



“derivata lungo le colonne” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 1$



“derivata lungo le righe” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 25$



“derivata lungo le colonne” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 25$



“derivata lungo le righe” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 128$



“derivata lungo le colonne” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 128$

Una “scelta mediata”

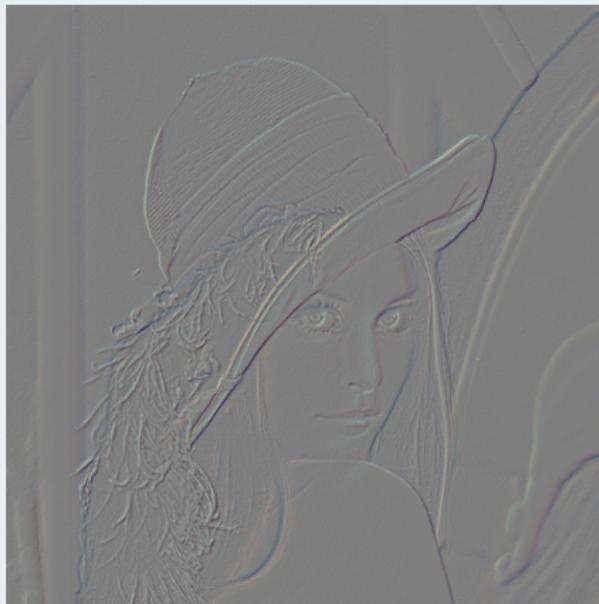
Appaiono abbastanza evidenti le differenze tra le “derivate lungo le righe” e le “derivate lungo le colonne”.

Infatti la scannerizzazione lungo le righe “non vede” i salti di colore/tonalità orizzontali e, analogamente, quella lungo le colonne “non vede” i salti di colore/tonalità verticali.

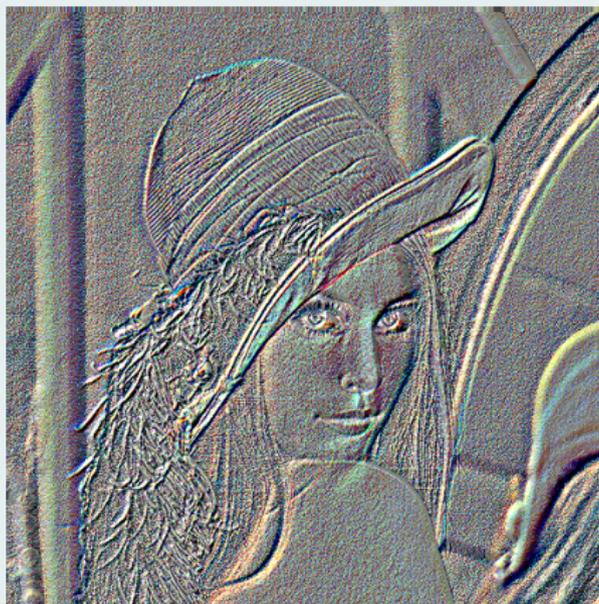
*Una soluzione abbastanza soddisfacente a questo problema può essere quella di considerare l'immagine che si ottiene considerando la **media DF** delle due tabelle DF_x e DF_y , ottenuta ponendo*

$$df_{i,j} = \text{round}((df_{x,i,j} + df_{y,i,j})/2)$$

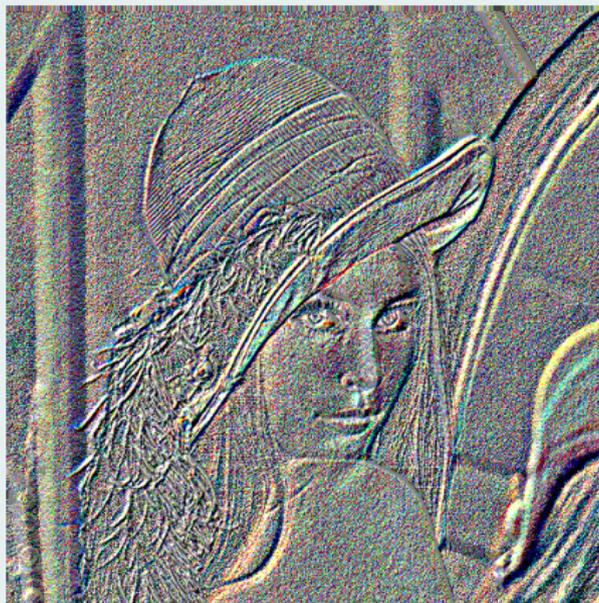
per $i = 1, \dots, m$ e $j = 1, \dots, n$.



“derivata media” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 1$



“derivata media” dell’immagine dell’Esempio 5.2 con fattore di amplificazione $\alpha = 25$



"derivata media" dell'immagine dell'Esempio 5.2 con fattore di amplificazione $\alpha = 128$

“Fusione” di immagini e combinazioni lineari convesse

Supponiamo ora di avere due immagini di uguale altezza e di volerle fondere in un'unica immagine ponendole una a fianco dell'altra, eventualmente con una parziale sovrapposizione.

Il modo più ovvio è quello di unificare opportunamente le loro matrici rappresentative (o terne di matrici se le immagini sono a colori), che hanno lo stesso numero m di righe, ottenendo così un'unica matrice (o terna di matrici) il cui numero di colonne n è minore o uguale alla somma $n_1 + n_2$ dei numeri di colonne delle due singole matrici.

In generale, il risultato visivo che si ottiene non è molto soddisfacente in quanto si nota un'evidente discontinuità in corrispondenza della separazione tra le due immagini originali.



immagine n.1



immagine n.2



immagine ottenuta affiancando due parti delle immagini precedenti

Per eliminare le evidenti discontinuità, o almeno attenuarle, individuiamo una “zona di transizione” tra le due immagini al modo seguente.

Chiamiamo F la matrice rappresentativa dell'immagine n.1, G quella dell'immagine n.2 e H quella dell'immagine che si ottiene come fusione delle due immagini.

Conveniamo che la colonna j_1 -esima di H sia l'ultima di quelle (da sinistra verso destra) che contengono esclusivamente una parte di F (immagine n.1) e la colonna j_2 -esima di H sia la prima di quelle che contengono esclusivamente una parte di G (immagine n.2).

Per le colonne di indice j con $j_1 < j < j_2$ calcoliamo invece la **combinazione lineare convessa** delle colonne corrispondenti di F e G , con **pesi** proporzionali alle distanze lungo le righe dalle colonne di indice j_1 e j_2 . Cioè, per $i = 1, \dots, m$, poniamo:

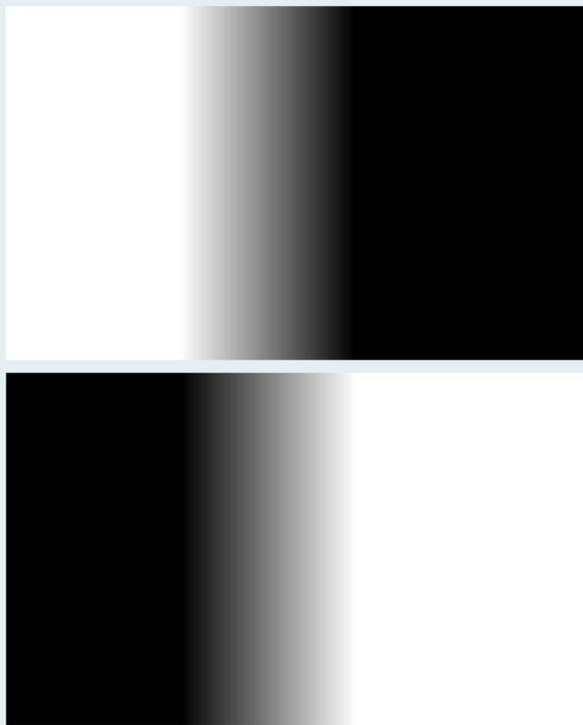
$$h_{i,j} = \begin{cases} f_{i,j} & \text{se } j \leq j_1 \\ \text{round} \left(\frac{j_2-j}{j_2-j_1} f_{i,j} + \frac{j-j_1}{j_2-j_1} g_{i,j} \right) & \text{se } j_1 < j < j_2 \\ g_{i,j} & \text{se } j \geq j_2 \end{cases}$$



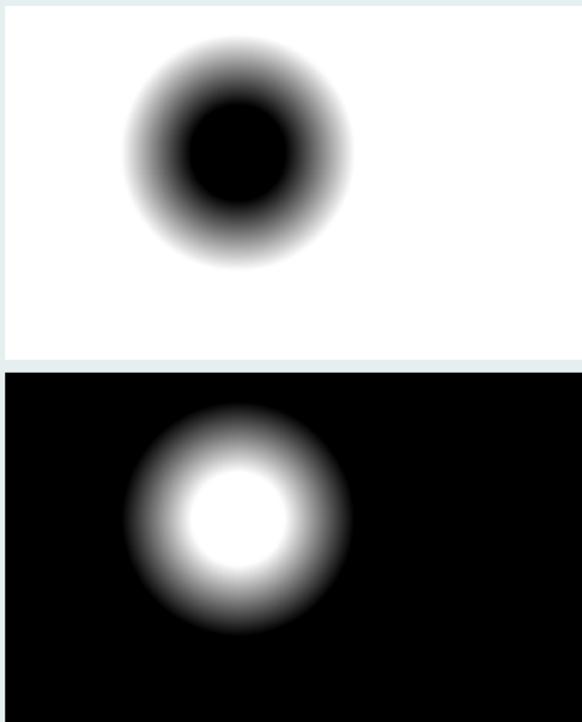
immagine ottenuta affiancando due parti delle immagini precedenti con "zona di transizione" larga $j_2 - j_1 = 30$ pixel a cavallo della discontinuità

Fusioni di immagini di tipo più complicato

Nell'esempio precedente abbiamo utilizzato una coppia di "filtri" (uno opposto dell'altro) che si possono visualizzare come segue:



Possiamo eseguire molti altri tipi di "fusioni": ad esempio, inserire una parte circolare dell'immagine n.2 all'interno dell'immagine n.1 con i "filtri"



Le formule corrispondenti sono ovviamente più complicate in quanto bisogna definire una zona di transizione a forma di “corona circolare” nella quale il peso di ciascuna delle due immagini varia linearmente da 0 a 1 lungo i raggi dei due cerchi concentrici che la delimitano.

Conveniamo per semplicità che le due immagini (di matrici rappresentative F e G) abbiano le stesse dimensioni m e n . Indichiamo con (I, J) le coordinate del pixel che funge da **centro** dei due cerchi concentrici, scegliamo i loro raggi $r_1 < r_2$ e, per ogni $(i, j) \in [1, m] \times [1, n]$, calcoliamo la distanza

$$\delta_{i,j} = \sqrt{(i - I)^2 + (j - J)^2}$$

tra il centro dei cerchi ed il corrispondente pixel.

La matrice H della nuova immagine viene così definita:

$$h_{i,j} = \begin{cases} f_{i,j} & \text{se } \delta_{i,j} \geq r_2 \\ \text{round} \left(\frac{\delta_{i,j} - r_1}{r_2 - r_1} f_{i,j} + \frac{r_2 - \delta_{i,j}}{r_2 - r_1} g_{i,j} \right) & \text{se } r_1 < \delta_{i,j} < r_2 \\ g_{i,j} & \text{se } \delta_{i,j} \leq r_1 \end{cases}$$



immagine n.1



immagine n.2 ("negativo" dell'immagine n.1)



*inserimento di una parte circolare dell'immagine n.2 nell'immagine n.1
senza "zona di transizione"*



inserimento di una parte circolare dell'immagine n.2 nell'immagine n.1 con "zona di transizione" larga $r_2 - r_1 = 50$ pixel a cavallo della discontinuità

Combinazioni lineari convesse, equazioni lineari e “immagini nascoste”

Terminiamo la rassegna con una sorta di “trucco matematico” che ci permette di “nascondere” un’immagine (immagine n.2 rappresentata da una terna di matrici G) dentro un’altra immagine (immagine n.1 rappresentata da una terna di matrici F) utilizzando due strumenti già visti: la *rimodulazione dei colori* e la *combinazione lineare convessa*.

L’idea consiste nel decidere un fattore $\alpha > 0$ abbastanza “piccolo” (diciamo, ad esempio, 0.05) e nel definire la terna di matrici

$$H = \text{round}\left((1 - \alpha)F + \alpha G\right)$$

L’immagine corrispondente ad H (immagine n.3) è molto simile all’immagine n.1 e non si intravede l’immagine n.2 in essa “nascosta”.

Più piccolo è α , meglio si nasconde l’immagine n.2.



immagine n.1



immagine n.2 (la stessa dell'Esempio 3)



*immagine n.3 (immagine n.1 contenente l'immagine n.2 "nascosta" con
fattore $\alpha = 0.05$)*

Per “estrarre” l’immagine n.2 dall’immagine n.3 basta interpretare la formula precedente come un’*equazione nell’incognita G* e risolverla (trascurando gli arrotondamenti agli interi nel calcolo di H).
Bisogna però disporre anche dell’immagine n.1 e conoscere il fattore α .

Un’approssimazione G' di G è data quindi dalla formula

$$G' = \text{uint8}\left(\alpha^{-1}(H - (1 - \alpha)F)\right)$$

L’equazione per G si può anche riscrivere nella forma equivalente

$$\alpha(G - 127.5) + 127.5 = H - (1 - \alpha)(F - 127.5)$$

La matrice al membro destro rappresenta dunque l’immagine n.2 rimodulata con il fattore α .

Pertanto l’immagine n.2 “estratta” (rappresentata da G') è quella che si ottiene operando la rimodulazione inversa con il fattore α^{-1} .

Più piccolo è α , peggiore è la qualità dell’immagine “estratta”.



immagine estratta dall'immagine n.3 con fattore $\alpha = 0.05$

Anche se utilizziamo per l'estrazione valori di α moderatamente diversi da 0.05, i risultati ottenuti continuano ad essere accettabili.
Quindi la conoscenza precisa di α non è fondamentale.



immagine estratta dall'immagine n.3 con fattore $\alpha = 0.04$



immagine estratta dall'immagine n.3 con fattore $\alpha = 0.03$



immagine estratta dall'immagine n.3 con fattore $\alpha = 0.06$



immagine estratta dall'immagine n.3 con fattore $\alpha = 0.07$

Grazie per l'attenzione!