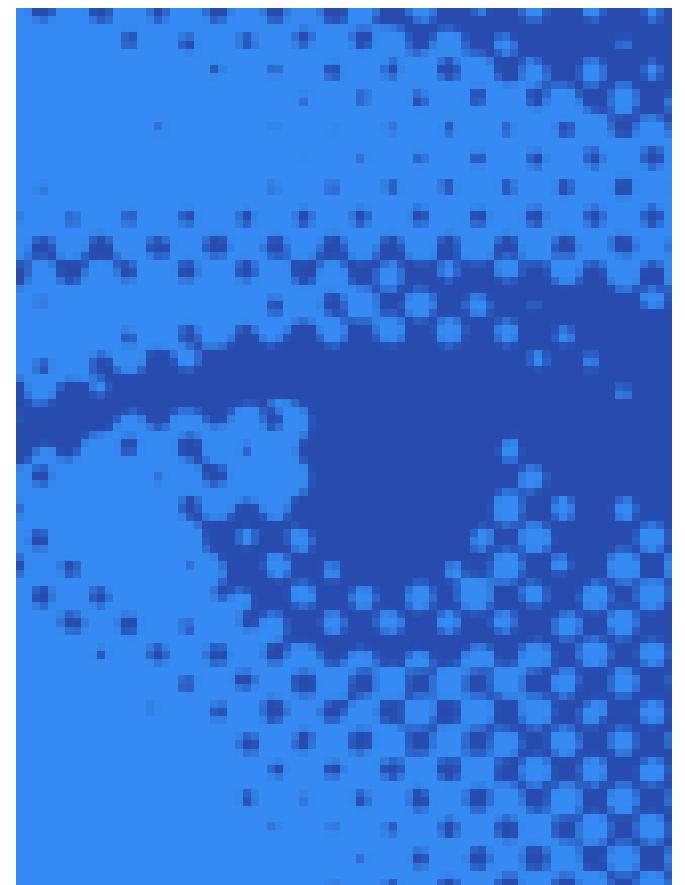


Analisi di Immagini Digitali

**SPEGNETE I
CELLULARI!!!**

Programma (in ordine sparso)

- Campionamento e quantizzazione
- Istogrammi
- Formati grafici
- Morfologia matematica
- Filtri
- Strutture piramidali
- Segmentazione
- Trasformate DCT / wavelet
- Compressione
- Quello che riusciremo ad aggiungere... :o



“Le Bibbie”

- *R.C.Gonzales, R.E.Woods*
Elaborazione delle Immagini Digitali
Pearson – Prentice Hall, 2008
testo generico
- *R.C.Gonzales, R.E.Woods, S.L.Eddins*
Digital Image Processing using Matlab
Prentice Hall, 2004
esercizi svolti
- *A.S.Glassner*
Principles of Digital Image Synthesis
Morgan Kaufmann Publishers, 1995
spazio delle frequenze
- *L.G.Shapiro, G.C.Stockman*
Computer Vision
Prentice Hall, 2001
testo generico
- *P.Soille*
Morphological Image Analysis
Springer-Verlag, 2003
morfologia matematica

Computer vision

L'analisi automatica delle immagini è un settore dell'intelligenza artificiale che fornisce una descrizione automatica di una scena del mondo reale. Esistono due approcci a questo problema:

- antropomorfo
- pragmatico

analisi preliminare
caratteristiche fondamentali
segmentazione
classificazione
analisi strutturale
interpretazione

Sistema visivo

La luminanza è il flusso luminoso, misurato in cd/m^2 , emesso da una superficie di 1 m^2 entro un angolo solido di 1 sr in direzione perpendicolare alla superficie.

Sole	$10^9 \text{ cd}/\text{m}^2$
Faro abbagliante di automobile	$10^7 \text{ cd}/\text{m}^2$
Strada a mezzogiorno	$10^5 \text{ cd}/\text{m}^2$
Cielo diurno	$10^4 - 10^6 \text{ cd}/\text{m}^2$
Lampada fluorescente	$10^3 - 10^5 \text{ cd}/\text{m}^2$
Luna piena	$10^3 - 10^4 \text{ cd}/\text{m}^2$
Visione fotopica (percezione colori)	$10 \text{ cd}/\text{m}^2$
Visione scotopica (percezione luminosità)	$0.01 \text{ cd}/\text{m}^2$
Illuminazione stradale	$1 \text{ cd}/\text{m}^2$
Cielo notturno con luna piena	$0.01 \text{ cd}/\text{m}^2$
Cielo notturno senza luna	$10^{-6} - 10^{-3} \text{ cd}/\text{m}^2$

Il lumen misura il flusso luminoso emesso in tutte le direzioni. Con differente angolo solido, il flusso rimane inalterato mentre le candele variano.

Sistema visivo

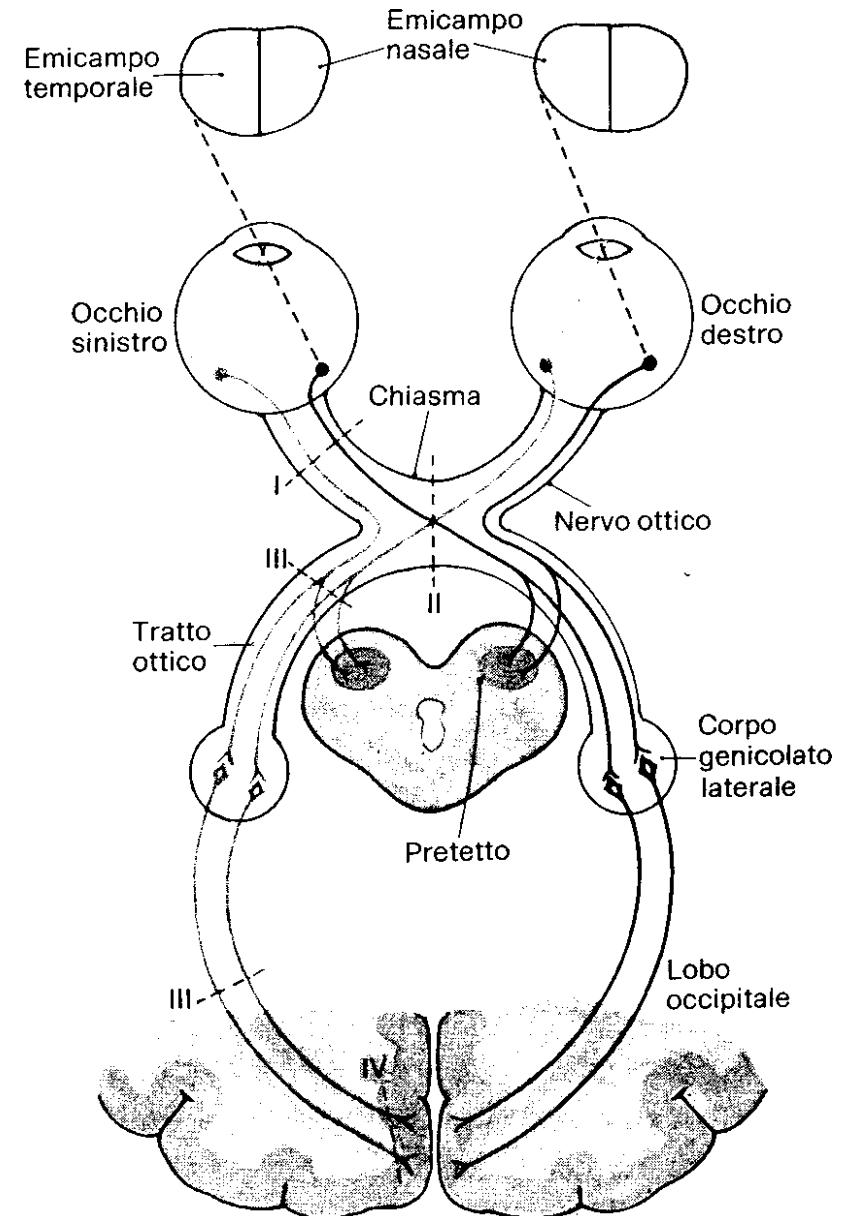
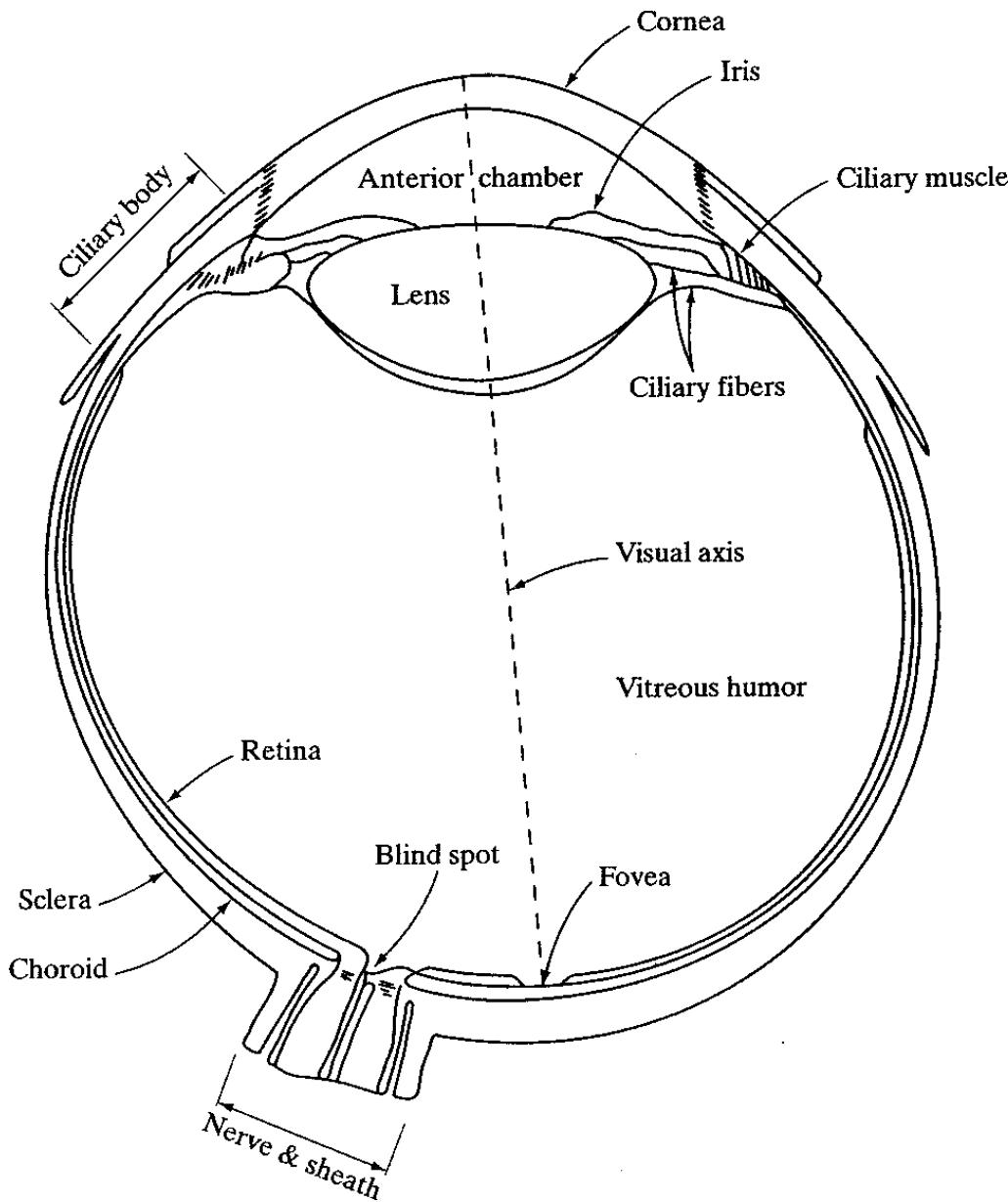


Immagine digitale

Un'immagine digitale D è una funzione definita sullo spazio discreto bidimensionale i cui valori discreti sono detti livelli di grigio g .

$$D = \{(i,j,g) : i \in \{0, \dots, w-1\}, j \in \{0, \dots, h-1\}, g \in \{0, \dots, G-1\}\}$$

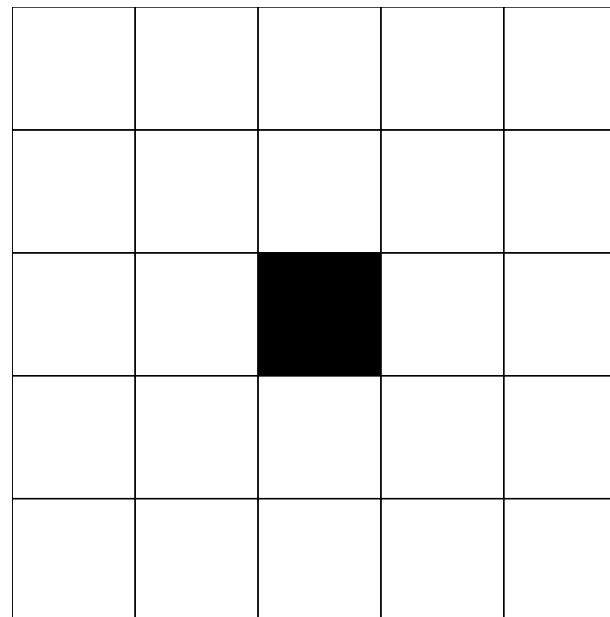


Immagine digitale

La quantità d'informazione necessaria per rappresentare un'immagine digitale dipende dalla precisione con cui si codificano le coordinate spaziali e le intensità luminose.

$$i = \min \{ \lfloor w \times (x - x_{\min}) / (x_{\max} - x_{\min}) \rfloor, w-1 \} \quad (\text{campionamento})$$

$$j = \min \{ \lfloor h \times (y - y_{\min}) / (y_{\max} - y_{\min}) \rfloor, h-1 \} \quad (\text{campionamento})$$

$$g = \min \{ \lfloor G \times (l - l_{\min}) / (l_{\max} - l_{\min}) \rfloor, G-1 \} \quad (\text{quantizzazione})$$

Sia Δ la minima variazione apprezzabile di luminosità. Allora il numero totale di livelli di grigio è dato da:

$$G = \lceil (l_{\max} - l_{\min}) / \Delta \rceil$$

Teorema del campionamento

Assumendo che l'immagine ammetta frequenze massime ν_x e ν_y , allora può essere completamente determinata da ampiezza campionate ad intervalli spaziali dati da:

$$\Delta_x = \frac{1}{2\nu_x}, \Delta_y = \frac{1}{2\nu_y}$$

Per ottenere pixel quadrati, basta imporre:

$$\Delta = \min\{\Delta_x, \Delta_y\}.$$

Teorema del campionamento

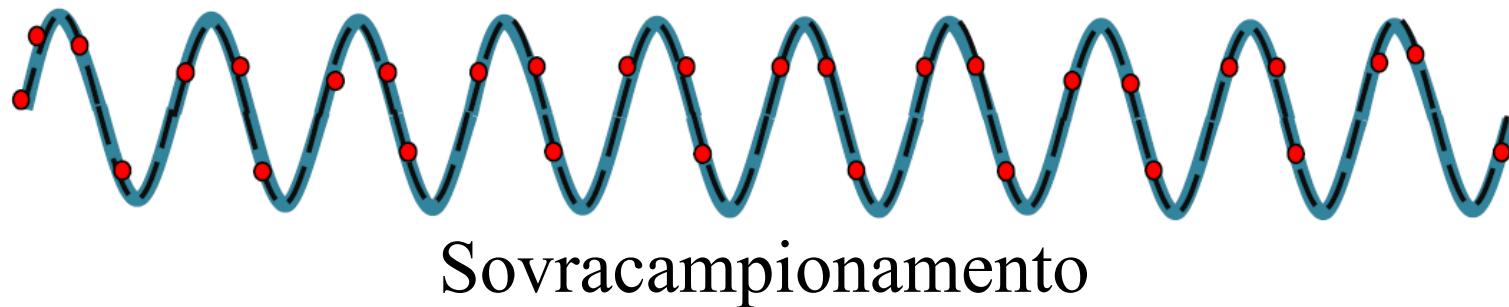
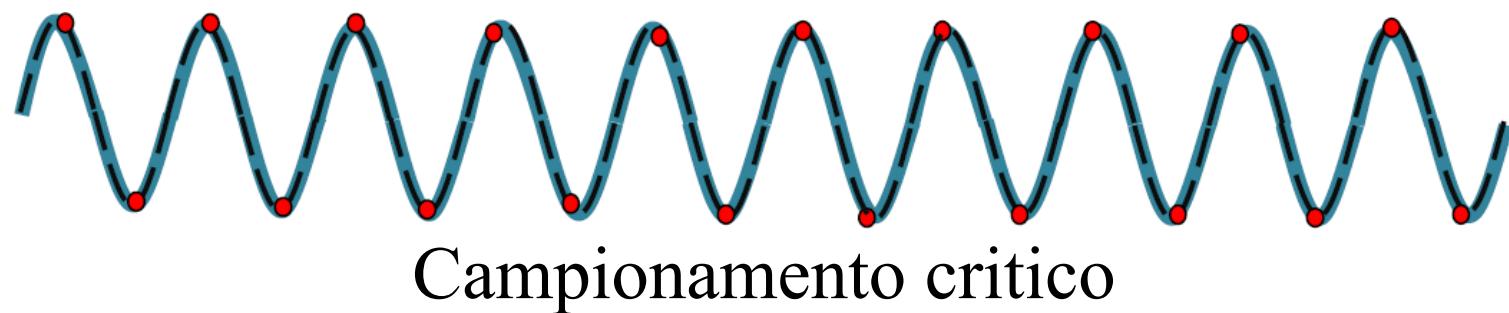
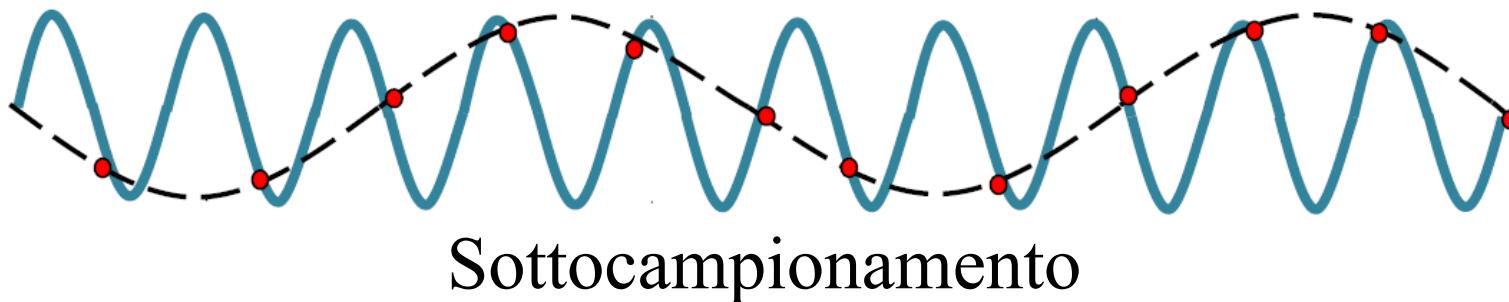


Immagine digitale

Intorno forte (distanza “city block” $d_4=d_x+d_y$)

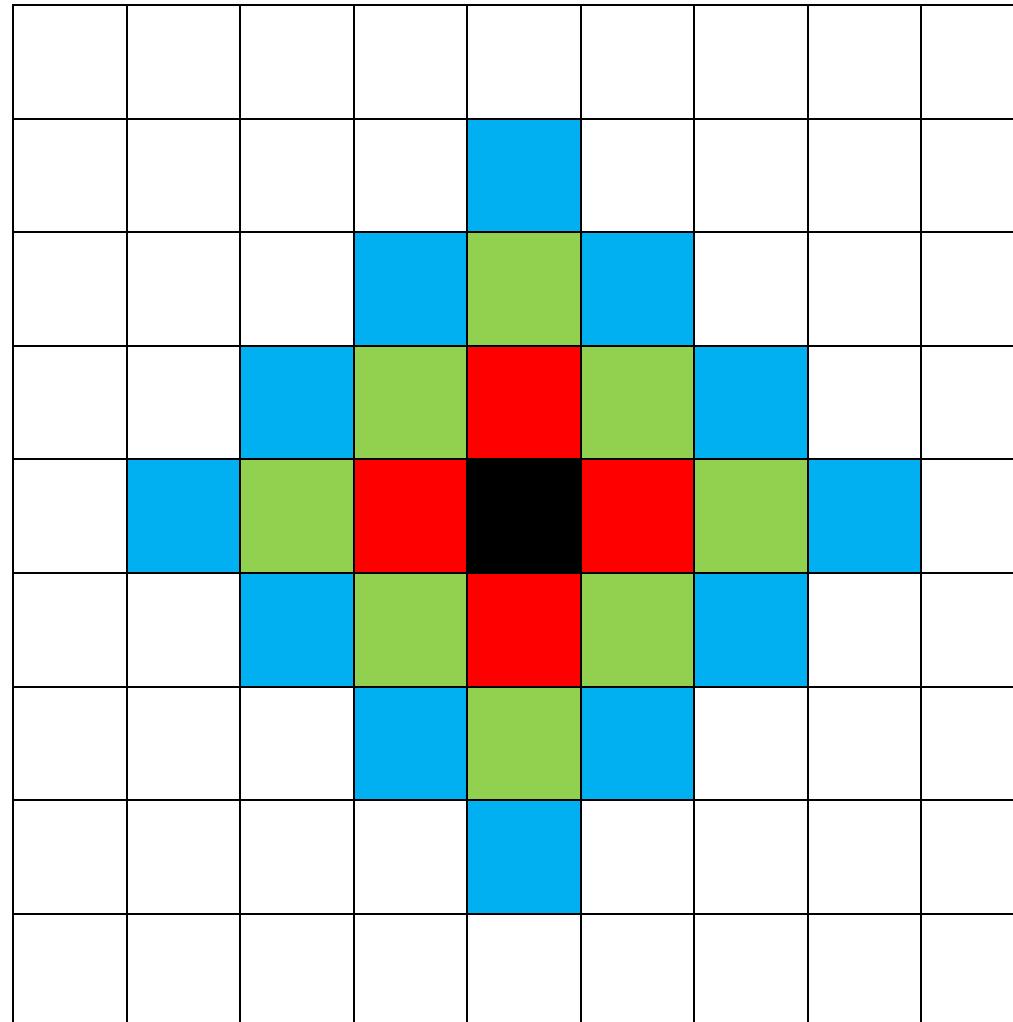


Immagine digitale

Intorno debole (distanza “chessboard” $d_8=\max\{d_x, d_y\}$)

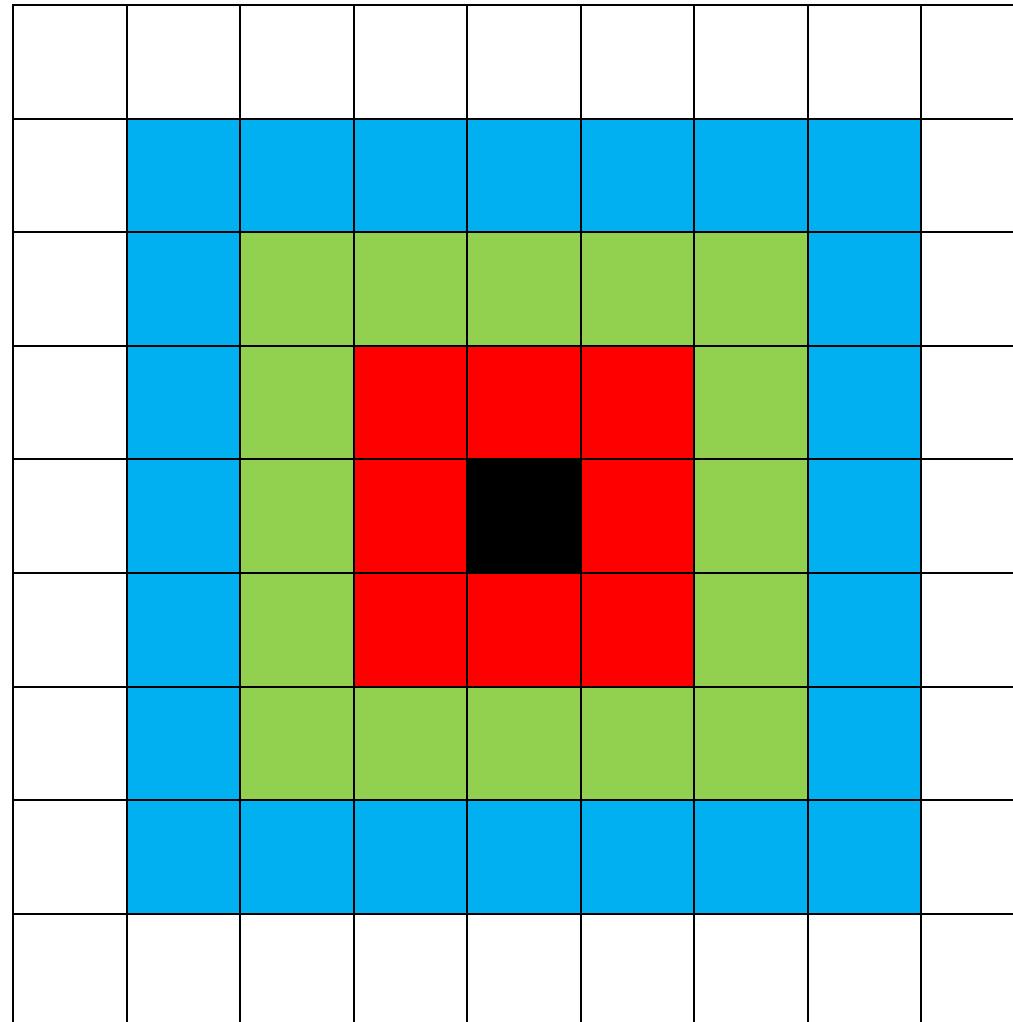
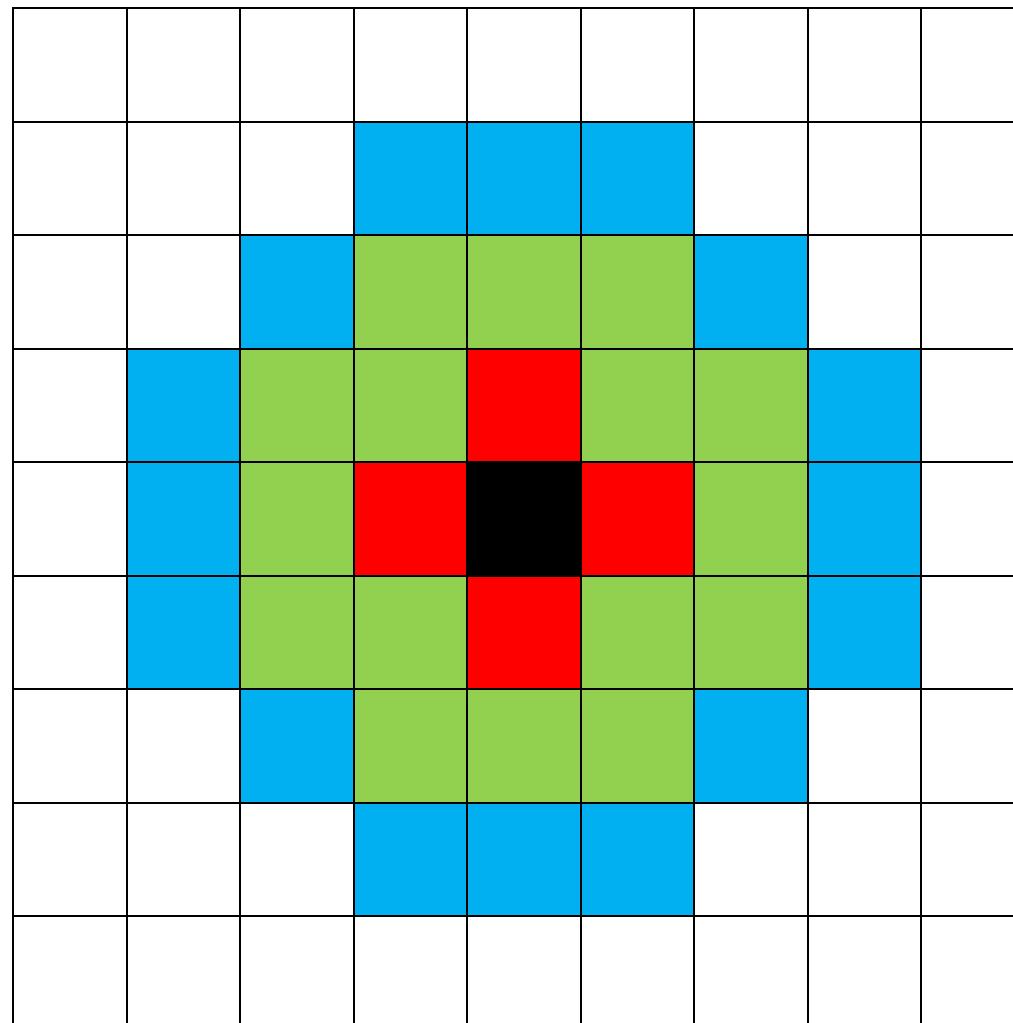
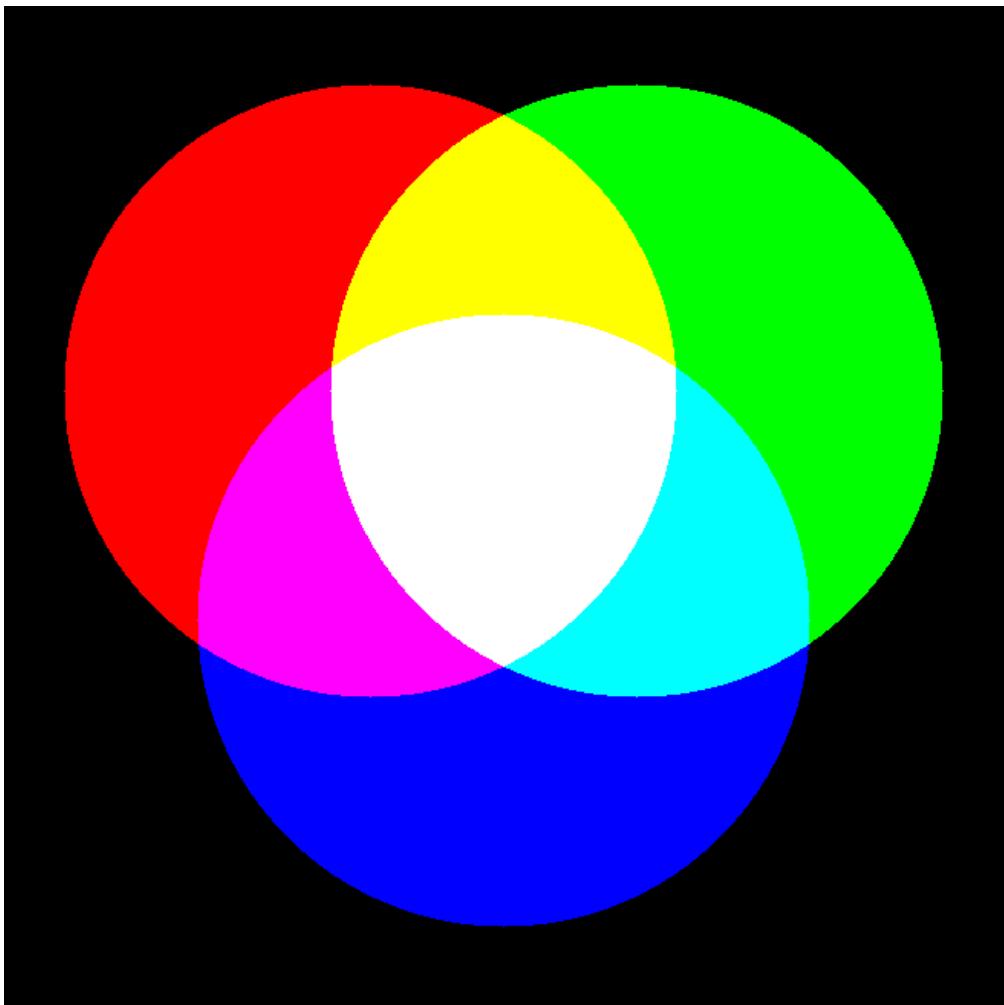


Immagine digitale

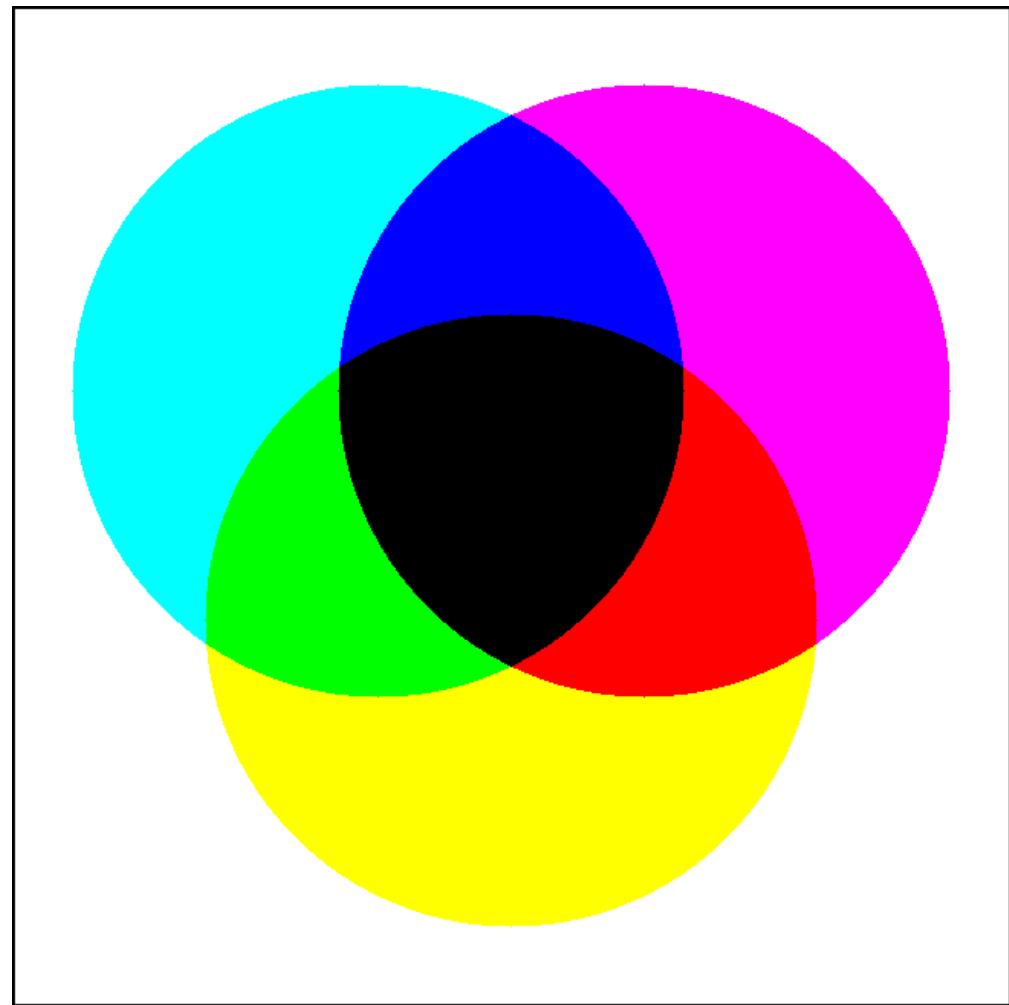
Intorno pseudoeuclideo (distanza “ottagonale” $d_0 = \max \{d_8, \lfloor 2(d_4+1)/3 \rfloor \}$)



Un mondo a colori...

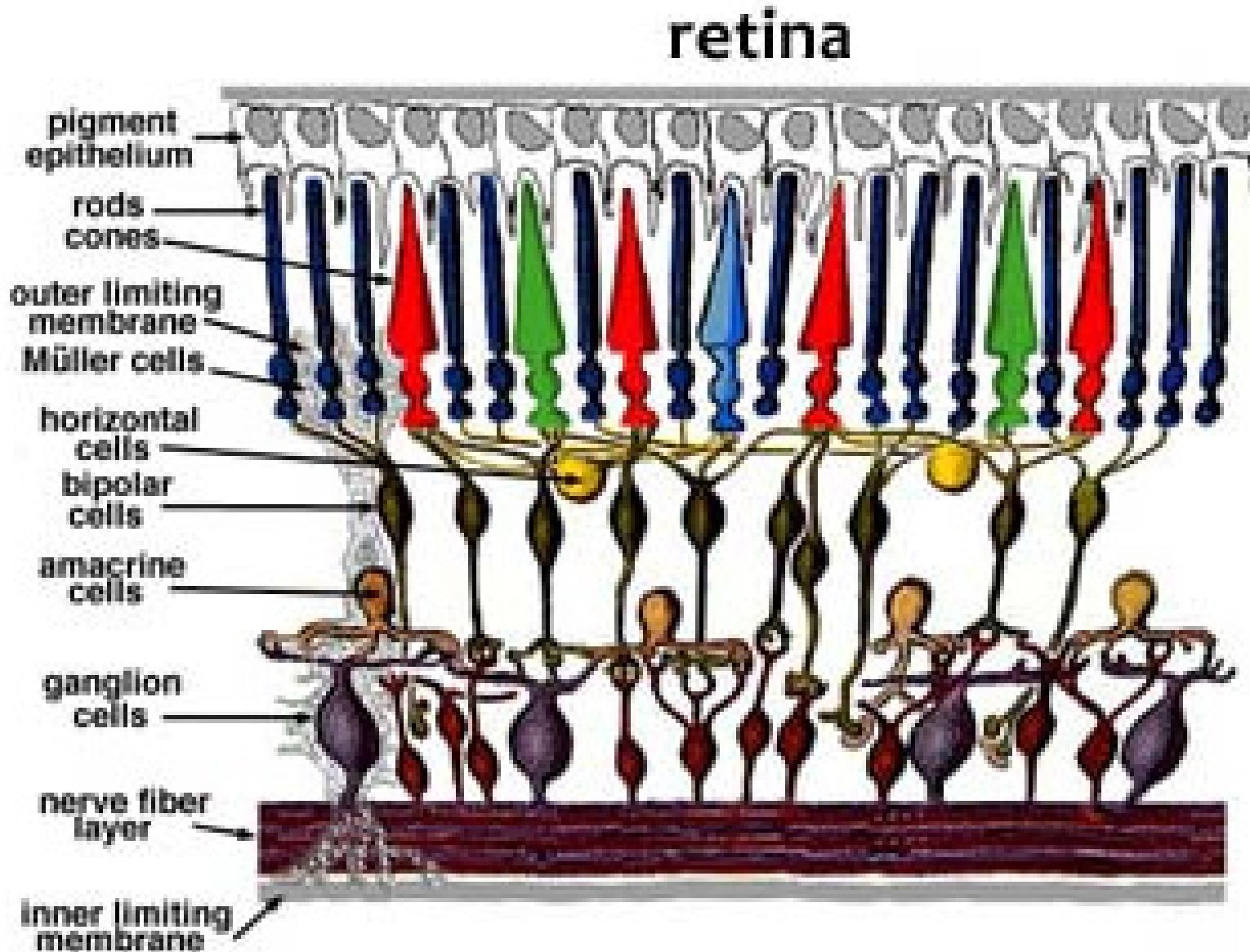


Sintesi additiva RGB
(ad esempio, monitor)

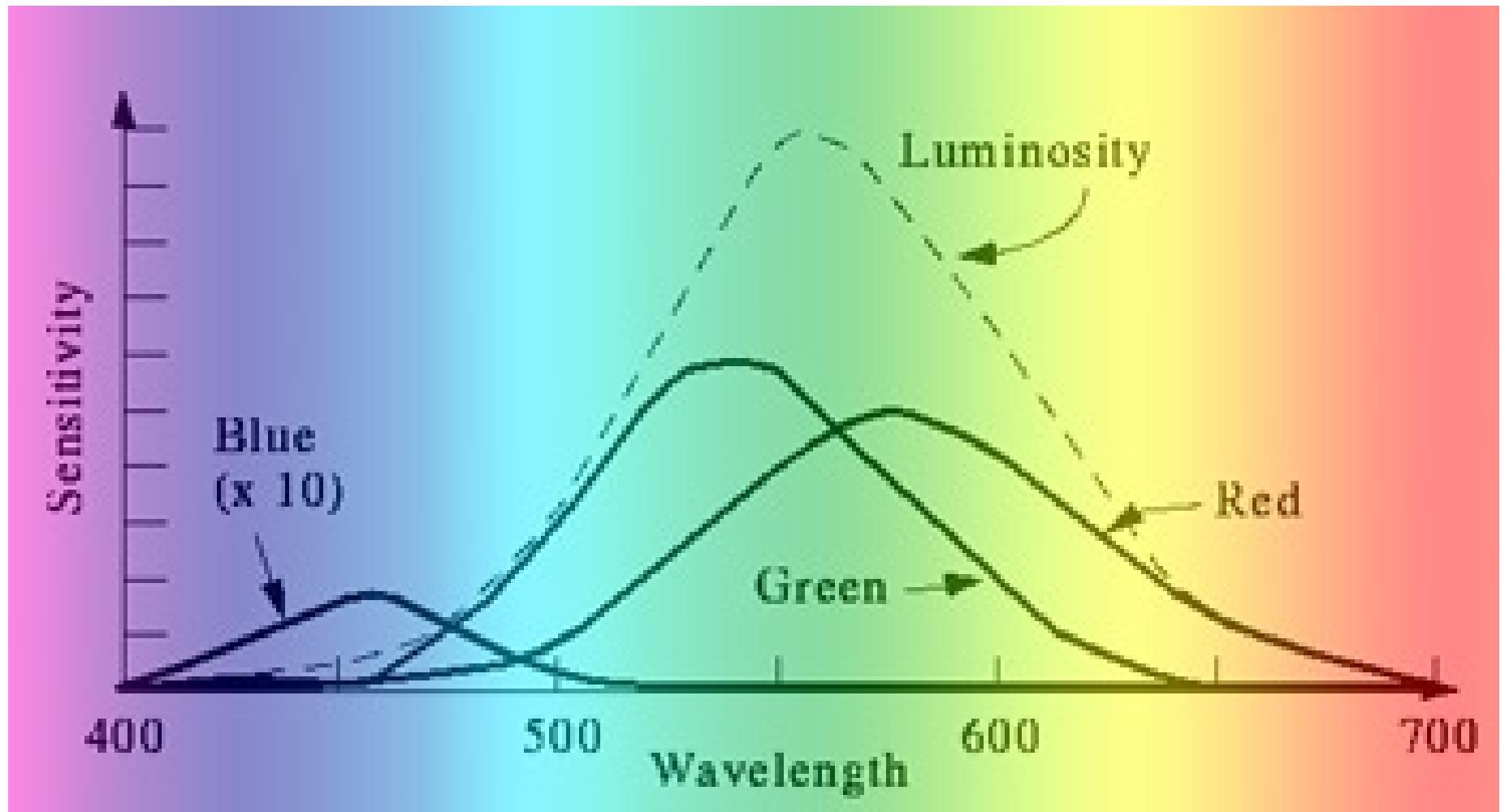


Sintesi sottrattiva CMY (CMYK)
(ad esempio, stampante)

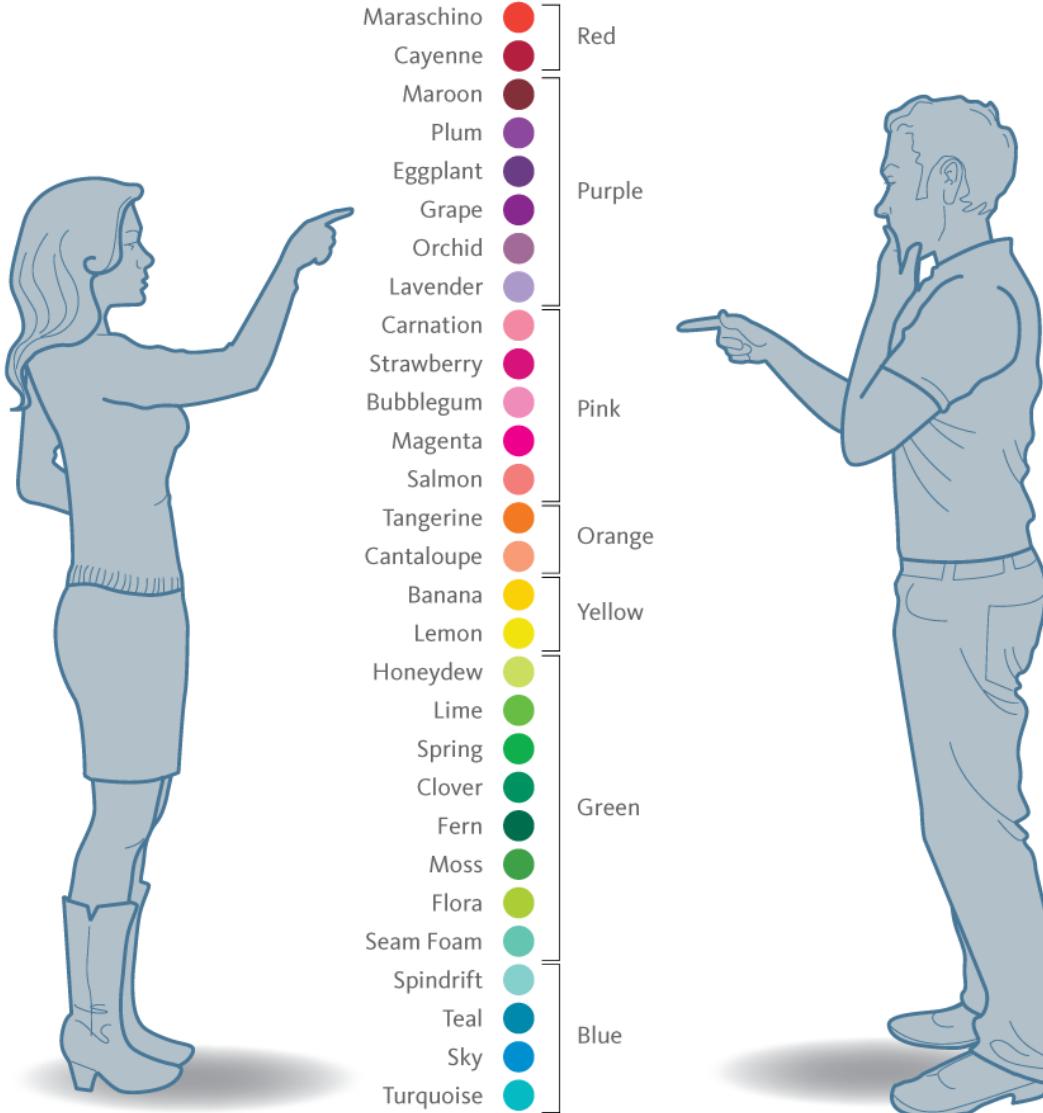
Un mondo a colori...



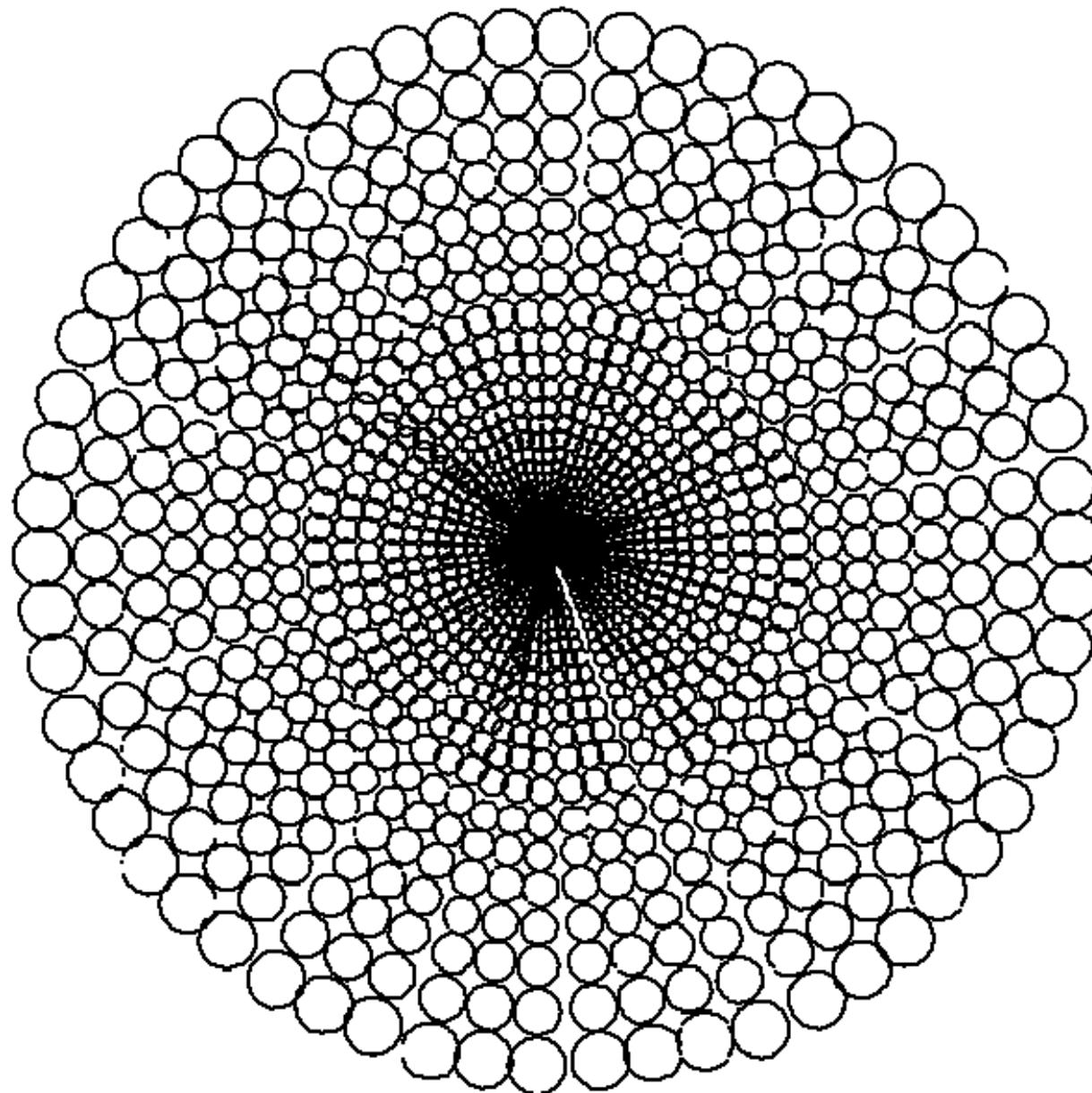
Un mondo a colori...



Un mondo a colori...



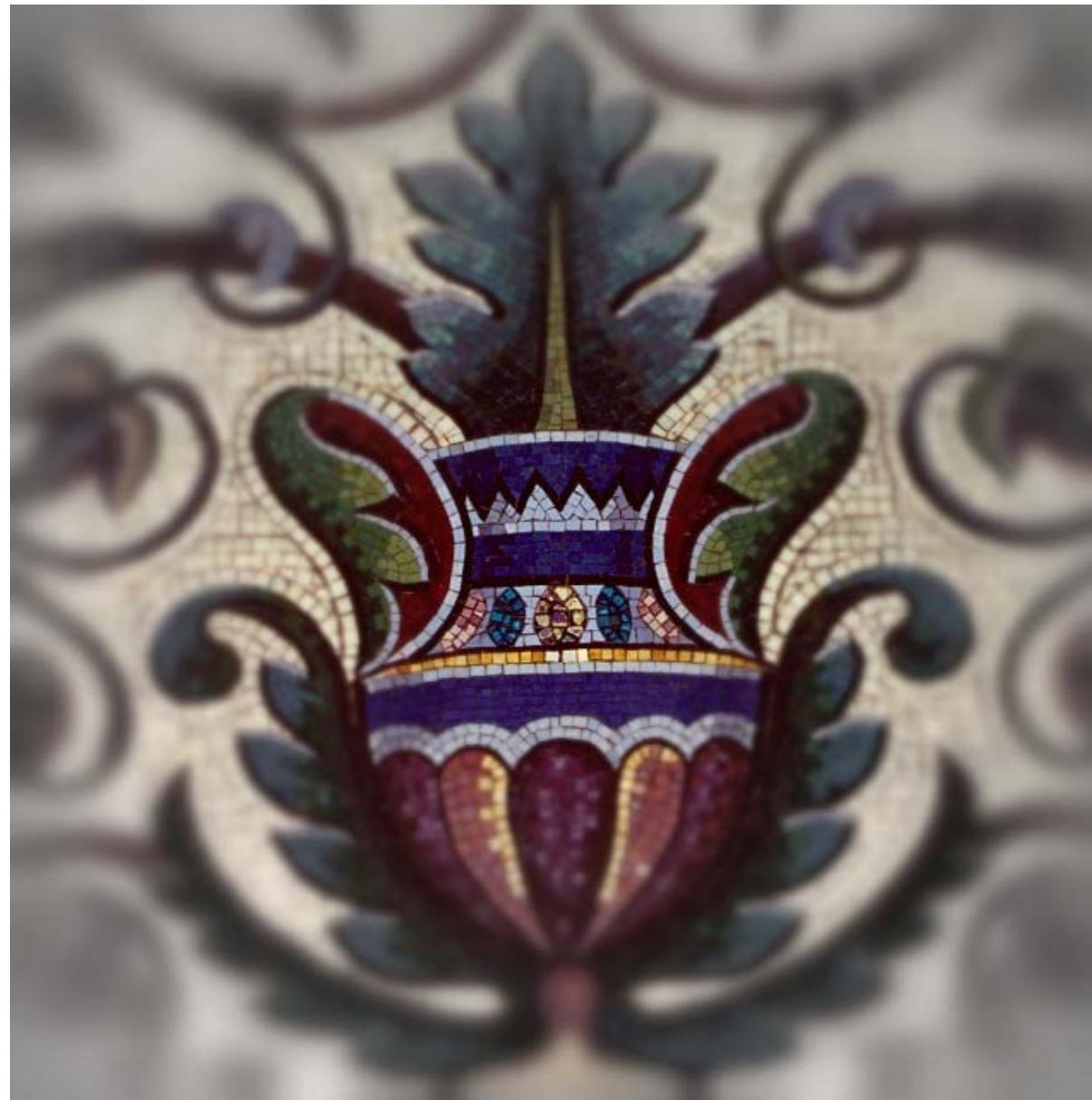
Visione foveata



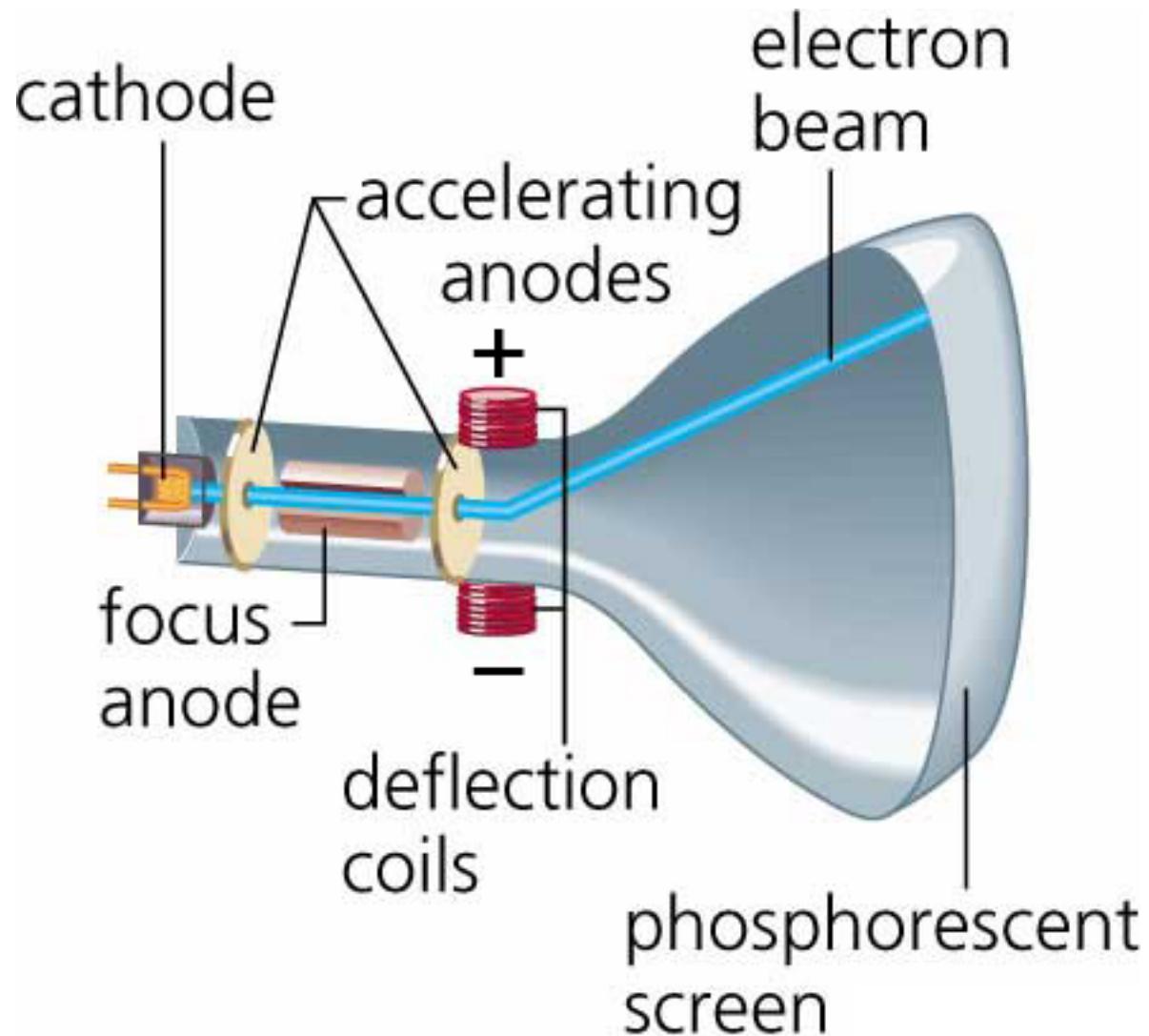
Visione foveata



Visione foveata



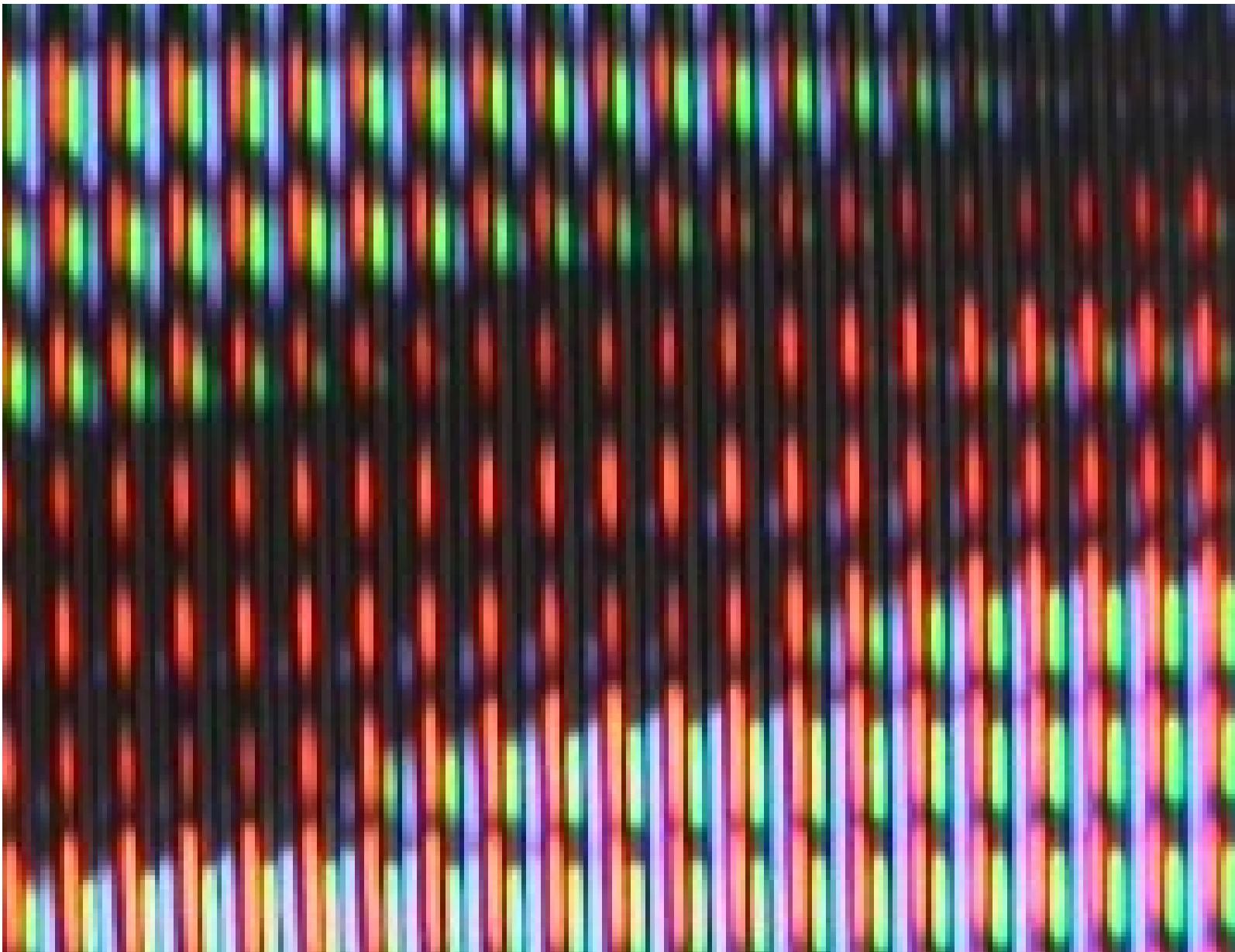
Sistema di output



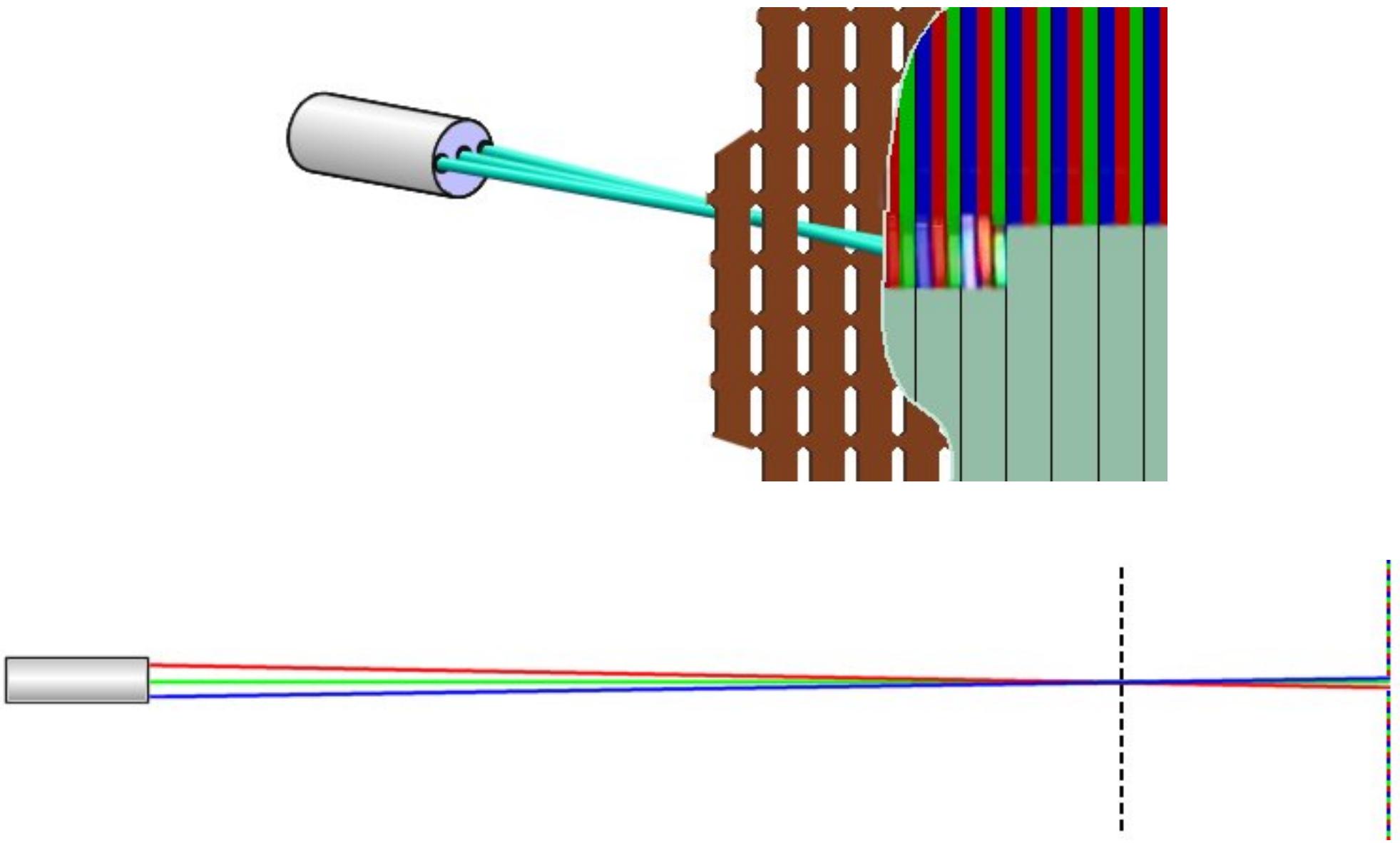
Sistema di output



Sistema di output a colori



Sistema di output a colori

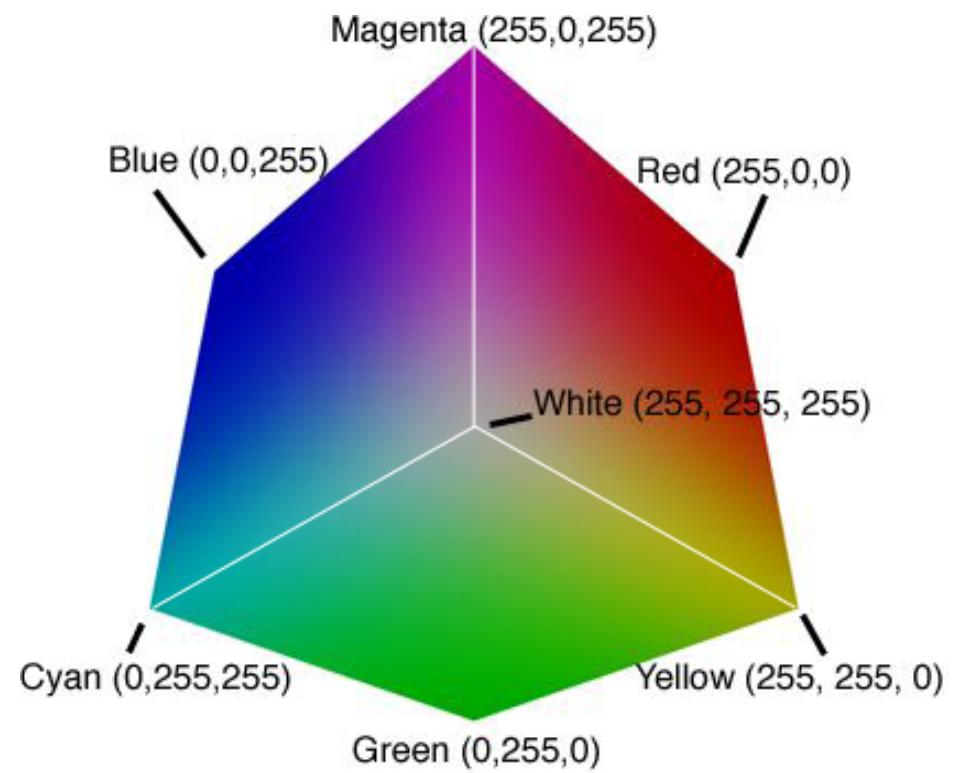
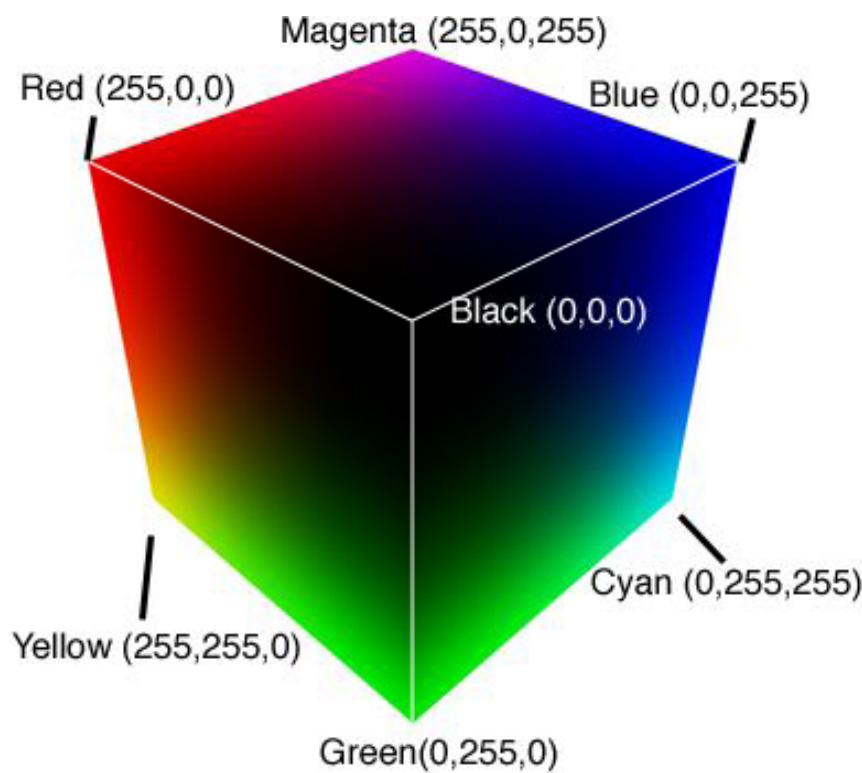


Sistema di output a colori



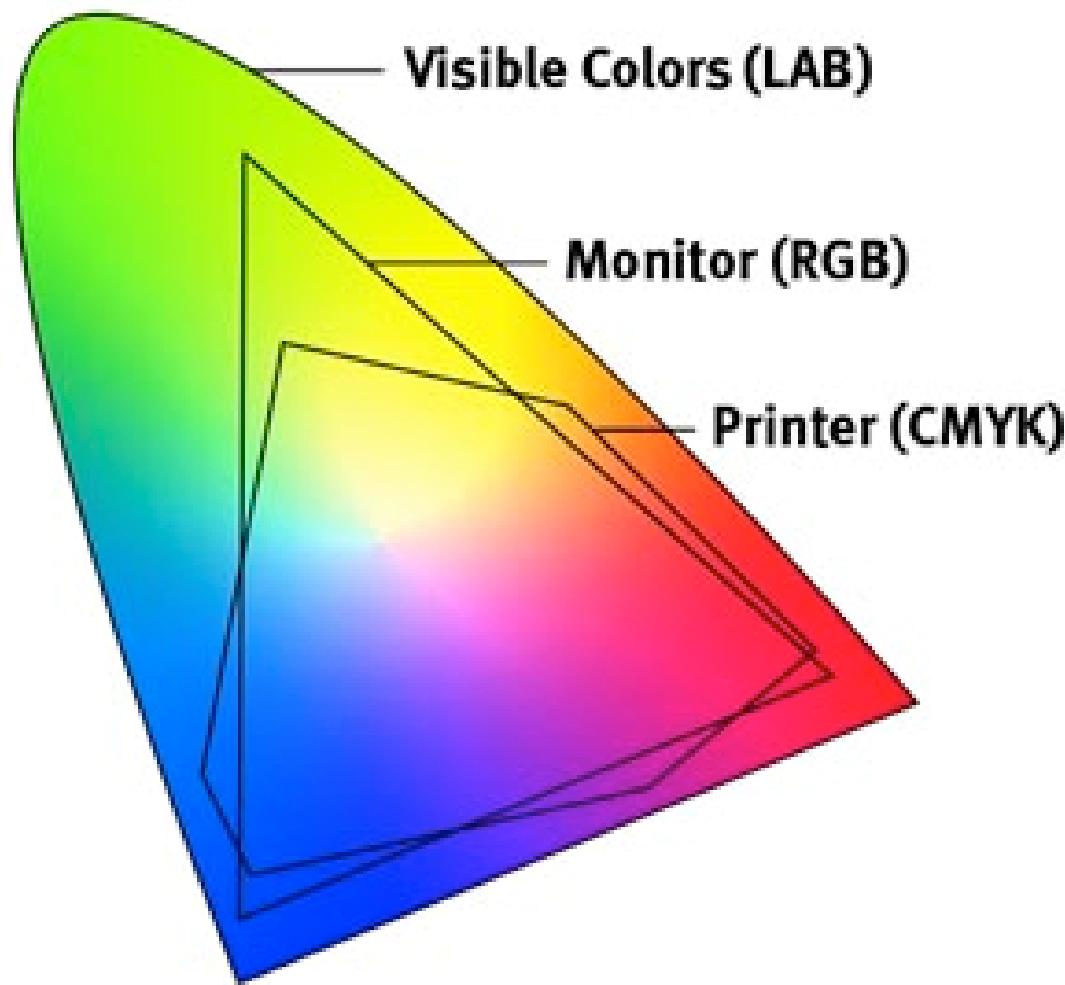
Spazio dei colori RGB

La percezione del colore avviene mediante tricromia: stimolazione di recettori per il rosso (700 nm), verde (546 nm) e blu (436 nm). La rappresentazione RGB è facilmente gestibile in hardware.

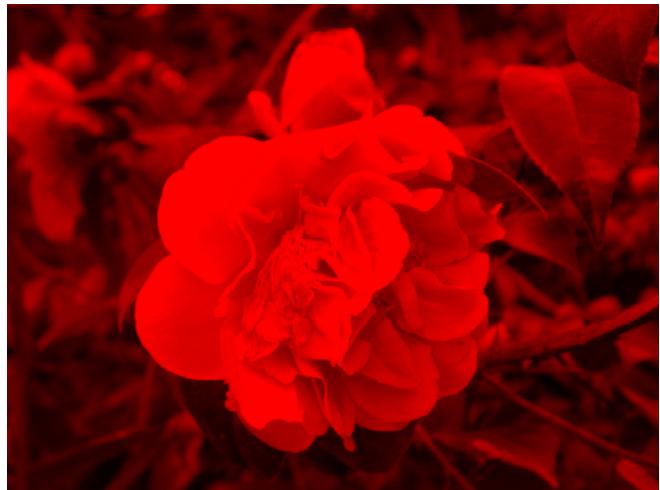


Spazio dei colori RGB (gamut)

Non tutti i colori sono rappresentabili.



Spazio dei colori RGB (Maxwell,1855)



Rappresentazione R



Rappresentazione G



Rappresentazione B



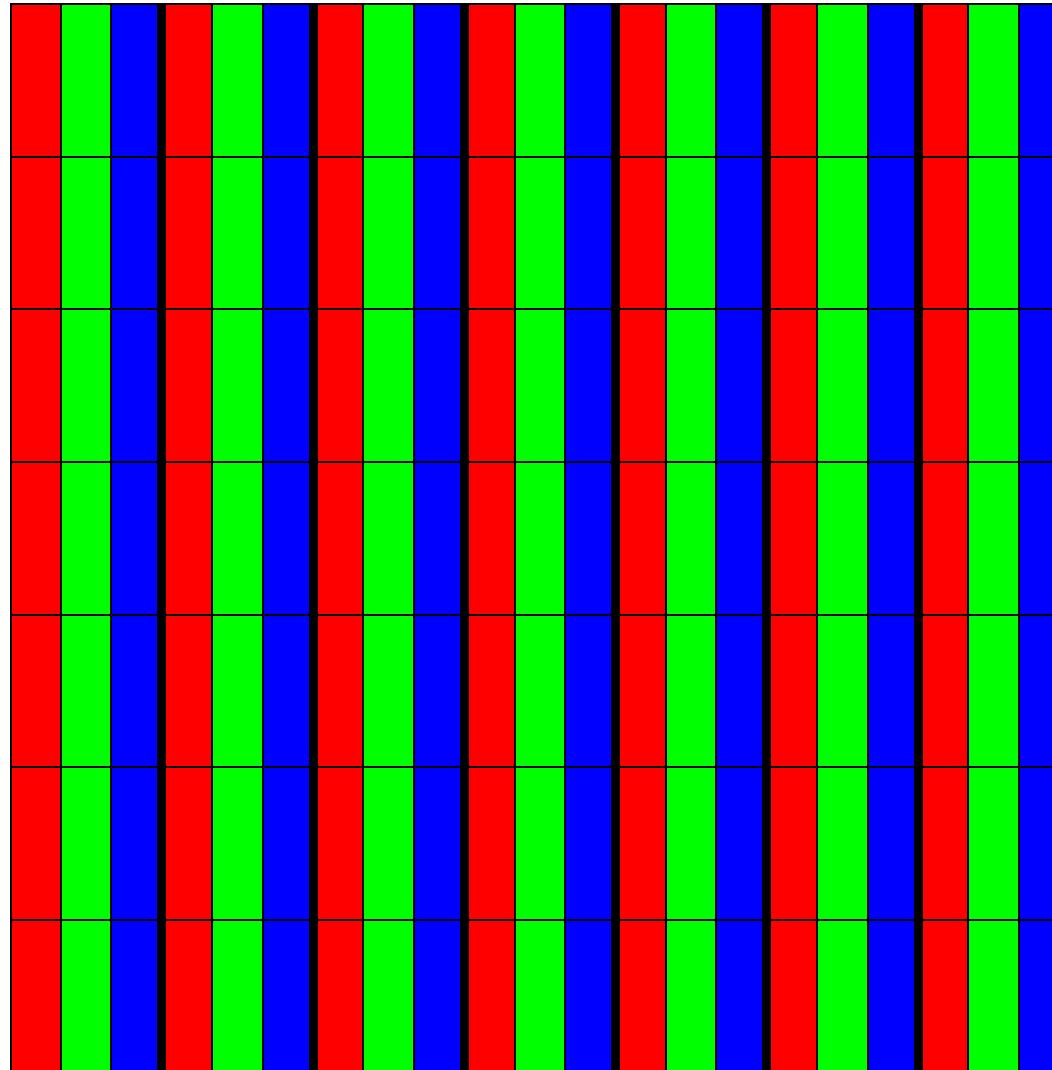
Immagine RGB

Spazio dei colori RGB (Maxwell, 1855)

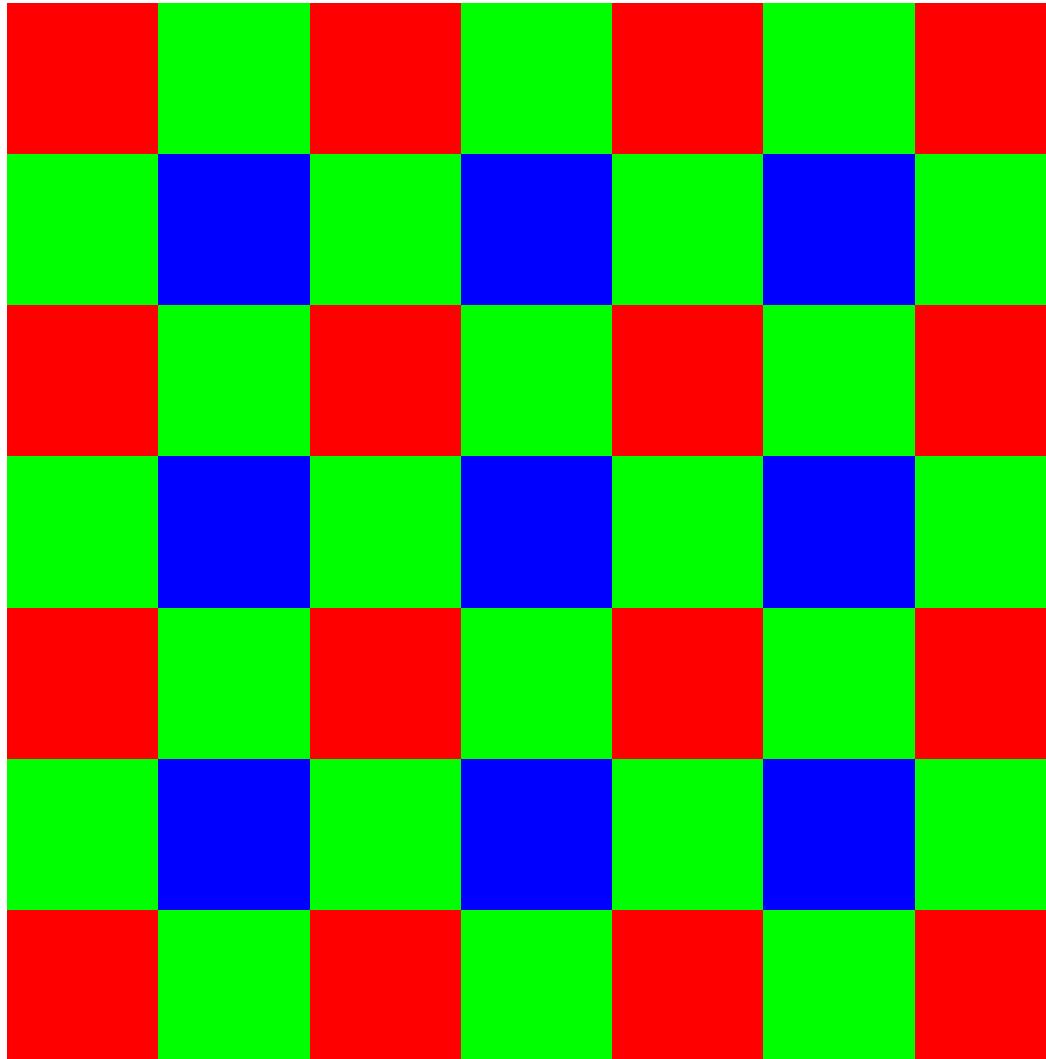


Emir of Bukhara (foto del 1911)

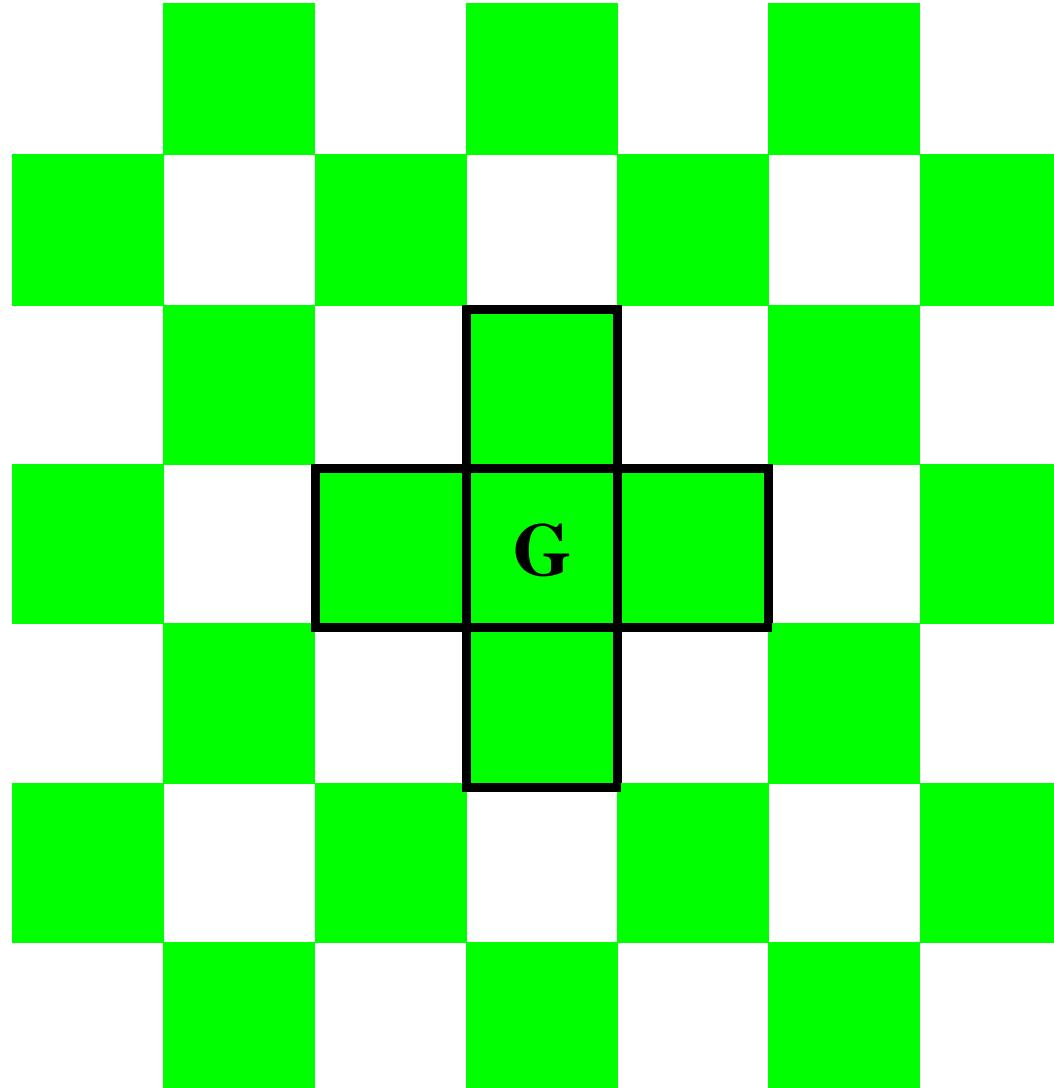
Spazio dei colori RGB (CCD + filtro di Bayer)



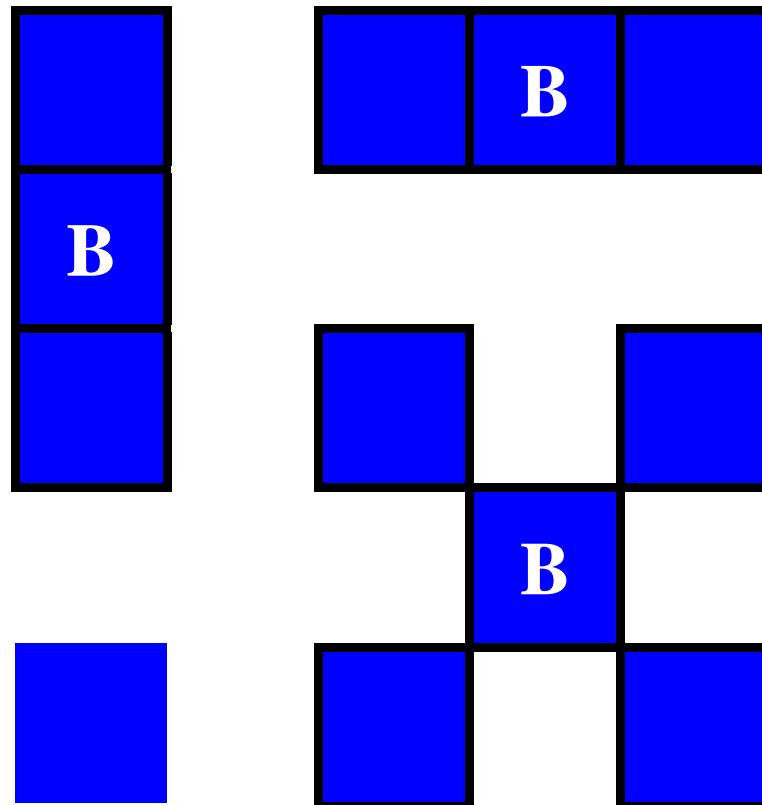
Spazio dei colori RGB (CCD + filtro di Bayer)



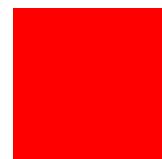
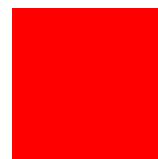
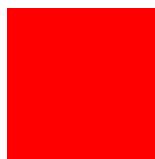
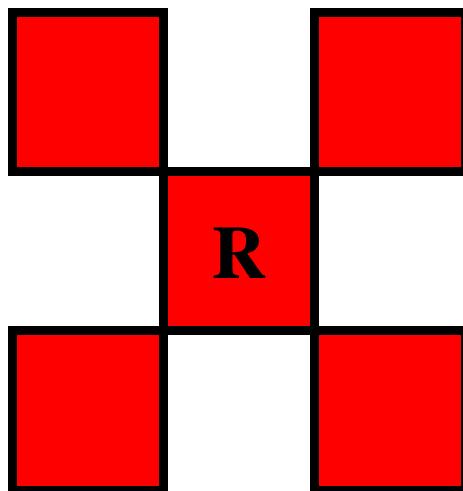
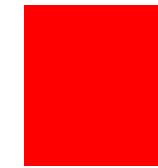
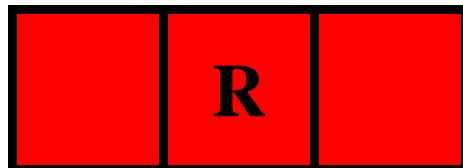
Spazio dei colori RGB (CCD + filtro di Bayer)



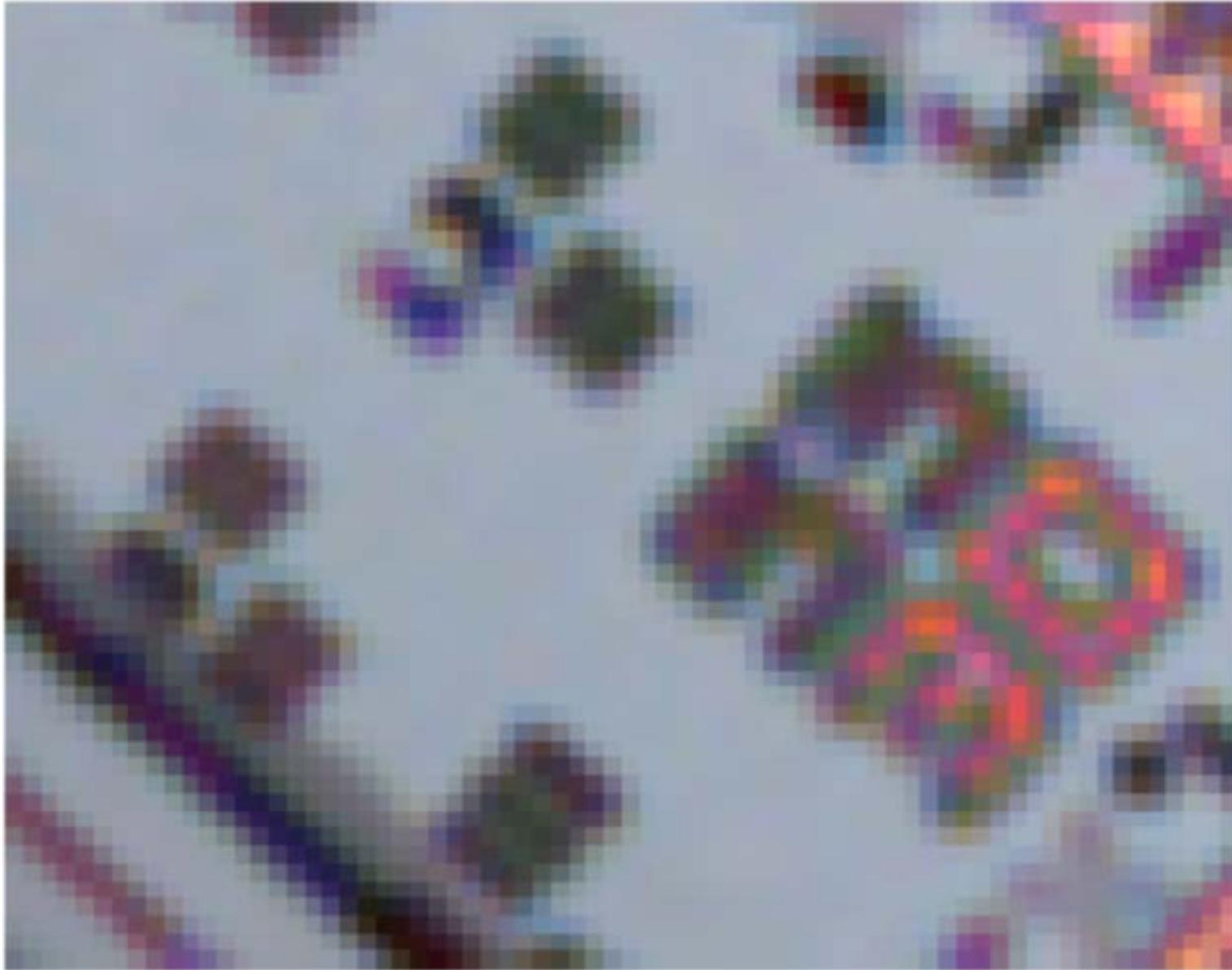
Spazio dei colori RGB (CCD + filtro di Bayer)



Spazio dei colori RGB (CCD + filtro di Bayer)



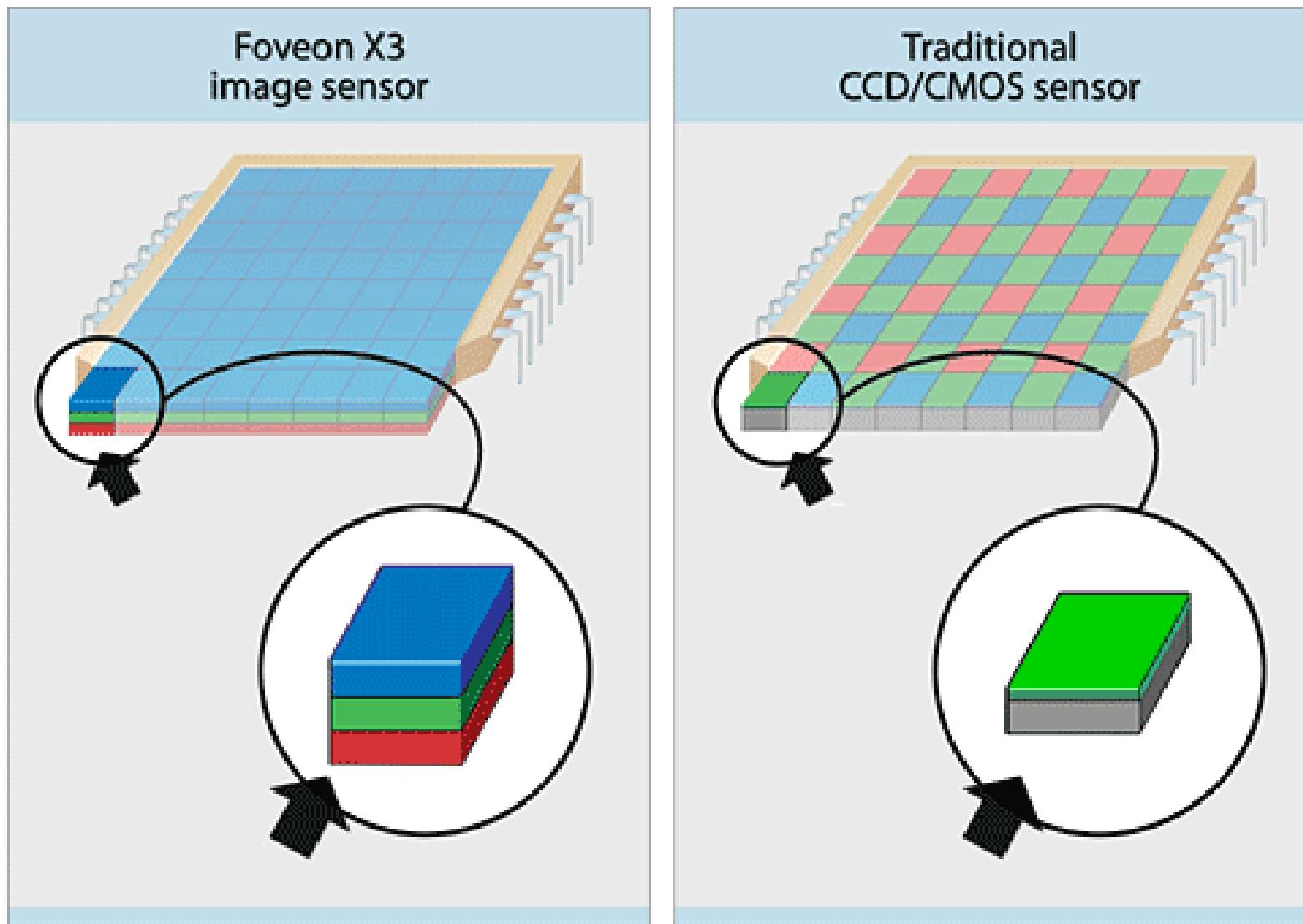
Spazio dei colori RGB (CCD + filtro di Bayer)



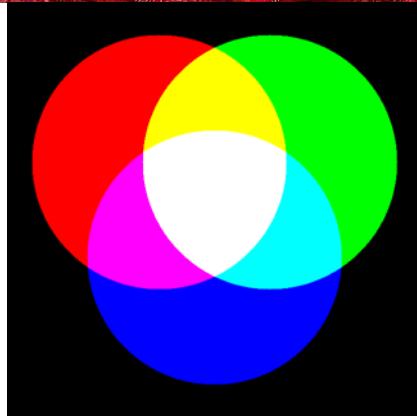
Spazio dei colori RGB (3 CCD)



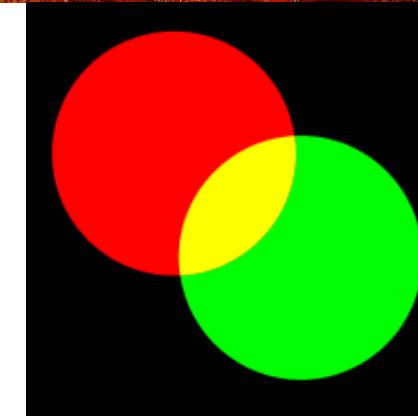
Spazio dei colori RGB (Foveon X3)



Spazio dei colori RG (technicolor, 1916)

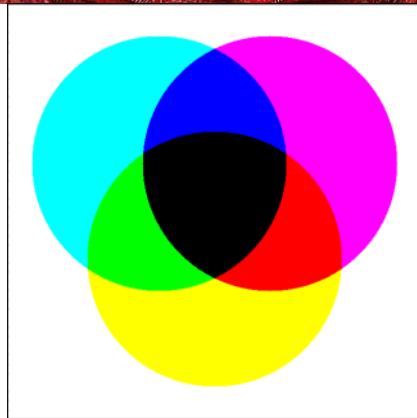


Sintesi additiva RGB

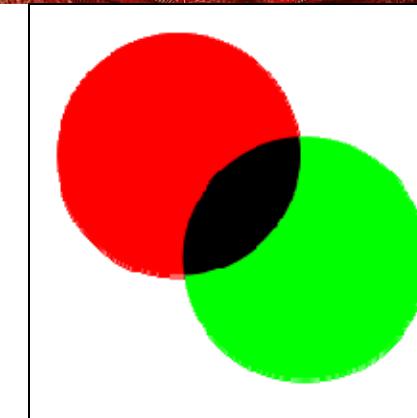


Sintesi additiva RG

Spazio dei colori RG (technicolor, 1922)



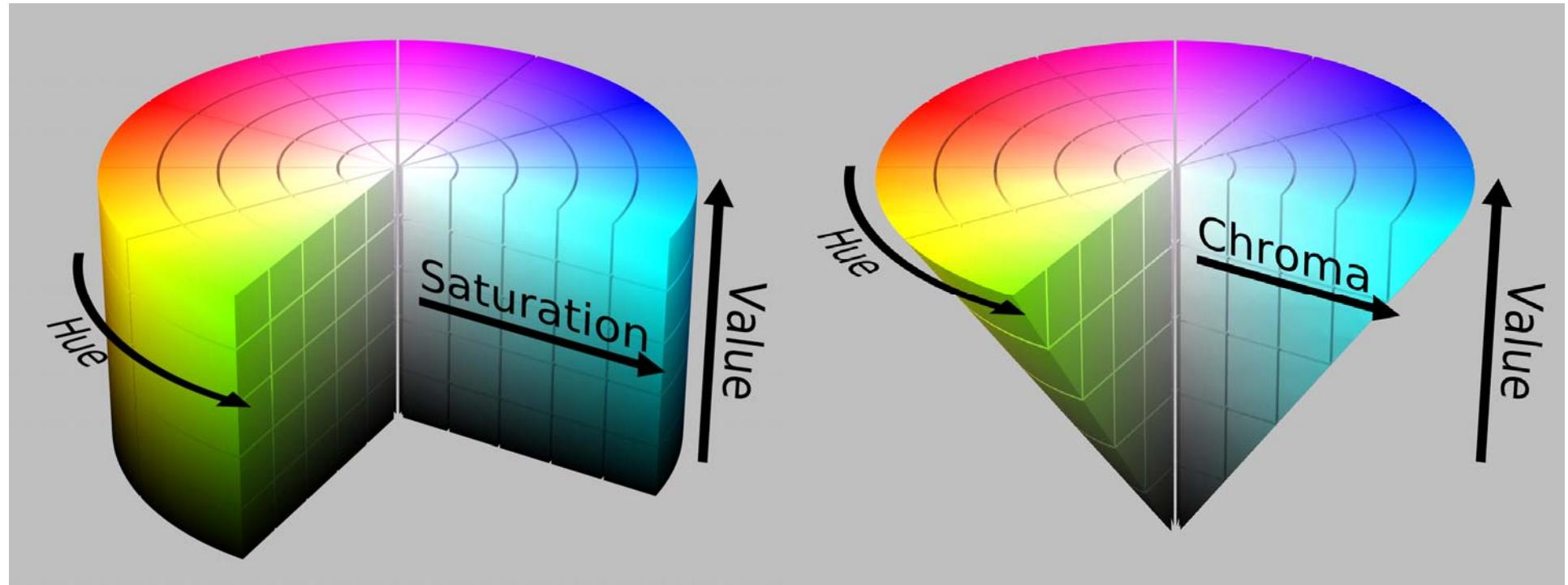
Sintesi sottrattiva RGB



Sintesi sottrattiva RG

Spazio dei colori HSV / HCV

Una rappresentazione naturale dei colori.



Spazio dei colori HSV



$H=0^\circ$



$H=60^\circ$



$H=120^\circ$



$H=180^\circ$



$H=240^\circ$



$H=300^\circ$

Spazio dei colori HSV



S=0



S=20



S=40



S=60

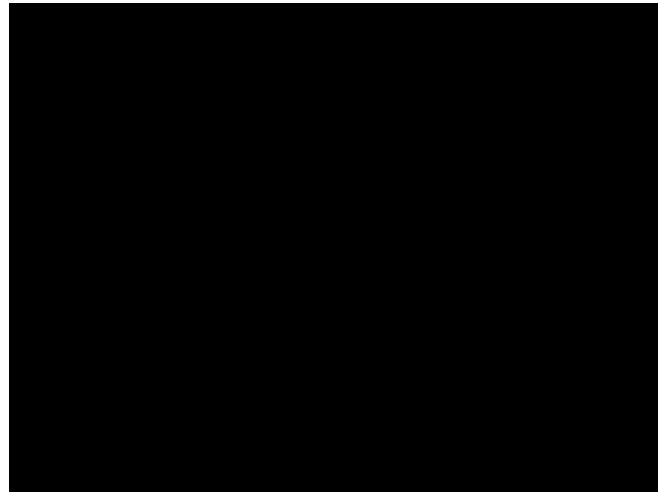


S=80



S=100

Spazio dei colori HSV



V=0



V=20



V=40



V=60



V=80

V=100

Spazio dei colori YUV

Separa la luminanza Y dalla crominanza U e V.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ -.147 & -.289 & .436 \\ .615 & -.515 & -.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Operatori aritmetico/logici

$$\mathbb{0} = \{(i, j, g) : g = 0\}$$

$$\mathbb{1} = \{(i, j, g) : g = 1\}$$

$$\mathbb{255} = \{(i, j, g) : g = 255\}$$

$$k \times I = \{(i, j, g) : g = \min\{G - 1, [k \times g]\}\}, k \geq 0$$

$$k + I = \{(i, j, g) : g = \min\{G - 1, [k + g]\}\}, k \geq 0$$

$$\min(I_1, I_2) = \{(i, j, g) : g = \min\{g_1, g_2\}\}$$

$$\max(I_1, I_2) = \{(i, j, g) : g = \max\{g_1, g_2\}\}$$

Operatori aritmetico/logici

$$I_1+I_2=\{(i,j,g) : g=\min\{G-1, g_1+g_2\}\} \quad (\text{saturazione a } G)$$

$$I_1+I_2=\{(i,j,g) : g=[(g_1+g_2)/2]\}$$

$$I_1-I_2=\{(i,j,g) : g=\max\{0, g_1-g_2\}\} \quad (\text{saturazione a } 0)$$

$$I_1-I_2=\{(i,j,g) : g=|g_1-g_2|\}$$

$$I_1\times I_2=\{(i,j,g) : g=[g_1\times g_2/(G-1)]\}$$

$$\text{sqr}(I)=\{(i,j,g) : g=[g^2/(G-1)]\}$$

$$\text{sqrt}(I)=\{(i,j,g) : g=\left[\sqrt[G-1]{g}\right]\}$$

$$\log(I)=\{(i,j,g) : g=[\log(\varepsilon+g)\times(G-1)/\log(G-1)]\}$$

Operatori aritmetico/logici

$\text{and}(I_1, I_2) = \{(i, j, g) : g = \text{and}(g_1, g_2)\}$

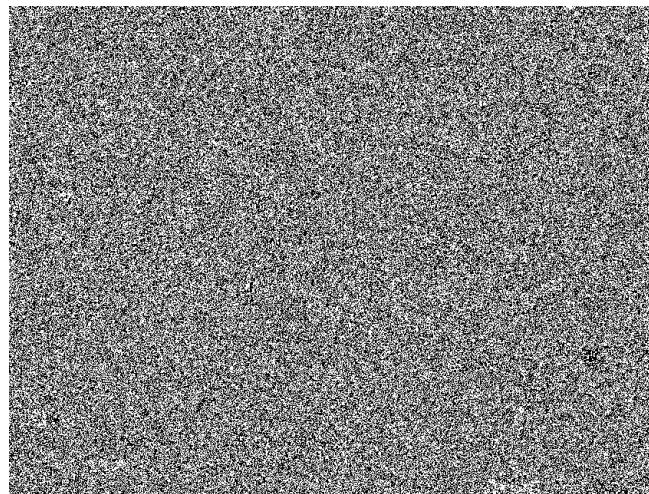
$\text{or}(I_1, I_2) = \{(i, j, g) : g = \text{or}(g_1, g_2)\}$

$\text{not}(I) = \bar{I} = \{(i, j, g) : g = \text{not}(g)\} = \{(i, j, g) : g = G - g\}$

$\text{shift}(I, k) = \{(i, j, g) : g = \text{shift}(g, k)\}$

Operatori aritmetico/logici

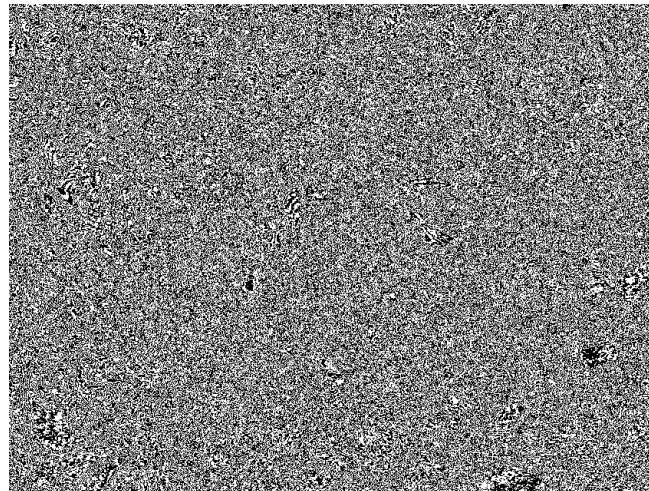
```
flower=rgb2gray(imread('flower.tif'));
i1=bitshift(bitshift(flower,7,'uint8'),-7,'uint8');
figure; imshow(i1,[0,1]);
```



bitplane #1 (LSBP)

Operatori aritmetico/logici

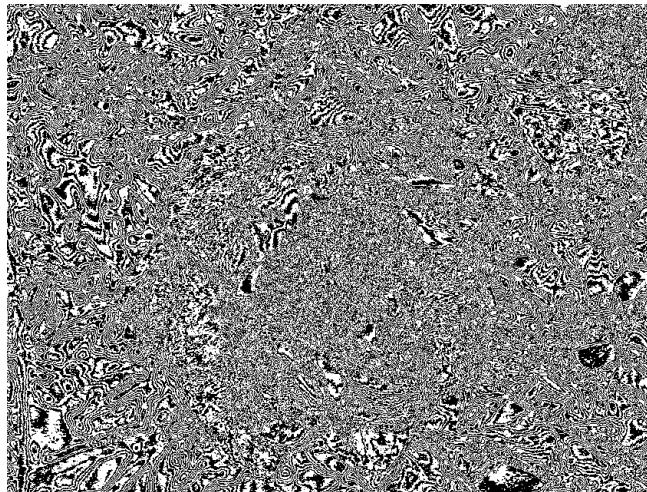
```
i2=bitshift(bitshift(flower,6,'uint8'),-7,'uint8');  
figure; imshow(i2,[0,1]);
```



bitplane #2

Operatori aritmetico/logici

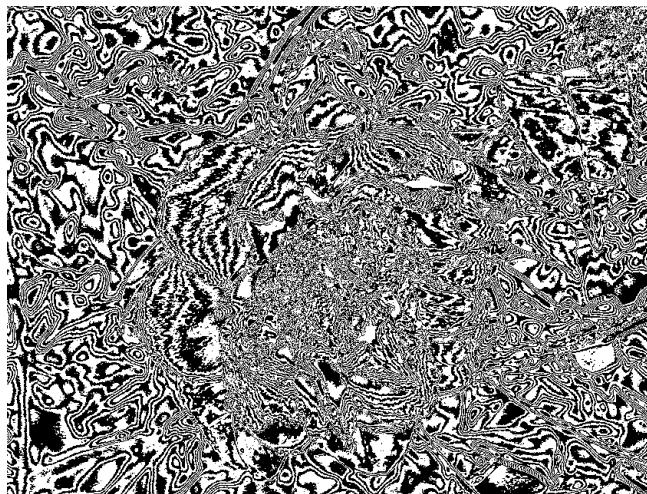
```
i3=bitshift(bitshift(flower,5,'uint8'),-7,'uint8');  
figure; imshow(i3,[0,1]);
```



bitplane #3

Operatori aritmetico/logici

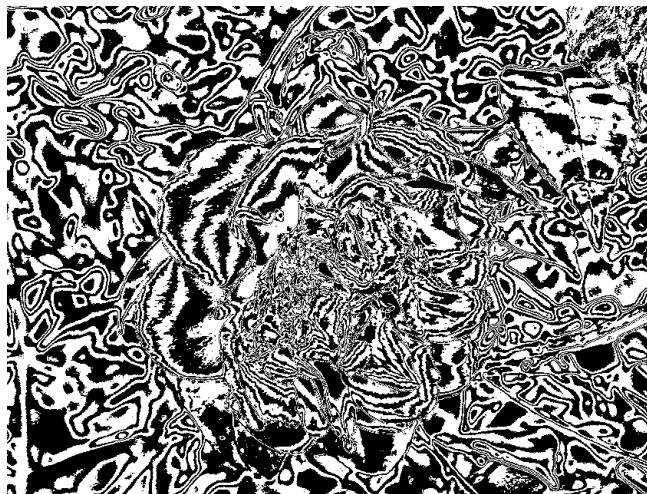
```
i4=bitshift(bitshift(flower,4,'uint8'),-7,'uint8');  
figure; imshow(i4,[0,1]);
```



bitplane #4

Operatori aritmetico/logici

```
i5=bitshift(bitshift(flower,3,'uint8'),-7,'uint8');  
figure; imshow(i5,[0,1]);
```



bitplane #5

Operatori aritmetico/logici

```
i6=bitshift(bitshift(flower,2,'uint8'),-7,'uint8');  
figure; imshow(i6,[0,1]);
```



bitplane #6

Operatori aritmetico/logici

```
i7=bitshift(bitshift(flower,1,'uint8'),-7,'uint8');  
figure; imshow(i7,[0,1]);
```



bitplane #7

Operatori aritmetico/logici

```
i8=bitshift(bitshift(flower,0,'uint8'),-7,'uint8');  
figure; imshow(i8,[0,1]);
```



bitplane #8 (MSBP)

Gray code

Di solito, i bitplane della codifica di Gray sono più uniformi.

```
function g=dec2gray(d)
g=bitxor(d,bitshift(d,-1));
```

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

codifica binaria

```
function d=gray2dec(g)
d=g;
for i=[-32,-16,-8,-4,-2,-1]
    d=bitxor(d,bitshift(d,i));
end;
```

0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	1	0
3	0	0	0	1	0
4	0	1	1	0	0
5	0	1	1	1	1
6	0	1	0	1	0
7	0	1	0	0	0
8	1	1	0	0	0
9	1	1	1	0	1
10	1	1	1	1	1
11	1	1	1	1	0
12	1	0	1	1	0
13	1	0	1	1	1
14	1	0	0	1	1
15	1	0	0	0	0

codifica Gray

Gray code



bitplane #8 (MSB)



bitplane #7



bitplane #6



bitplane #5



Gray bitplane #8



Gray bitplane #7

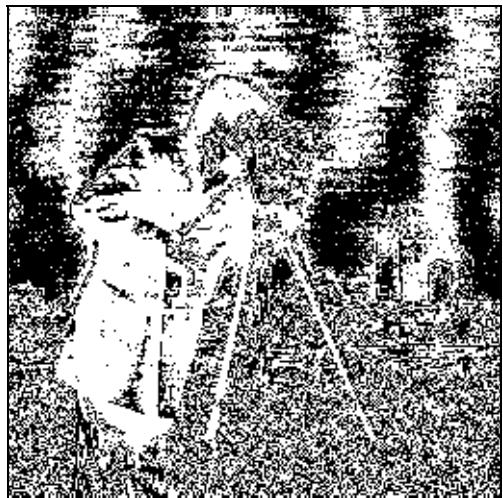


Gray bitplane #6

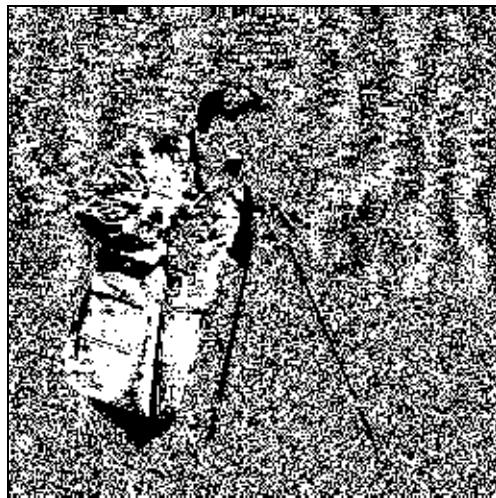


Gray bitplane #5

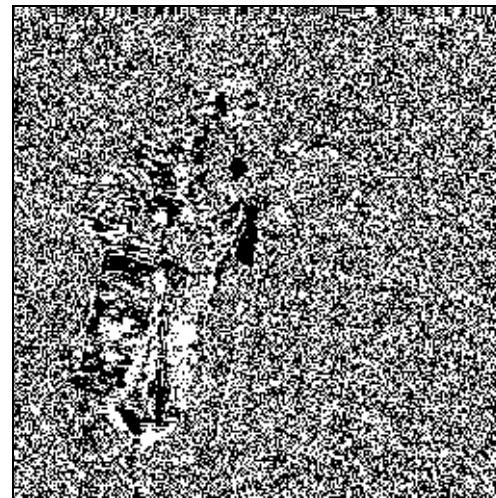
Gray code



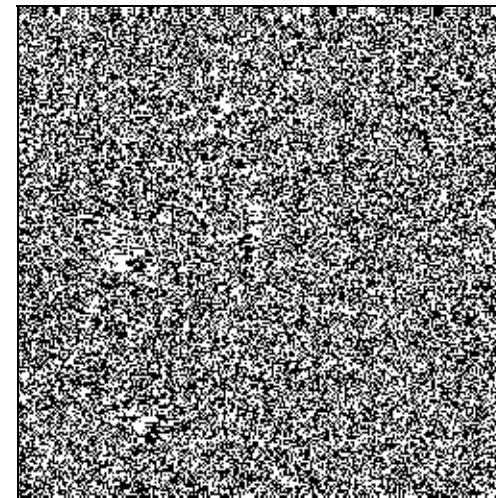
bitplane #4



bitplane #3



bitplane #2



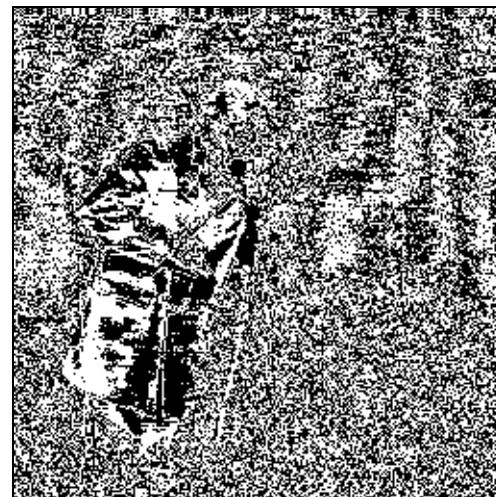
bitplane #1 (LSB)



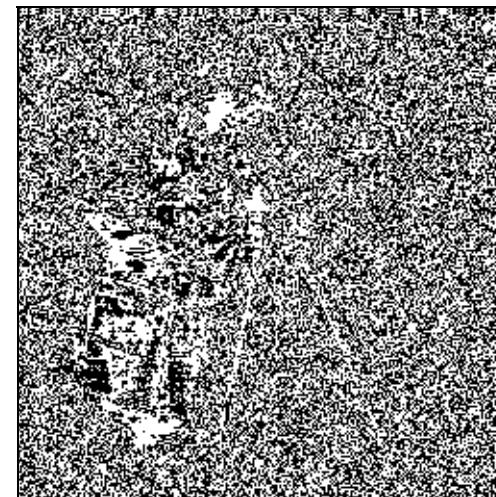
Gray bitplane #4



Gray bitplane #3



Gray bitplane #2



Gray bitplane #1

Filtrati di convoluzione

Generalmente le immagini presentano delle deformazioni dovute allo stesso sistema di acquisizione.

$$I'_D(i,j) = \sum_{x=-\left\lfloor \frac{w}{2} \right\rfloor}^{\left\lfloor \frac{w}{2} \right\rfloor} \sum_{y=-\left\lfloor \frac{h}{2} \right\rfloor}^{\left\lfloor \frac{h}{2} \right\rfloor} I_D(i+x, j+y) \times k(x, y) + N(i, j)$$

Il procedimento (in genere iterativo) tenta cioè di risolvere un sistema di $w \times h$ equazioni in $w \times h$ incognite. Se N è noto, allora il sistema ammette soluzione esatte.

Se l'operatore di deconvoluzione è omogeneo e lineare, allora può essere scritto in forma vettoriale:

$$D' = K \otimes D + N$$

Filtrati di convoluzione (blur)

```
lena=imread('lena.tif');
figure; imshow(lena,[0,255]);
mean3x3=[[1 1 1]; [1 0 1]; [1 1 1]]/8;
blurred3x3=conv2(lena, mean3x3, 'same');
figure; imshow(blurred3x3,[0,255]);
```



Lena Sjööblom (Miss November 1972)



mean 3×3

Filtrati di convoluzione (blur)

```
mean7x7=ones(7); mean7(4,4)=0; mean7=mean7/sum(mean7(:));  
blurred7x7=conv2(lena, mean7x7, 'same');  
figure; imshow(blurred7x7,[0,255]);  
blurred3x3bis=conv2(conv2(blurred3x3, mean3x3, 'same'), mean3x3, 'same');  
blurred3x3bis=conv2(conv2(blurred3x3bis, mean3x3, 'same'), mean3x3, 'same');  
figure; imshow(blurred3x3bis,[0,255]); % repeated 3 times
```

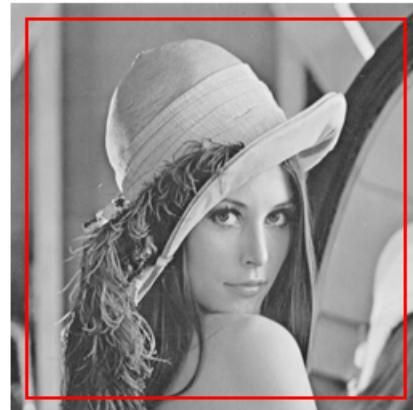


mean 7×7

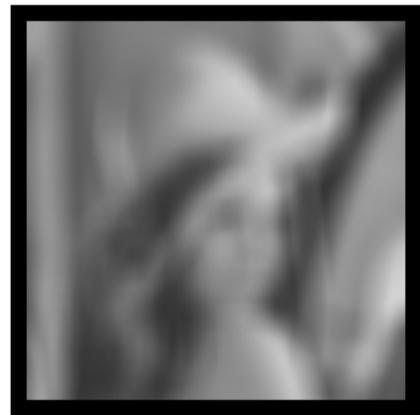


mean 3×3 , repeated 5 times

Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



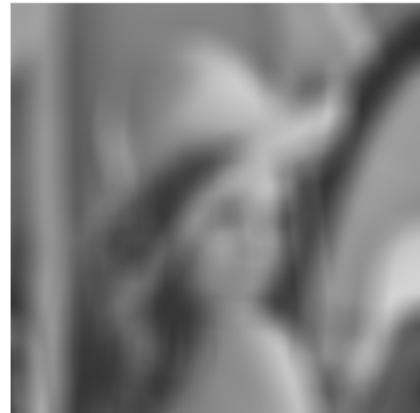
Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



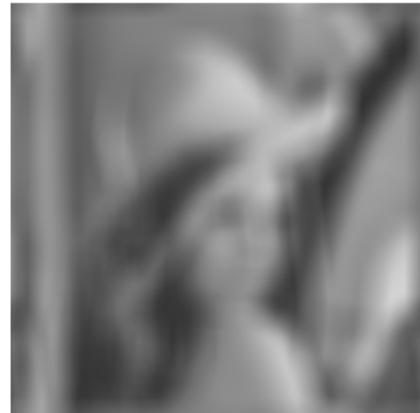
Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



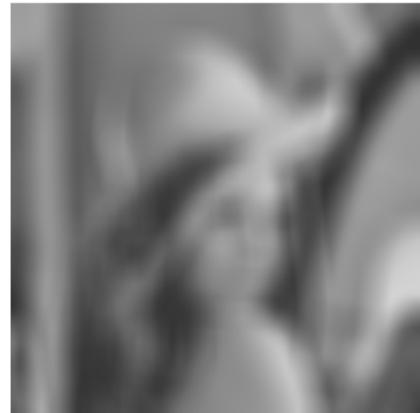
Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)



Filtri di convoluzione (problema del bordo)

