

Capitolo 2

Sorgenti

- Rappresentazione digitale di
 - Testo
 - Immagini
 - Audio
 - Video

Testo - ASCII

Codifica → associazione tra caratteri alfabeto e stringhe di bit

- ASCII
- 7 bit → 128 caratteri
 - 95 stampabili (lettere minuscole, maiuscole, numeri, segni interpunzione)
 - 33 non stampabili (controllo formattazione, cancellazione, escape)
- Solo inglese

0010 0001	33	21	!	0100 0010	66	42	B
0010 0010	34	22	"	0100 0011	67	43	C
0010 0011	35	23	#	0100 0100	68	44	D
0010 0100	36	24	\$	0100 0101	69	45	E
0010 0101	37	25	%	0100 0110	70	46	F
0010 0110	38	26	&	0100 0111	71	47	G
0010 0111	39	27	'	0100 1000	72	48	H
0010 1000	40	28	(0100 1001	73	49	I
0010 1001	41	29)	0100 1010	74	4A	J
0010 1010	42	2A	*	0100 1011	75	4B	K
0010 1011	43	2B	+	0100 1100	76	4C	L

Testo - ASCII

Binary	Oct	Dec	Hex	Abbr	PR	CS	CEC	Description
0000 0000	000	0	00	NUL	⁰ NUL	^@		Null character
0000 0001	001	1	01	SOH	¹ SOH	^A		Start of Header
0000 0010	002	2	02	STX	² STX	^B		Start of Text
0000 0011	003	3	03	ETX	³ ETX	^C		End of Text
0000 0100	004	4	04	EOT	⁴ EOT	^D		End of Transmission
0000 0101	005	5	05	ENQ	⁵ ENQ	^E		Enquiry
0000 0110	006	6	06	ACK	⁶ ACK	^F		Acknowledgment
0000 0111	007	7	07	BEL	⁷ BEL	^G	\a	Bell
0000 1000	010	8	08	BS	⁸ BS	^H	\b	Backspace
0000 1001	011	9	09	HT	⁹ HT	^I	\t	Horizontal Tab
0000 1010	012	10	0A	LF	¹⁰ LF	^J	\n	Line feed
0000 1011	013	11	0B	VT	¹¹ VT	^K		Vertical Tab
0000 1100	014	12	0C	FF	¹² FF	^L	\f	Form feed
0000 1101	015	13	0D	CR	¹³ CR	^M	\r	Carriage return
0000 1110	016	14	0E	SO	¹⁴ SO	^N		Shift Out
0000 1111	017	15	0F	SI	¹⁵ SI	^O		Shift In
0001 0000	020	16	10	DLE	¹⁶ DLE	^P		Data Link Escape
0001 0001	021	17	11	DC1	¹⁷ DC1	^Q		Device Control 1 (oft. XON)
0001 0010	022	18	12	DC2	¹⁸ DC2	^R		Device Control 2
0001 0011	023	19	13	DC3	¹⁹ DC3	^S		Device Control 3 (oft. XOFF)

0001 0100	024	20	14	DC4	²⁰ DC4	^T		Device Control 4
0001 0101	025	21	15	NAK	²¹ NAK	^U		Negative Acknowledgement
0001 0110	026	22	16	SYN	²² SYN	^V		Synchronous Idle
0001 0111	027	23	17	ETB	²³ ETB	^W		End of Trans. Block
0001 1000	030	24	18	CAN	²⁴ CAN	^X		Cancel
0001 1001	031	25	19	EM	²⁵ EM	^Y		End of Medium
0001 1010	032	26	1A	SUB	²⁶ SUB	^Z		Substitute
0001 1011	033	27	1B	ESC	²⁷ ESC	^[\e	Escape
0001 1100	034	28	1C	FS	²⁸ FS	^\		File Separator
0001 1101	035	29	1D	GS	²⁹ GS	^]		Group Separator
0001 1110	036	30	1E	RS	³⁰ RS	^^		Record Separator
0001 1111	037	31	1F	US	³¹ US	^_		Unit Separator
0111 1111	177	127	7F	DEL	¹²⁷ DEL	^?		Delete

Figura 2.1: Caratteri di controllo ASCII.

ISO-8859-n

- 8 bit
- Primo bit = 0 → codice ASCII
- Primo bit = 1 → alfabeto n (n=1,...,16 → turco, cirillico, west europe...) [pag.19]

Binary	Oct	Dec	Hex	1	2	3	4	5	6	7	8	9	10	11	13	14	15	16	
10100000	240	160	A0	Non-breaking space (NBSP)															
10100001	241	161	A1	ı	À	Ĥ	Ä	Ë		‘		ı	À	ŋ	”	□	ı	À	
10100010	242	162	A2	ç	˘	κ	Ɔ		’	ç	ç	Ë	ʘ	ç	□	ç	ą		
10100011	243	163	A3	£	Ł	£	Ŕ	Í			£	Ç	ʘ		£		Ł		
10100100	244	164	A4	α			€	α	€		α	Ī	ŋ	α	Č		€		
10100101	245	165	A5	¥	Ł		İ	S		□	¥	İ	ŋ	„	č	¥	„		
10100110	246	166	A6	ı	Š	Ĥ	Ł	I			ı	Ç	ʘ	ı	□		Š		
10100111	247	167	A7	§				İ			§		ı	§					
10101000	250	168	A8	”			J			”		Ł	ą	Ø	Ŵ		š		
10101001	251	169	A9	©	Š	ı	Š	Љ			©		Đ	ą	©				
10101010	252	170	AA	ª	Ş	Ë	Ĥ		□	×	ª	Š	ʘ	Ŕ	Ŵ	ª	□		
10101011	253	171	AB	«	İ	Ç	Ç	Ɔ			«		Ɔ	ʘ	□		«		
10101100	254	172	AC	¬	Ž	Ĵ	Ɔ	Ć	‘		¬		Ž	ŋ	¬	Ÿ	¬	Ž	
10101101	255	173	AD	soft hyphen (SHY)											ı	SHY			
10101110	256	174	AE	®	Ž		Ž	Ÿ			®	Ū	ŋ		®		ž		

UTF-8 e USC

- **Lunghezza variabile**

- Qualsiasi tipo di carattere o simbolo.

Corrispondenza tra codice e simbolo → Universal Character Set (USC) basato su 31 bit

- **k BYTE** per ogni carattere ($k = 1, \dots, 6$)

- **K = 1**

- **MSB = 0 / 7 bit per codifica ASCII**

- **K > 1**

- Primo byte → **k bit = 1 / 0 / 7-k bit = codifica**
- Altri byte → **prefisso 10 / 6 bit per codifica**

$$b = \begin{cases} 7 & \text{se } k = 1 \\ (7 - k) + 6(k - 1) = 5k + 1 & \text{se } k > 1 \end{cases}$$

K=1 → b=7
K=2 → b=11
K=3 → b=16
K=4 → b=21
K=5 → b=26
K=6 → b=31

Immagini

- **Definizione**
- **Pixel**
- **Dimensione e risoluzione**
- **Fattore di forma**
- **Rappresentazione → piani colore**
- **Monitor**
 - CRT
 - LCD
 - plasma, proiettori, oled
 - schede video
- **Formati RAW → rappresentazione digitale senza alcuna compressione**
 - PBM/PGM/PPM
 - XBM
 - XPM

Immagini – definizione e pixel

- **Definizione di immagine in formato digitale**
 - Insieme di elementi (pixel) giustapposti in uno spazio bidimensionale
- **Pixel**
 - **Elemento di immagine**
 - Concetto **ASTRATTO** non legato alla geometria dell'elemento (quindi non “punto”!!!) bensì al contenuto di informazione → **CAMPIONE** dell'immagine
 - “Punto” → significato fisico e geometrico → stampa
 - **Contenuto informativo** → **intensità variabile**
 - Bianco e nero → acceso o spento
 - Scale di grigio → scalare (es tra 0 e 255)
 - Colori → vettore a 3 o 4 dimensioni
 - Rappresentazione → **piani colore**

Dimensione, risoluzione, fattore di forma

- **Dimensione**
 - Numero di pixel orizzontali (W) X numero di pixel verticali (H)
 - Es: 1024x768
- **Risoluzione**
 - Dots per pollice → stampa → concetto fisico di punto
- **Spazio di memorizzazione**
 - Con n bpp → $nWH/8$ byte
- **Fattore di forma**
 - Rapporto tra la dimensione orizzontale e quella verticale (W/H)
 - Es: 4/3, 16/9, 16/10

Piani Colore

- **Definizione**

- Sistemi di rappresentazione del colore, basati sul meccanismo della visione a colori nell'occhio umano

- **Occhio umano**

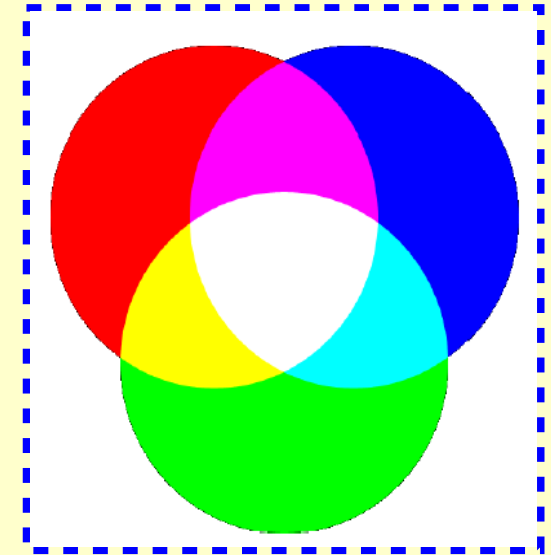
- Coni → recettori che rispondono su tre lunghezze d'onda (giallo-verde, blu-verde, blu-viola)
- Sovrapposizione della sensibilità spettrale dei tre coni

- **Alcuni piani colore**

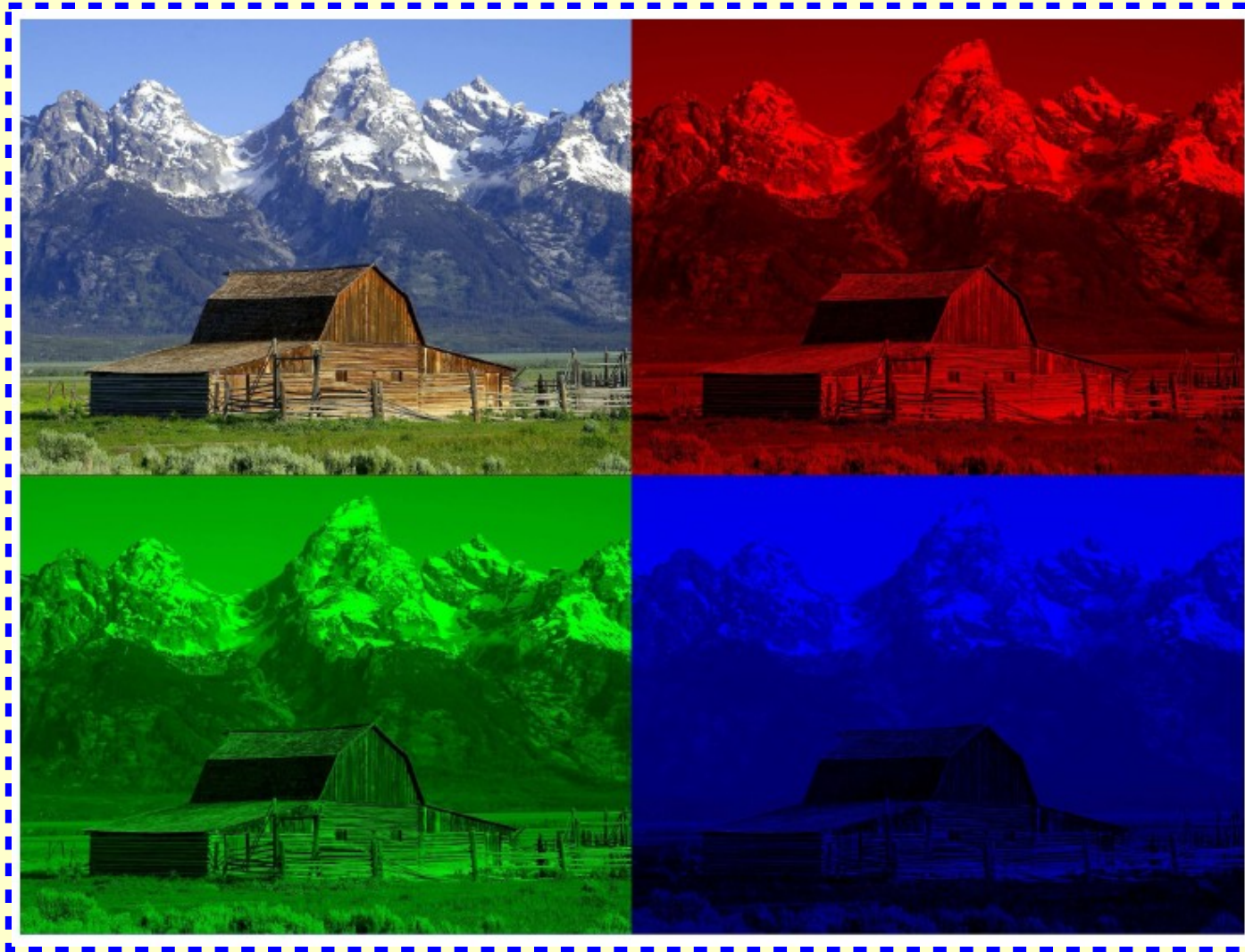
- RGB
- CMYK e CMY
- YUV
- YPbPr e YCbCr
- HSV

Red Green Blue (RGB)

- **Sintesi additiva**
- **Rappresentazione di ogni pixel**
 - 24bit per pixel (bpp) → 8 per ogni colore → 16M colori
 - Capacità occhio umano → 10M colori
 - Decimale, esadecimale, reale tra 0 e 1
(128,128,128) , (#808080), (0.5,0.5,0.5)
 - Nero=#000000, Bianco=#FFFFFF
 - Codifica non lineare → gamma correction $f(x)=x^{1/\gamma}$, γ tipico 2.2
 - Modalità RGB
 - **Truecolor** → 24bpp, 8 bit per colore
 - **Highcolor** → 16bpp, 5 bit per rosso e blu e 6 per verde
 - **32bpp** → 8 bit per colore e 8 bit per trasparenza o per ottimizzare accesso a memoria o comunicazioni tra microprocessore e scheda grafica
 - **48 bpp** → 16 bit per colore, applicazioni professionali, per evitare sgranatura durante i passaggi di arrotondamento dovuti a operazioni di filtraggio digitale

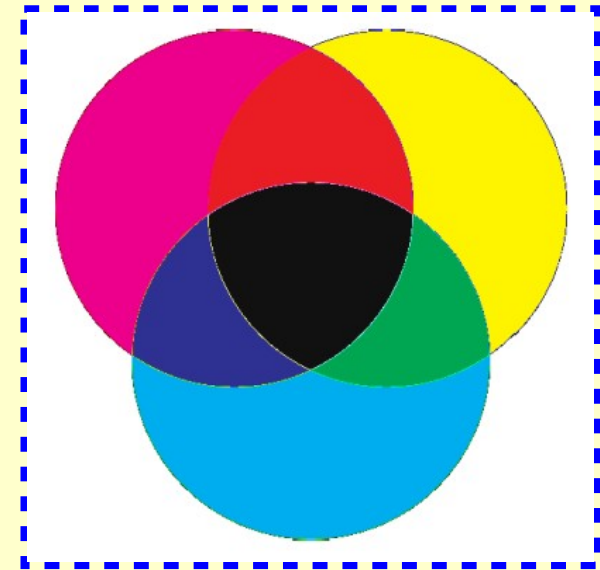


Red Green Blue (RGB)



Cyan Magenta Yellow black (CMYK - CMY)

- **Sintesi sottrattiva**
 - Colore percepito → parte di luce non assorbita
 - Processo di stampa
- **Nero**
- **Conversione RGB e CMYK** (non biunivoca)
- **Conversione RGB e CMY**



$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 - C(1 - K) - K \\ 1 - M(1 - K) - K \\ 1 - Y(1 - K) - K \end{bmatrix}$$

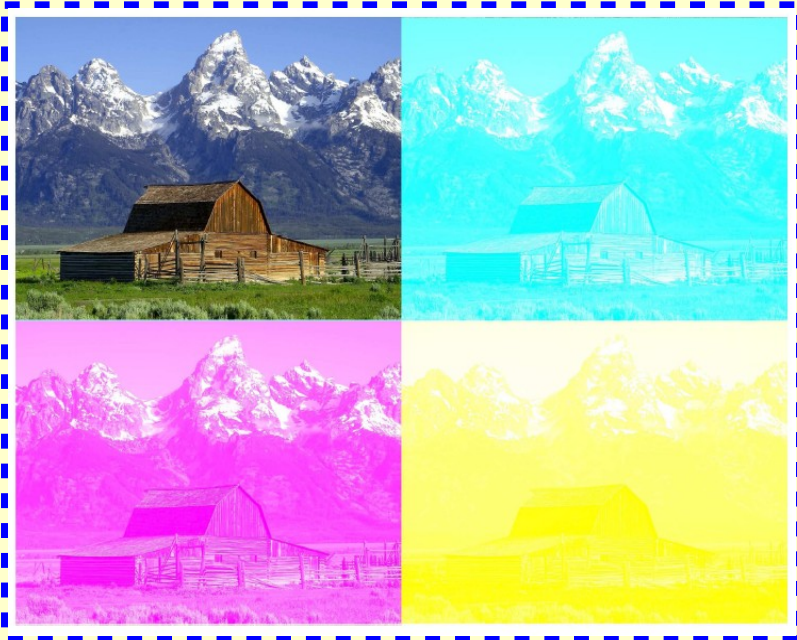
$$\begin{bmatrix} C \\ M \\ Y \\ K \end{bmatrix} = \begin{bmatrix} \frac{1 - R - \min\{1 - R, 1 - G, 1 - B\}}{1 - \min\{1 - R, 1 - G, 1 - B\}} \\ \frac{1 - G - \min\{1 - R, 1 - G, 1 - B\}}{1 - \min\{1 - R, 1 - G, 1 - B\}} \\ \frac{1 - Y - \min\{1 - R, 1 - G, 1 - B\}}{1 - \min\{1 - R, 1 - G, 1 - B\}} \\ \min\{1 - R, 1 - G, 1 - B\} \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 - C \\ 1 - M \\ 1 - Y \end{bmatrix}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 - R \\ 1 - G \\ 1 - B \end{bmatrix}$$

Cyan Magenta Yellow black (CMYK)

Cyan Magenta Yellow (CMY)



Luminanza e Crominanze (YUV)

- Luminanza (Y) Crominanze (U e V)
- Separa immagine a **scala di grigio** dalle sue componenti **colore**

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299R + 0.587G + 0.114B \\ -0.147R - 0.289G + 0.436B \\ 0.615R - 0.515G - 0.100B \end{bmatrix}$$

$(Y = k_R R + k_G G + k_B B)$

$k_R + k_G + k_B = 1$

$k_R + k_G + k_B = 0$

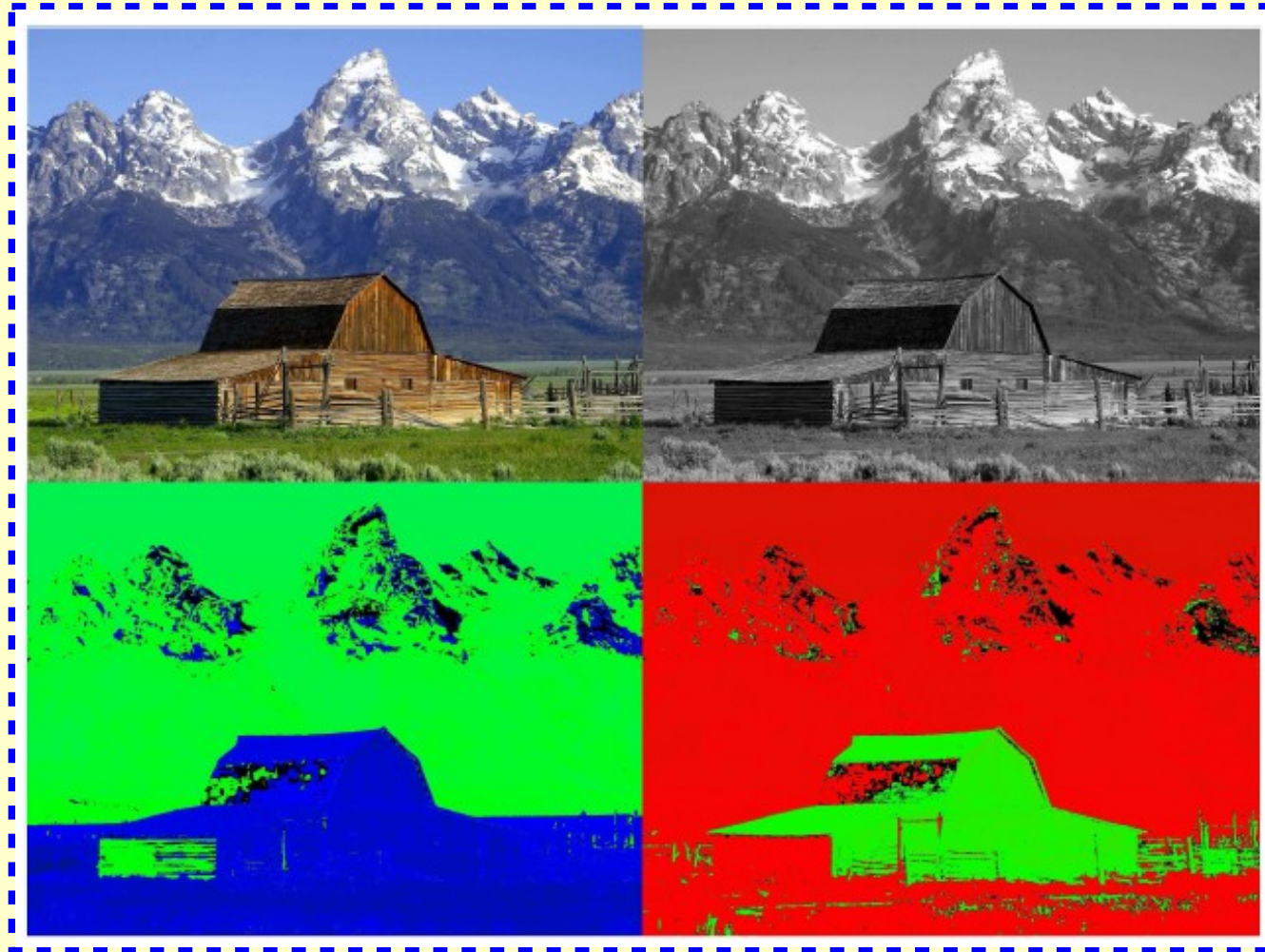
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} Y + 1.140V \\ Y - 0.395U - 0.581V \\ Y + 2.0324U \end{bmatrix}$$

V legata a Rosso

Verde combinazione di U e V

U legata a Blu

Luminanza e Crominanze (YUV)



Luminanza e Crominanze (YPbPr e YCbCr)

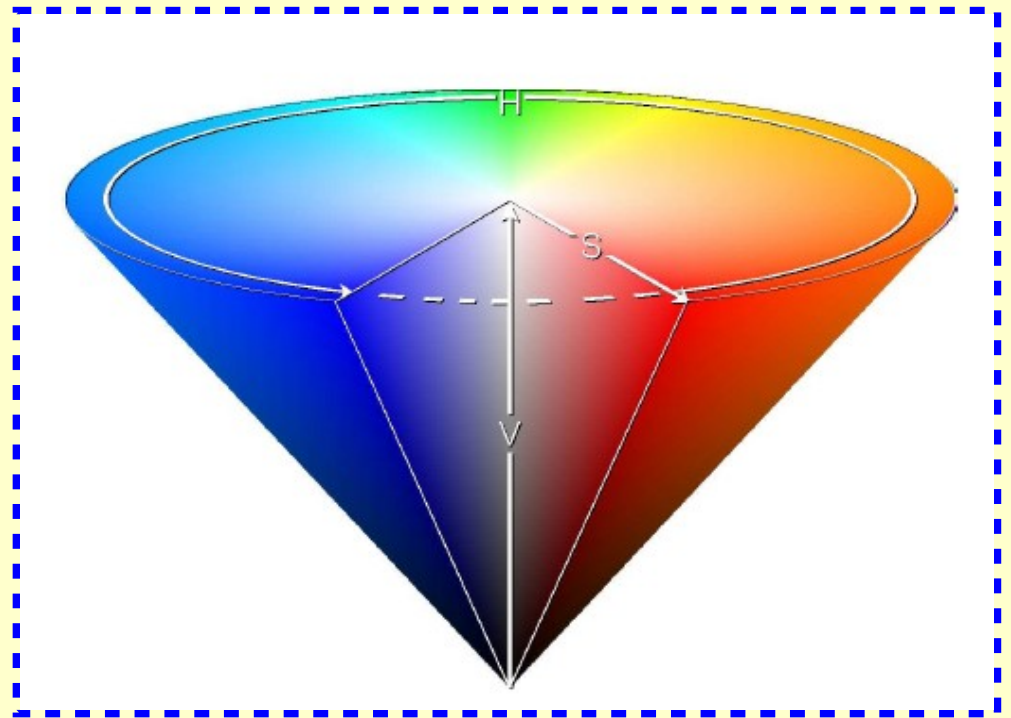
- Formato analogico (YPbPr) e numerico (YcbCr)
- BT.601 → $K_b = 0.114$, $K_r = 0.299$
- BT.709 (HDTV) → $K_b = 0.0722$, $K_r = 0.212$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} K_r R + (1 - K_r - K_b)G + K_b B \\ 0.5 \frac{B - Y}{1 - K_b} \\ 0.5 \frac{R - Y}{1 - K_r} \end{bmatrix}$$

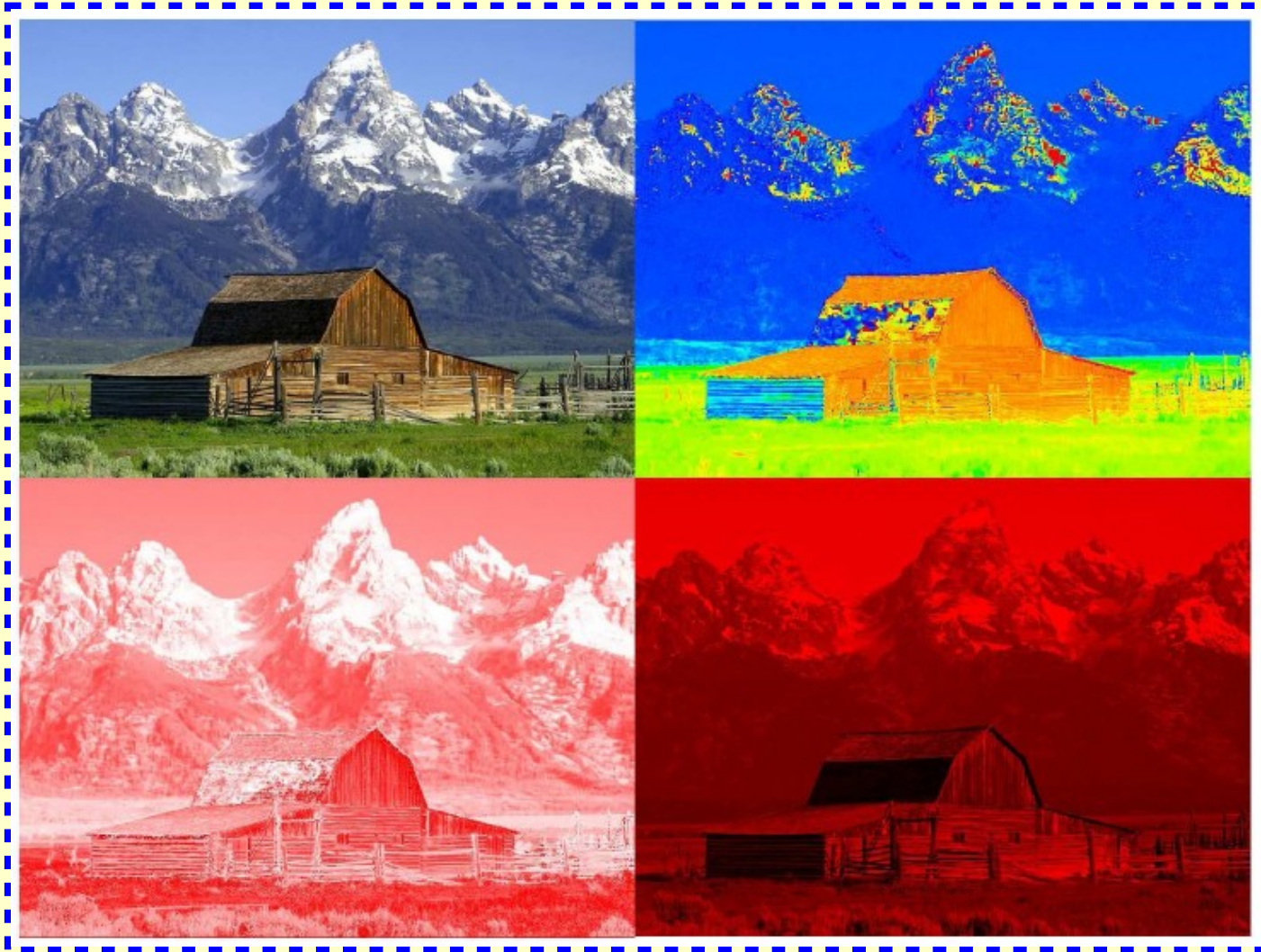
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} Y + 2(1 - K_b)Cb \\ Y - 2 \frac{K_r(1 - K_r)Cr + K_b(1 - K_b)Cb}{1 - K_r - K_b} \\ Y + 2(1 - K_r)Cr \end{bmatrix}$$

Hue Saturation Value (HSV)

- **H (tonalità):** reale colore della sorgente, dove ogni colore ha una diversa frequenza d'onda
 - Gradi [0° - 360°]
- **S (saturazione):** vivicità del colore
 - [0-1] o [0% - 100%]
- **V (luminosità):** quantità di energia, espressa su una scala di grigio
 - [0-1] o [0% - 100%]
- Conversione tra RGB e HSV
fortemente non lineare [\[pag. 30\]](#)

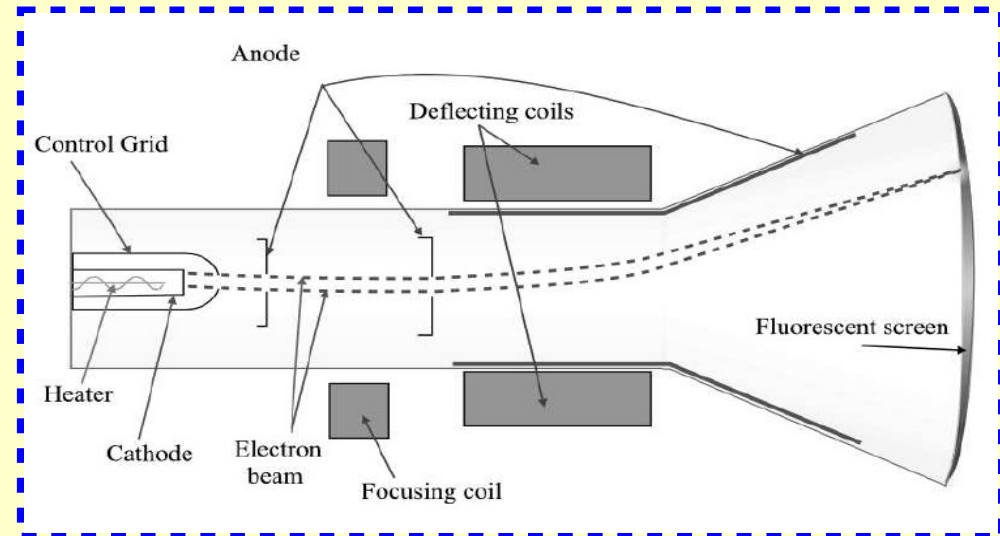


Hue Saturation Value (HSV)

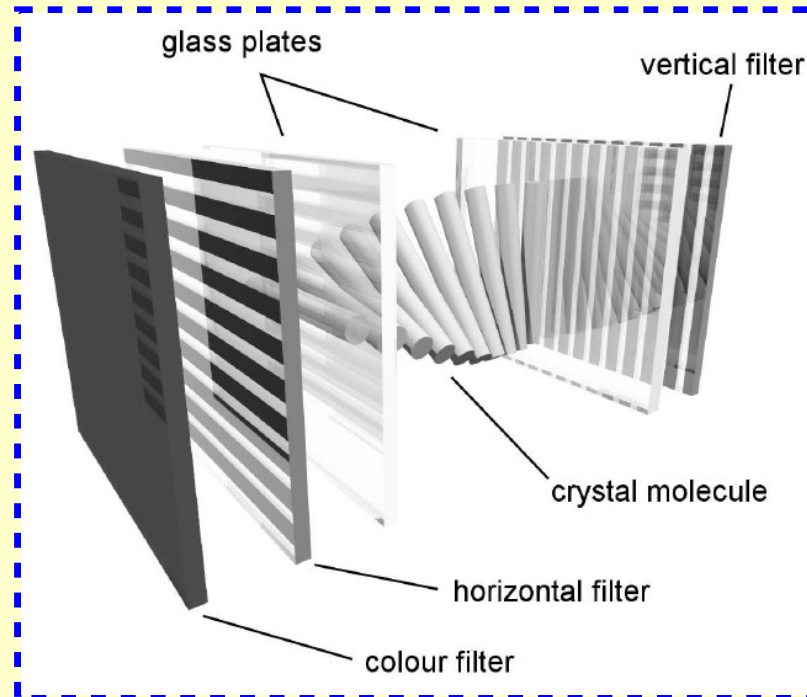


Monitor - CRT

- Fascio di elettroni → accelerati → fosfori → fotoni
- Tempo di persistenza
- Elettrodi per accelerare, modulare intensità di luminosità, deflettere
- Colori (RGB) → tre fasci di elettroni → spot con tre fosfori
- Tecnica interlacciata → quadri successivi composti da sottoinsieme di righe
- Flicker → sfarfallamento
- Standard (frequenza di semiquadro → corrente alternata)
 - **PAL** (Europa)
 - **625 linee (576 visibili)**
 - **4/3 → 720x576**
 - **25 quadri/sec, interlacciato (50 semiquadri/sec)**
 - **NTSC** (America, Giappone)
 - **525 linee (480 visibili)**
 - **4/3 → 640x480**
 - **30 quadri/sec, interlacciato (60 semiquadri/sec)**
 - **SECAM** (Francia)
 - Numero linee analogo a PAL ma diverso formato dei segnali



Monitor - LCD



- **Pixel**
- **Condizioni di riposo**
Cristalli liquidi → rotazione di 90° → pixel trasparente
- **Campo elettrico**
cristalli liquidi si orientano parallelamente al campo → pixel opaco
- **Colori** → tre celle per ogni pixel
- **Thin Film Transistor (TFT)** → transistor CMOS per ogni pixel → maggiore velocità di scansione dello schermo

Monitor – Schede video

- Dispositivo in grado di riprodurre un'immagine su uno schermo
- **Memoria** → pixel
- **Hardware dedicato** → scandire periodicamente memoria → **DAC** → segnali necessari per pilotare video
- S.O. accede alla memoria con un opportuno **driver**
- Componenti aggiuntivi:
 - **Accelerazione grafica** (processore dedicato: disegnare forme geometriche, eseguire primitive come colorare area...)
 - **Piani immagine** (sfondi e soggetti)
- Formati uscita analogici (scheda digitale – segnale in uscita analogico)
 - **YCbCr, video composito, S-video**
- Formati uscita digitale (scheda digitale – segnale uscita digitale)
 - **High Definition Multimedia Interface (HDMI)**

Formati Raw

- Formati di rappresentazione delle immagini senza alcuna compressione

PBM / PGM / PPM

- PBM → portable bitmap file format → **1 bpp (bianco e nero)**
- PGM → portable graymap file format → **8 bpp (scale di grigio)**
- PPM → portable pixmap file format → **24 bpp (colori)**
- Due metodi di rappresentazione dati: ASCII e binario
- **File HEADER:**
 - **Tipologia** di file: P1, P2, P3, P4, P5, P6 (ascii/binari PBM, PGM, PPM)
 - **Dimensione** immagine: numero pixel orizzontali e verticali
 - Numero massimo di **valori** per singola componente RGB o di scala di grigio
- **File DATI:**
 - Informazioni per ogni pixel

Esempio PBM ASCII (wikipedia)

PBM example

A simple example of the PBM format is as follows (There is a newline character at the end of each line.):

```
P1
# This is an example bitmap of the letter "J"
6 10
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
1 0 0 0 1 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

The string *P1* identifies the file format. The hash sign introduces a comment. The next two numbers give the width and height of the image (in this case, 6 and 10 pixels, respectively). The rest of the file is a single row of numbers (in this case, only zeros and ones).

Here is the resulting image: J



Here it is again magnified 20 times:



Esempio PGM ASCII (wikipedia)

PGM example

The PGM and PPM formats (both ASCII and binary versions) have an additional parameter for the maximum value (numbers of grey dimensions and before the actual pixel data. Black is 0 and max value is white. There is a newline character at the end of each line.

```
P2
# Shows the word "FEED" (example from Netpbm main page on PGM)
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 0 11 11 11 11 0 0 15 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Example (magnified):



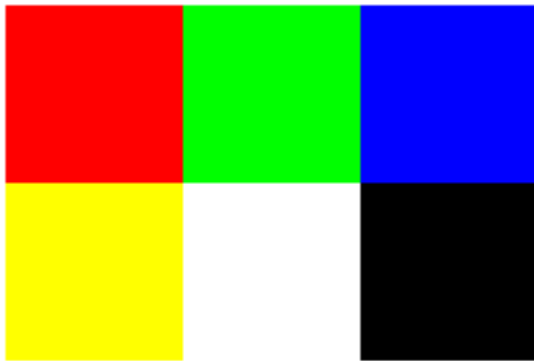
Esempio PPM ASCII (wikipedia)

PPM example

This is an example of a color RGB image stored in PPM format. There is a newline character at the end of each line.

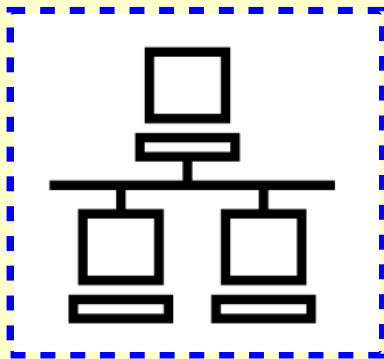
```
P3
# The P3 means colors are in ASCII, then 3 columns and 2 rows,
# then 255 for max color, then RGB triplets
3 2
255
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```

The image (magnified):



X BitMap (XBM)

- Formati grafici di cursori per il sistema X
- Formato di memorizzazione → ASCII
- Bianco e nero
- Porzione di codice scritto in C
 - 2 clausole **#define** → numero pixel orizzontali e verticali
 - **Allocazione dei bit spenti e accesi** → 8 pixel rappresentati come byte in formato esadecimale



```
#define network.bm_width 32
#define network.bm_height 32
static char network.bm_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x0f, 0x00, 0x00, 0x08, 0x08, 0x00,
    0x00, 0x08, 0x08, 0x00, 0x00, 0x08, 0x08, 0x00, 0x00, 0x08, 0x08, 0x00,
    0x00, 0x08, 0x08, 0x00, 0x00, 0x08, 0x08, 0x00, 0x00, 0xf8, 0x0f, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0x1f, 0x00, 0x00, 0x04, 0x10, 0x00,
    0x00, 0xfc, 0x1f, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00,
    0xfe, 0xff, 0xff, 0x7f, 0x00, 0x01, 0x80, 0x00, 0x00, 0x01, 0x80, 0x00,
    0xf0, 0x1f, 0xf8, 0x0f, 0x10, 0x10, 0x08, 0x08, 0x10, 0x10, 0x08, 0x08,
    0x10, 0x10, 0x08, 0x08, 0x10, 0x10, 0x08, 0x08, 0x10, 0x10, 0x08, 0x08,
    0x10, 0x10, 0x08, 0x08, 0xf0, 0x1f, 0xf8, 0x0f, 0x00, 0x00, 0x00, 0x00,
    0xf8, 0x3f, 0xfc, 0x1f, 0x08, 0x20, 0x04, 0x10, 0xf8, 0x3f, 0xfc, 0x1f,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

prima riga

seconda riga

