

## Costruzione di Interfacce

### Lezione 2

Paolo Cignoni

cignoni@iei.pi.cnr.it  
<http://vcg.iei.pi.cnr.it/~cignoni/CI>

## Architettura di un renderer

- ❖ La pipeline di rendering; assumendo che
  - ❖ La scena è composta di entità geometriche semplici (primitive) descritte per mezzo di vertici
  - ❖ L'algoritmo di rendering che voglio usare è strutturato in maniera da processare e disegnare tutte le primitive una alla volta abbastanza indipendentemente (object order)
- ❖ Allora per ogni primitiva le operazioni da fare sono, in sequenza, le seguenti



## Pipeline di rendering

- ❖ Il fatto di strutturare il rendering
  - ❖ Indipendentemente per primitiva
  - ❖ Per ogni primitiva in una pipeline ben determinata
- ❖ Permette di progettare hw grafico che espliciti il parallelismo nei due livelli
  - ❖ Multiple rendering pipelines
  - ❖ I passi più lenti della pipeline possono essere parallelizzati più massicciamente

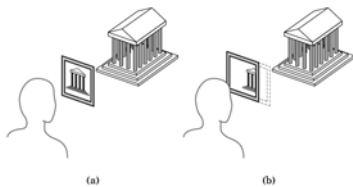


## Transformazioni di modellazione

- ❖ Ogni oggetto nella scena ha, di solito il proprio sistema di riferimento
- ❖ I vertici della scena da rendere devono essere trasformati in un unico sistema di riferimento: quello della camera.

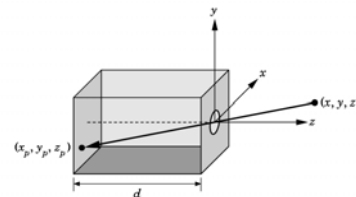
## Lighting e Clipping

- ❖ Dopo la trasformazione di modellazione si può decidere che cosa è visibile per la camera corrente (e quindi interrompere la pipeline per ciò che non è visibile)



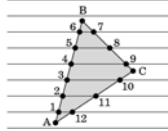
## Proiezione

- ❖ Si calcola dove ogni vertice cade nel piano di proiezione



## Rasterizzazione

- ❖ Per ogni primitiva a questo punto sappiamo dove finiscono nel frame buffer i suoi vertici.
- ❖ Il processo di trovare tutti i pixel che nel frame buffer appartengono alla primitiva è detto rasterizzazione.

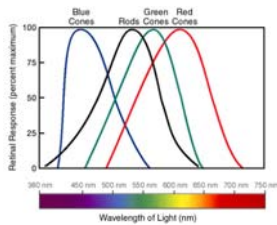


## Caveat

- ❖ Sulla pipeline di rendering torneremo più volte
- ❖ Gli step possono essere ben più dettagliati
- ❖ Non tutti gli step, non su tutti gli hw, sono implementati effettivamente in hw
- ❖ Questa pipeline di rendering NON è l'unica esistente

## Colore

- ❖ Luce e' una forma di radiazione elettromagnetica
- ❖ La retina umana ha tre tipi di recettori, i coni, sensibili a particolari lunghezze d'onda



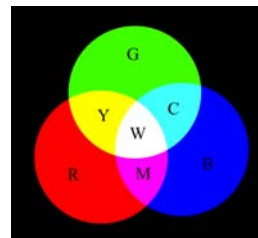
## Modelli di colore

- ❖ Gli spazi colore o modelli colore sono dei sistemi particolari di coordinate che consentono di definire all'interno di un sottoinsieme di colori (detto *gamut*) un particolare elemento
- ❖ A noi interessano gli spazi: RGB (monitor, sintesi additiva), CMYK (stampe, sintesi sottrattiva), HSV (intuitivo)

## RGB

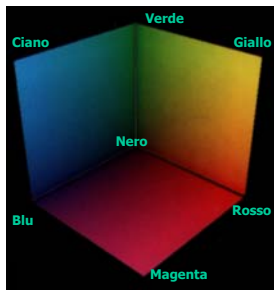
- ❖ Il gamut dei monitor a raggi catodici (CRT), è definito dalle primarie rosso, verde e blu o RGB (dall'inglese Red, Green, Blue)
- ❖ I monitor sintetizzano i colori eccitando tre tipi di fosfori (RGB), per cui un colore è ottenuto miscelando parti diverse di queste tre primarie

## RGB



- ❖ Queste tre componenti si mescolano additivamente:
- ❖ L'area in cui si sovrappongono due componenti il colore si somma (entrambi i recettori vengono stimolati)


## RGB



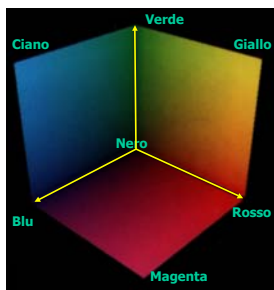
- ❖ La rappresentazione tipica dello spazio RGB è in forma di cubo
- ❖ Sui vertici si trovano il bianco, il nero, le primarie e i complementari

## RGB

- ❖ I colori ciano, magenta e giallo sono detti complementari dei colori primari
- ❖ Ad esempio, il ciano è il complementare del rosso poiché deriva dalla sottrazione del rosso (1,0,0) dal bianco (1,1,1):

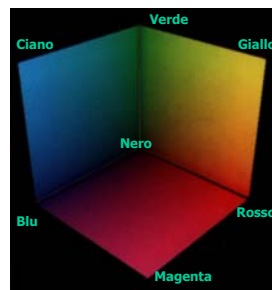
$$\begin{aligned} \text{bianco } (1,1,1) &- \\ \text{rosso } (1,0,0) &= \\ \text{ciano } (0,1,1) & \end{aligned}$$


## RGB



- ❖ Le tre coordinate RGB variano ciascuna da 0 (intensità minima) a 1 (intensità massima)

## RGB



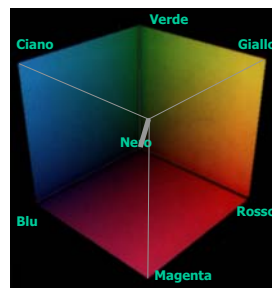
- ❖ Il colore nero di un pixel si otterrà *spegnendo* tutti e tre i fosfori (cioè  $R=0, G=0, B=0$ ), il colore bianco *accendendo* i fosfori al massimo (cioè  $R=1, G=1, B=1$ )

## Esempio

- ❖ In RGB il colore rosso è dato dalla tripletta ( $R=1, G=0, B=0$ ); variando R da 0 a 1 (mantenendo costanti G e B a 0) si ottengono tutti i rossi puri di diversa luminosità (da poco a molto luminoso)



## RGB



- ❖ La diagonale che unisce il nero con il bianco è detta linea dei grigi
- ❖ Infatti un grigio ha la caratteristica di avere tutte le tre componenti uguali, ad esempio, (0.5,0.5,0.5) è un grigio

## CMY(K)

- ❖ Ciano, Magenta e Giallo (Cyan, Magenta, Yellow CMY) sono i colori complementari di Rosso, Verde e Blu
- ❖ Quando vengono usati come filtri per sottrarre colore dalla luce bianca, questi colori sono chiamati **primarie sottrattive**

## CMY(K)



- ❖ Il modello CMY dal punto di vista geometrico è lo stesso di RGB con la differenza che, in questo caso, il bianco è l'origine (al posto del nero) e i colori sono definiti per sottrazione dalla luce bianca, anziché per addizione

## CMY(K)

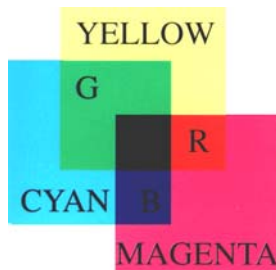
- ❖ Il modello CMY è usato nei dispositivi di stampa a colori (stampanti laser, *ink-jet*, a sublimazione, elettrostatiche) dove l'inchiostro colorato funziona come un filtro che sottrae alcune frequenze dal bianco del foglio

## CMY(K)



- ❖ Ad esempio, un inchiostro ciano depositato su un foglio bianco riflette tutti i colori ad eccezione del rosso (in termini di primarie additive, ciano è dato da bianco - rosso o da verde + blu)

## CMY(K)



- ❖ Aree in cui si sovrappongono ciano e magenta riflettono tutti i colori ad eccezione del rosso e del verde: quindi appaiono blu!

## CMY(K)

- ❖ La relazione esistente tra CMY e RGB è definita dalle semplici formule

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

- ❖ Usando questo modello per ottenere una superficie nera dobbiamo evitare che rifletta tutti i primari (rosso, verde e blu), dobbiamo quindi colorarla di ciano, magenta e giallo alla massima intensità

## CMY(K)

- ❖ Nei dispositivi di stampa a colori si è pensato di aggiungere ai tre inchiostri CMY del vero e proprio inchiostro nero (detto colore K) per due motivi:
  - ❖ mettendo insieme C, M e Y non si ottiene un *nero puro* poiché i tre inchiostri non sono filtri perfetti
  - ❖ l'inchiostro nero costa meno di quelli colorati!

## CMY(K)

- ❖ Quindi, anziché usare parti uguali di C, M e Y si usa K
- ❖ Si ha così il cosiddetto modello CMYK
- ❖ Si passa da CMY a CMYK con le formule:

$$\begin{aligned}K &= \min(C, M, Y) \\ C &= C - K \\ M &= M - K \\ Y &= Y - K\end{aligned}$$

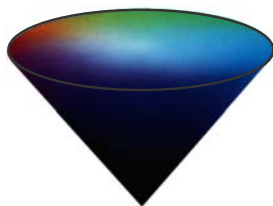
## HSV

- ❖ Sia RGB che CMYK sono modelli *hardware-oriented*, destinati a semplificare la descrizione dei colore utilizzando dispositivi di visualizzazione o stampa
- ❖ Per un operatore umano non esperto selezionare un rosa, un viola o un marrone (in questi spazi) è un'impresa molto faticosa e necessita di numerosi tentativi

## HSV

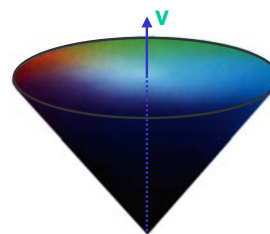
- ❖ Per questo problemi si introduce il modello HSV (Hue, Saturation, Value)
- ❖ Il modello nasce dall'idea di cercare di riprodurre il modo con cui un pittore prepara un suo colore sulla tavolozza: prende un colore puro e aggiunge del bianco per ottenere una tinta; poi aggiunge del nero per cambiare la luminosità ed ottiene un tono

## HSV



- ❖ La sua tipica rappresentazione geometrica è su un sistema di coordinate cilindriche come cono (o prisma a base esagonale) con il vertice rivolto verso il basso

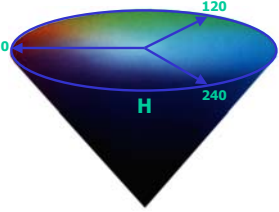
## HSV



- ❖ La coordinata V (*Value*) corrisponde alla luminosità e assume valori nell'intervallo da 0 (scuro) a 1 (chiaro)
- ❖ V è rappresentata dall'asse verticale

### HSV

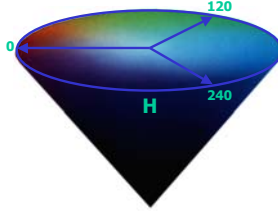
---



- ❖ La coordinata H (*Hue*) corrisponde al colore ed è la misura dell'angolo attorno all'asse verticale (V)
- ❖ Il rosso vale 0°, il verde vale 120° e il blu 240°

### HSV

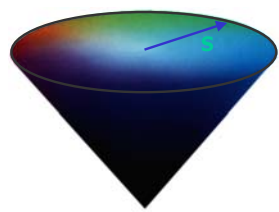
---



- ❖ I colori complementari sono opposti (+180°) ai primari

### HSV

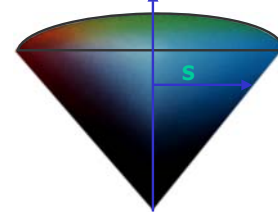
---



- ❖ La coordinata S (*Saturation*) corrisponde al livello di saturazione ed è rappresentato da un asse orizzontale avente angolo H
- ❖ S varia da 0 (bianco o completamente desaturo) a 1 (colore puro o completamente saturo)

### HSV

---



- ❖ La coordinata S (*Saturation*) corrisponde al livello di saturazione ed è rappresentato da un asse orizzontale avente angolo H
- ❖ S varia da 0 (bianco o completamente desaturo) a 1 (colore puro o completamente saturo)

### Caveat

---

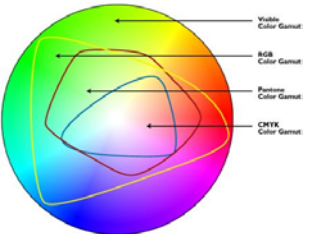
Gli spazi dei colori sono un astrazione  
In realtà ci sono vari aspetti da tenere in considerazione

- ❖ I device di output non possono visualizzare tutti i colori che possiamo vedere.
- ❖ Limitazioni sull'insieme dei colori sintetizzabili
- ❖ Non linearità nel mappaggio di colori dallo spazio in cui sono definiti a quelli reali

### Gamut

---

- ❖ Ogni device può sintetizzare solo un sottoinsieme di colori detto *gamut*



## Gamma Correction

- ❖ I device fisici mappano i colori in maniera fortemente non lineare:
- ❖ in generale non e' vero che il grigio RGB (.5,.5,.5) e' luminoso la metà di (1,1,1)
- ❖ Gamma correction si mappa ogni colore  $(r, g, b)$  in  $(r', g', b')$  dove  $\gamma$  è un valore nel range  $0.2 \sim 4$

