

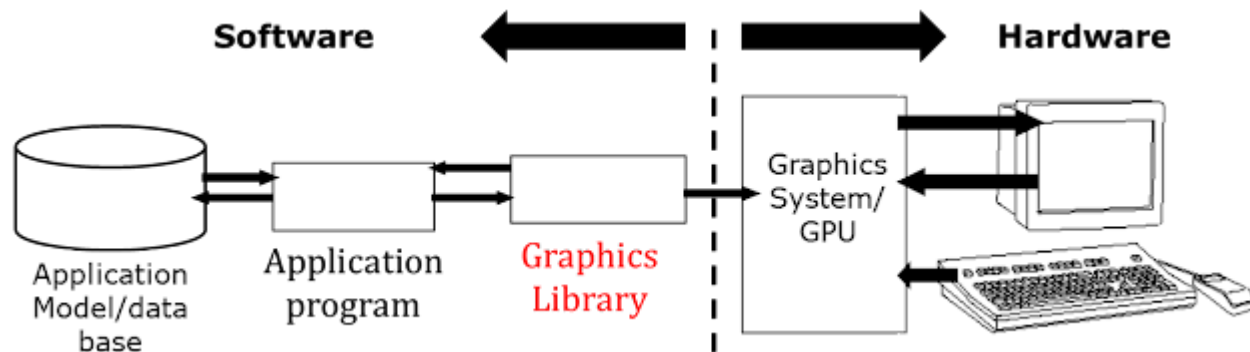


Il sistema grafico



Sistema Grafico

- **Sistema grafico**: dispositivi **hardware** e programmi **software** che interagiscono per produrre immagini grafiche.
 - I **dispositivi hardware** rappresentano la potenzialità del sistema grafico: le risorse che rendono possibile produrre immagini.
 - Il **software** serve ad organizzare le risorse hardware: utilizza l'hardware per creare e manipolare immagini grafiche..
-



Il programma applicativo si interfaccia all'hardware grafico mediante funzioni che appartengono ad una libreria grafica (OpenGL, DirectX, etc), che prendono il nome di *Application Programmer's Interface (API)*

I drivers della scheda Video sono software che permettano al sistema operativo di comunicare con la scheda video: interpretano l'output delle API e lo convertono in una forma che è riconoscibile per la scheda video.

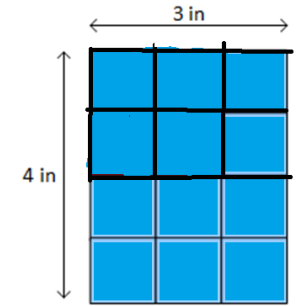


-
- Lo sviluppo della Computer Graphics è parallelo a quello della tecnologia hardware; quindi per poter conoscere e sfruttare fino in fondo le sue potenzialità è necessario avere una **visione chiara di come si è evoluto nel tempo l'hardware grafico** e delle conseguenze di quest'evoluzione nella pipeline grafica.
 - I due aspetti peculiari dell' hardware di un sistema grafico sono i dispositivi periferici di input/output e le schede grafiche.
 - I principali dispositivi periferici impiegati in computer grafica sono
 - **dispositivi di input grafico interattivo**
 - mouse
 - joystick, trackball
 - Scanner 2D/3D
 - VR glasses, kinect, leap
 - **Raster scan Display System** : Display, printers, Digital Camera.
 - **Schede Grafiche**
-



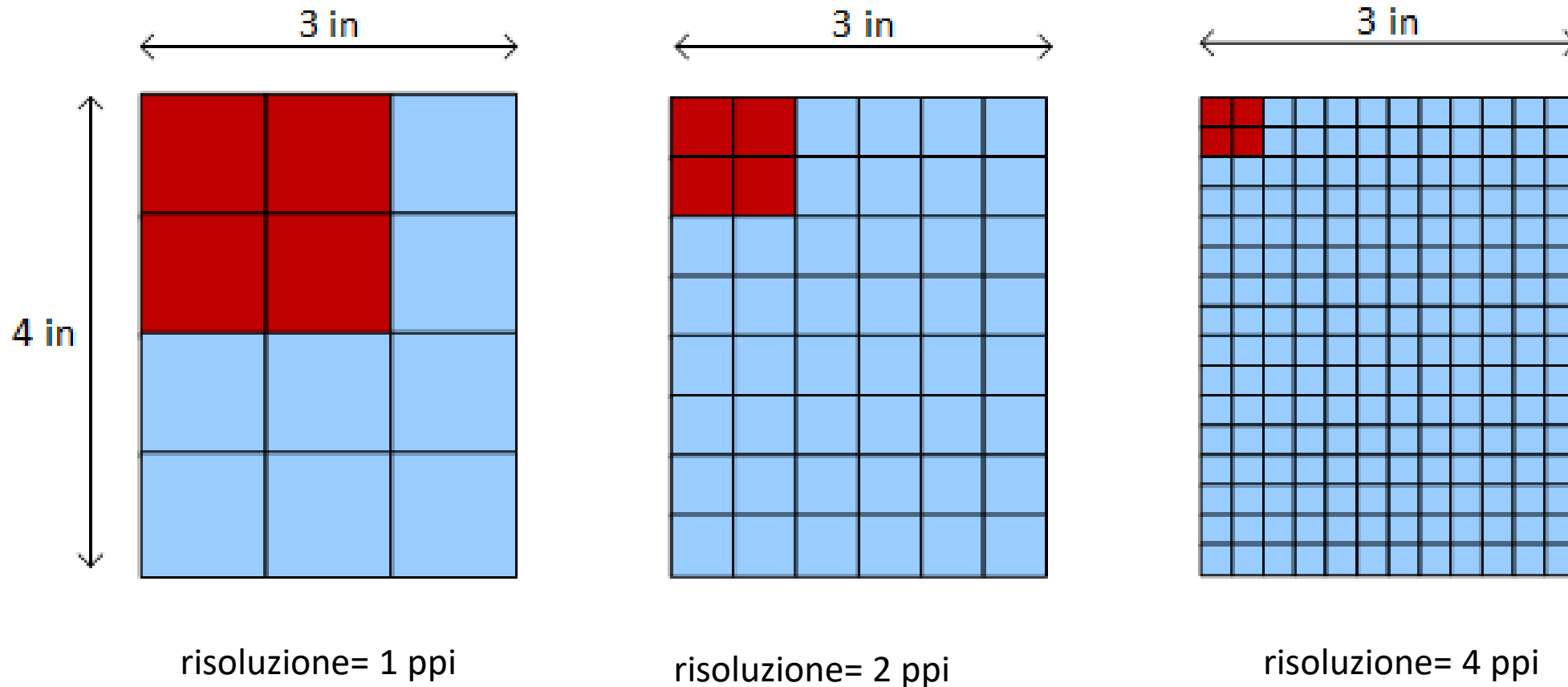
Raster devices

- Un raster device è composto da una matrice rettangolare di campioni o **pixel** (picture element).
- La **risoluzione** di un dispositivo raster misura la **densità dei pixel**: è il numero di pixel su unità di distanza o area .
- Video: inch – pixel per inch – **ppi** – oppure pixel per centimetro – ppc
- **Stampanti/scanner**: dot per inch - **dpi**
- Per esempio a risoluzione 300 pixel per pollice (ppi, pixel per inch), in un pollice lineare ci sono 300 pixel, mentre in un pollice al quadrato ci sono 300x300 pixel.
(1 pollice = 2.54 cm)
- La **risoluzione display** è il numero di pixel in ogni dimensione (es. 1024 × 768).
- La **dimensione fisica** di un dispositivo raster : larghezza x altezza (in centimetri o pollici)
- **dimensione fisica = risoluzione display / risoluzione.**



risoluzione= 1 ppi

Un raster device è un dispositivo che utilizza una griglia di punti, o pixel, per creare e visualizzare immagini.



Dispositivi che hanno la **stessa larghezza e altezza**, ma **risoluzioni diverse**. Di conseguenza, il quadrato rosso (larghezza=2pixel e altezza=2pixel) è progressivamente più piccolo negli schermi rettangolari via via che aumenta la risoluzione.



W=widescreen

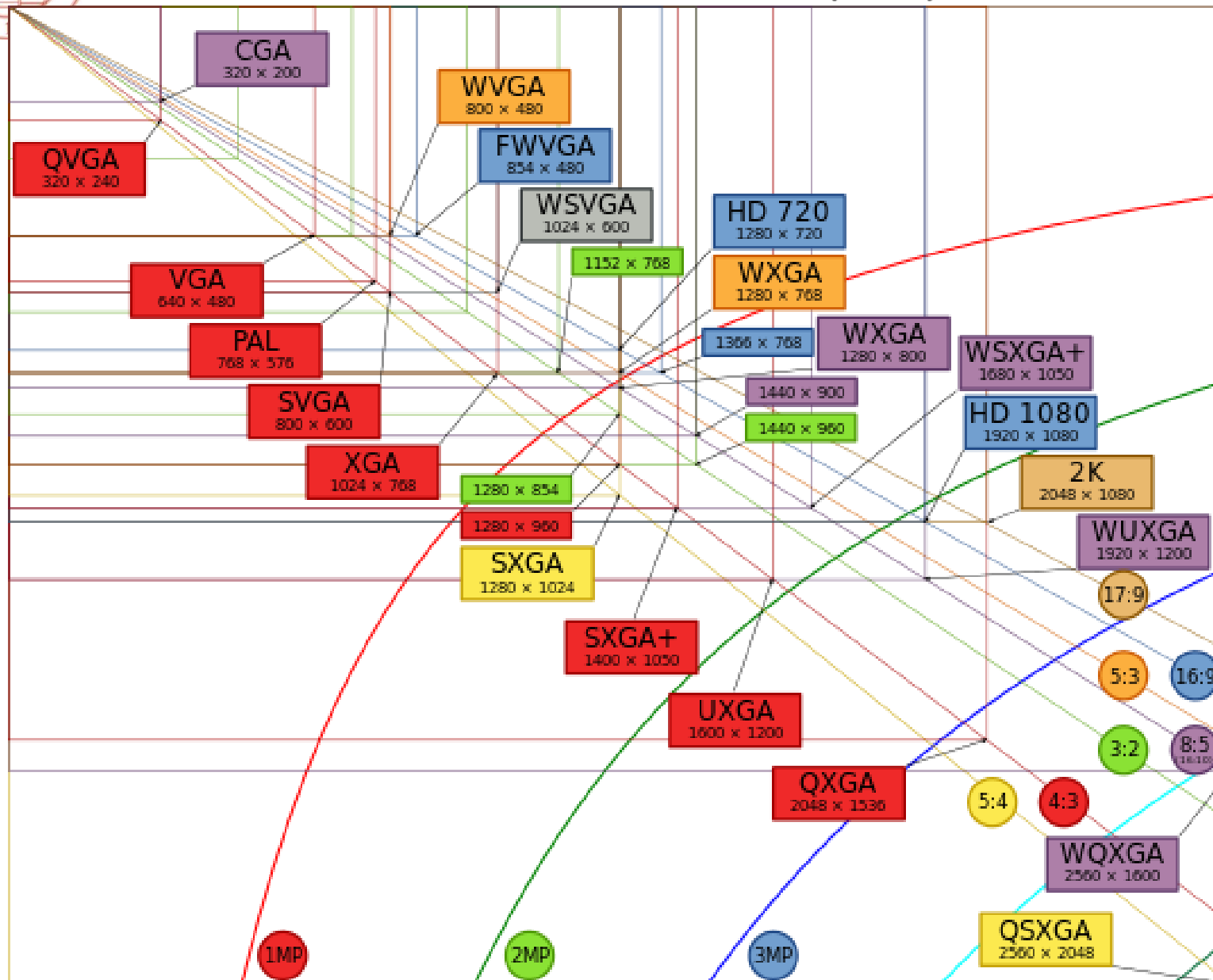
X=eXtended

S=Super

U=Ultra

H=High Definition

Standard display resolution



Quarter (Q)

Un quarto della risoluzione di base. Ad esempio QVGA indica una risoluzione di 320×240 che è la metà della larghezza e la metà dell'altezza, ossia un quarto del totale dei pixel, della risoluzione VGA di 640×480.

Nelle alte risoluzioni Q significa "Quad" ossia quattro volte.

Wide (W)

Risoluzione più ampia di quella base, di cui mantiene il numero di righe, che permette un rapporto d'aspetto di 16:9 o 16:10.

Quad(ruple) (Q)

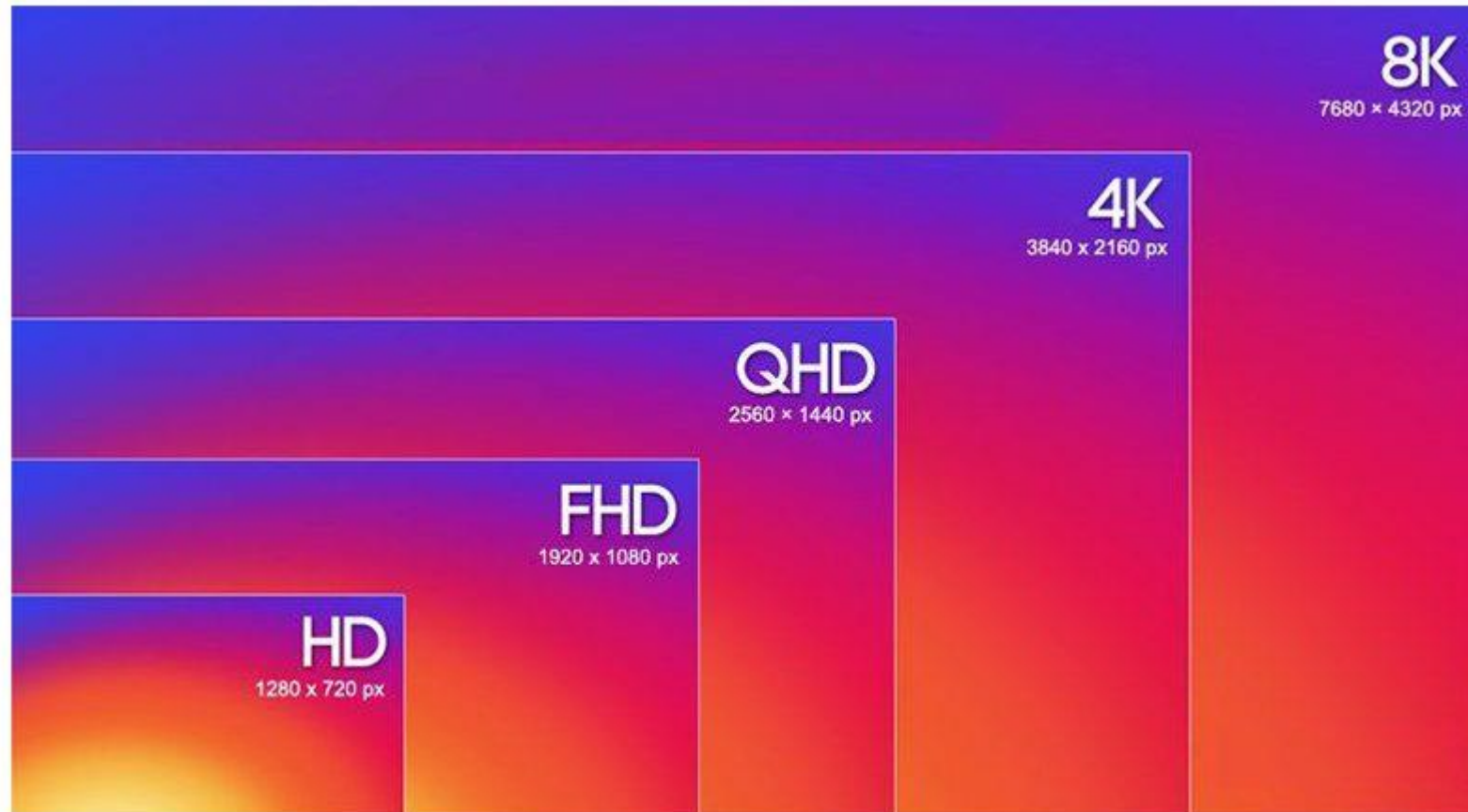
Quattro volte i pixel della risoluzione di base ossia due volte in orizzontale e due volte in verticale il numero di pixel.

Hexadecatuple (H)

Sedici volte i pixel della risoluzione di base ossia quattro volte in orizzontale e quattro volte in verticale il numero di pixel.



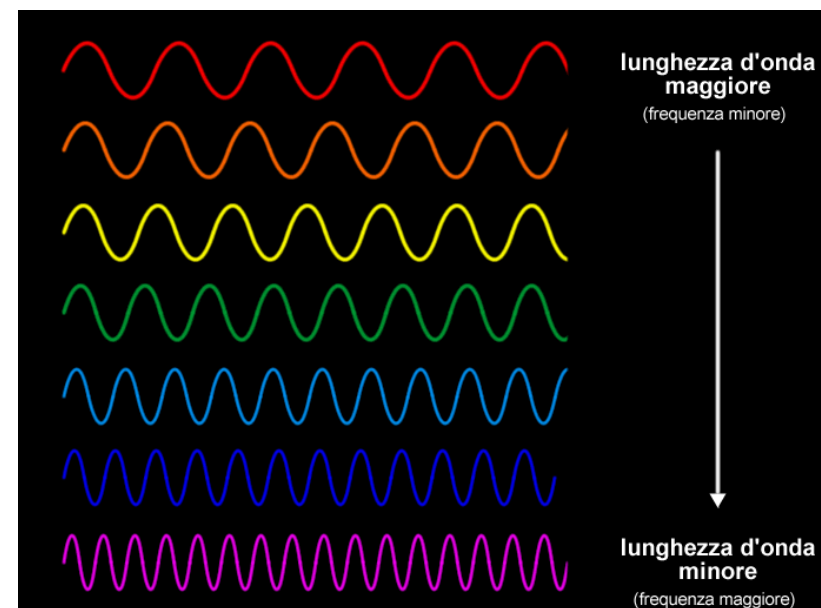
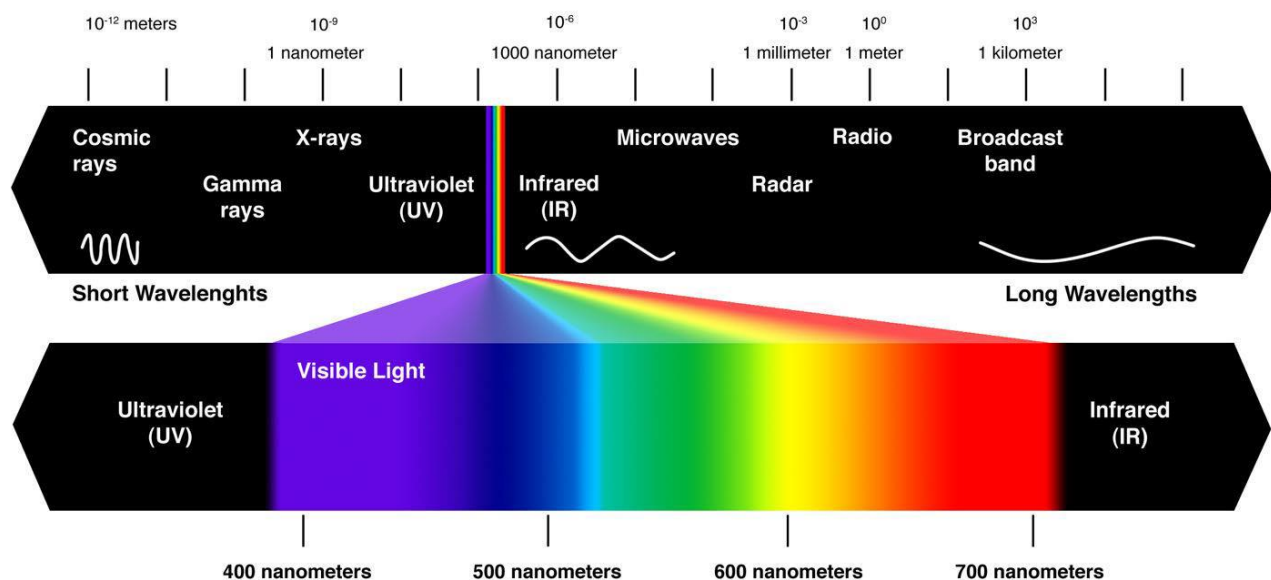
Standard display resolution



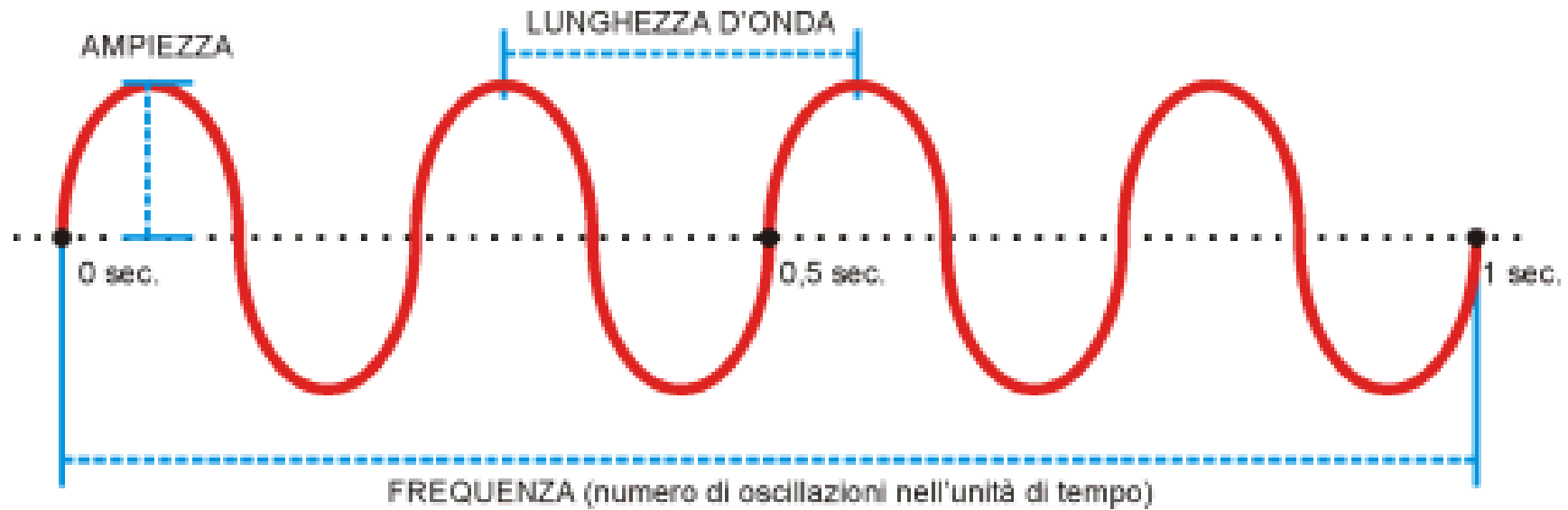


Luce e colori

- Il sistema visivo umano percepisce una parte molto limitata delle radiazioni elettromagnetiche, quelle con lunghezze d'onda compresa tra 400 nanometri (ultravioletti) e 800 nm (infrarossi)



La sensazione fisiologica che il sistema visivo umano produce **quando l'occhio viene colpito dalla luce** riflessa, emessa o trasmessa da un oggetto, viene definita **come colore**.



Lunghezza d'onda: distanza tra due massimi o due minimi dell'onda

Ampiezza: la distanza tra un massimo e il piano mediano che interseca l'onda

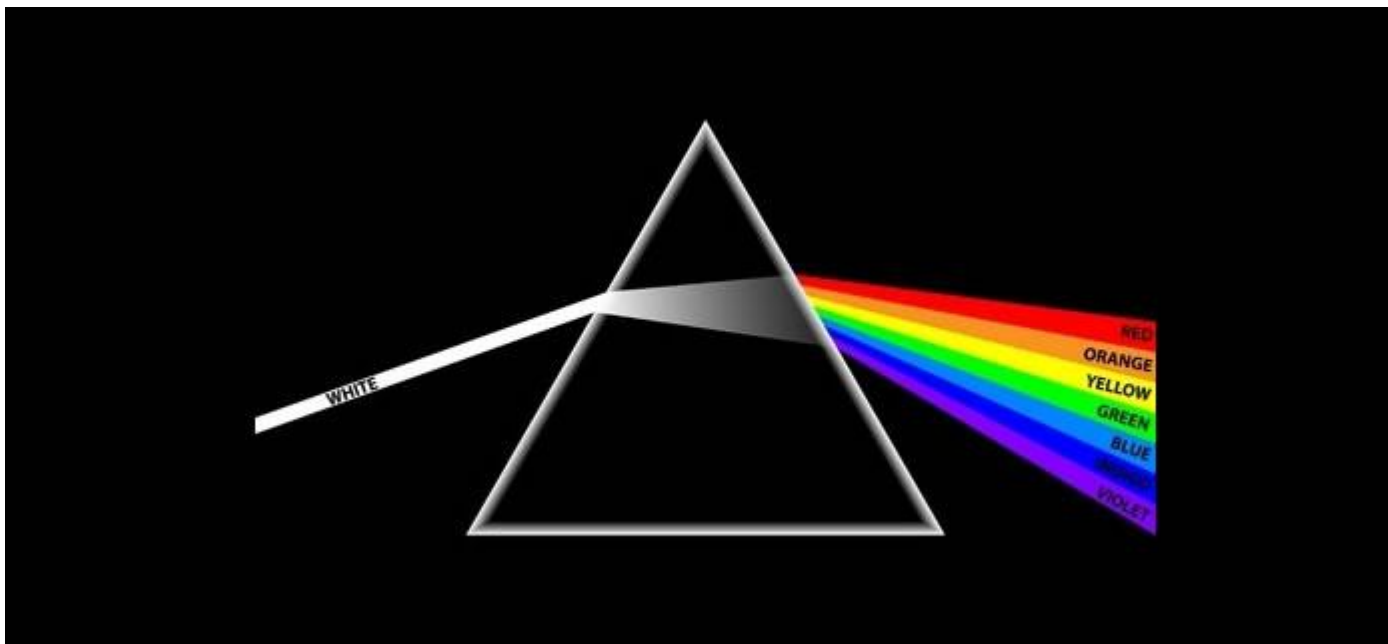
Frequenza: numero di oscillazioni nell'unità di tempo.

l'ampiezza dell'onda influisce **sull'intensità luminosa** dello stimolo elaborato dal cervello, mentre la **lunghezza dell'onda** influenza la **tonalità del colore percepito**: ad esempio, un'onda elettromagnetica di lunghezza compresa tra i 650 e i 700 nanometri (nm) provoca in un soggetto normodotato la visione del colore rosso.



Prisma di Newton

La luce bianca è composta da tutti i colori dello spettro



La luce bianca entra nel prisma e lo attraversa. Quando la luce fuoriesce dal prisma, viene scomposta nei sette colori fondamentali.

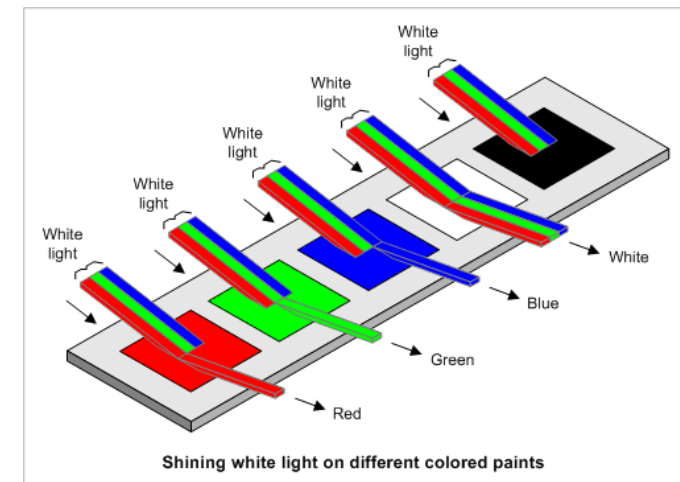
Gli oggetti posseggono delle particelle, chiamate **pigmenti colorati**, che sono responsabili dell'assorbimento selettivo della luce.

Gli oggetti appaiono di un certo colore perché assorbono (o sottraggono) tutti i colori della luce visibile tranne il colore che viene riflesso all'occhio

1. Una mela rossa assorbe tutte le lunghezze d'onda della luce eccetto il rosso, che riflette
2. Una palla blu assorbe tutte le lunghezze d'onda della luce tranne il blu, che riflette
3. Un filtro giallo assorbe tutte le lunghezze d'onda della luce, eccetto il giallo, che trasmette.

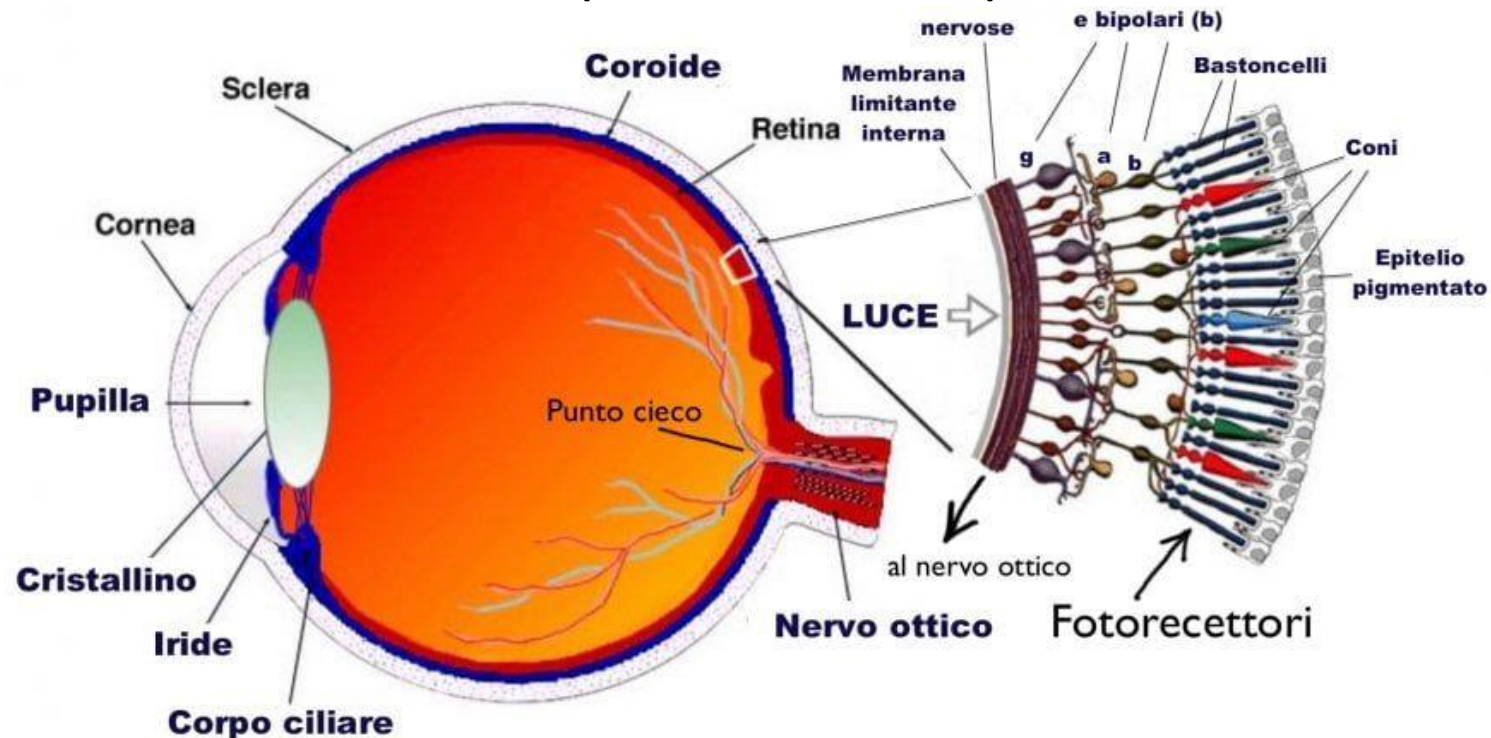
In altre parole la luce incidente, sulla superficie dell'oggetto, viene in parte riflessa e in parte no.

La componente non riflessa penetra all'interno dell'oggetto e viene in parte assorbita dai pigmenti colorati che restituiscono solo le componenti spettrali non assorbite (luce diffusa)





- **coni**: sono sensibili alla lunghezza d'onda (percezione del colore)
- **bastoncelli**: hanno lo scopo di adattarsi ai cambiamenti di intensita' di luce ad esempio alla luce crepuscolare e notturna





Teoria tricromatica di Young

- Ai primi dell'800, il medico inglese **Thomas Young** ritenne **impossibile** per l'esiguità della superficie della retina, **che in ogni punto potessero esistere infiniti fotoricettori diversi quanti i colori discriminati dall'occhio.**
 - **Teorizzò** quindi l'esistenza di **tre soli tipi di recettori**, associati ai tre colori primari pittorici: **giallo, magenta, e ciano.**
 - I tre tipi di fotorecettori, ognuno sensibile ad un colore primario, rilevavano tre diversi stimoli (da cui il nome di **modello tristimolo**), che il cervello poi univa generando il colore.
-



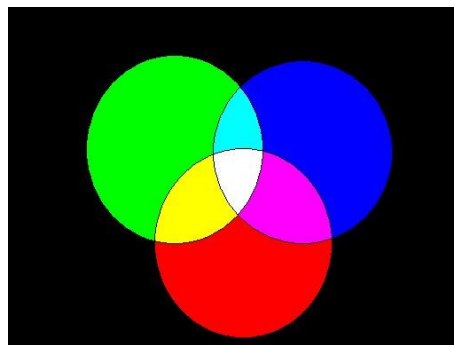
Modelli di Colore

Un modello di colore è un modello matematico astratto che descrive il modo in cui i colori possono essere rappresentati come tuple di numeri, tipicamente come tre o quattro valori di componenti di colore.

I principali modelli per la rappresentazione del colore in un'immagine digitalizzata sono i seguenti:

1. Modello RGB (red, green, blu)

Sintesi additiva: il colore risultante è ottenuto sommando i contributi luminosi dei tre colori primari.

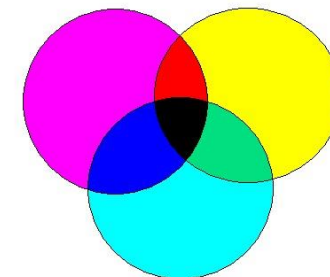


RGBA: è RGB con un canale aggiuntivo, alfa, per gestire la trasparenza

2. Modello CMY (cyan, magenta, yellow)

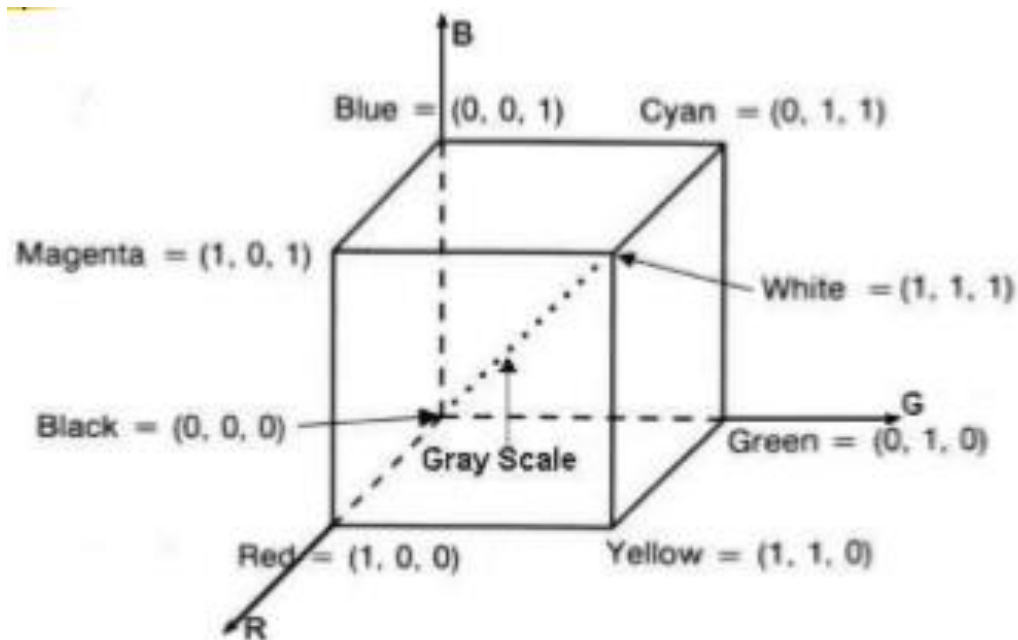
Sintesi sottrattiva: il colore risultante è ottenuto rimuovendo dalla luce bianca i contributi dei 3 colori primari (stampanti a colori). Gli oggetti appaiono colorati in un certo modo soltanto perchè le sostanze che li costituiscono assorbono dalla radiazione incidente l'energia associata a certe lunghezze d'onda.

- $(CMY) = (1 \ 1 \ 1) - (RGB)$

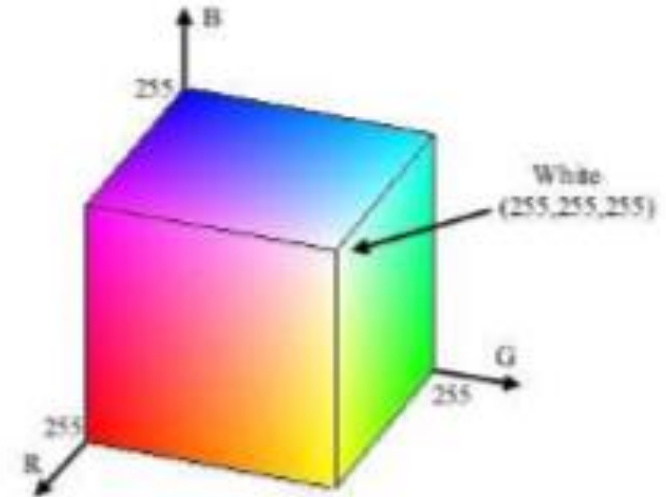
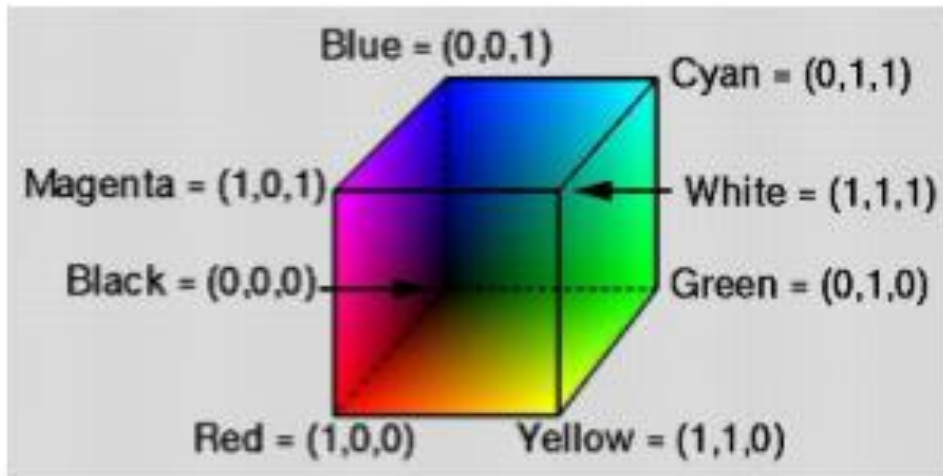


Modello RGB

- Nella rappresentazione 3D del modello colorimetrico possiamo associare ad ogni asse un colore primario (red, green, blue). La rappresentazione del modello RGB è il cubo unitario

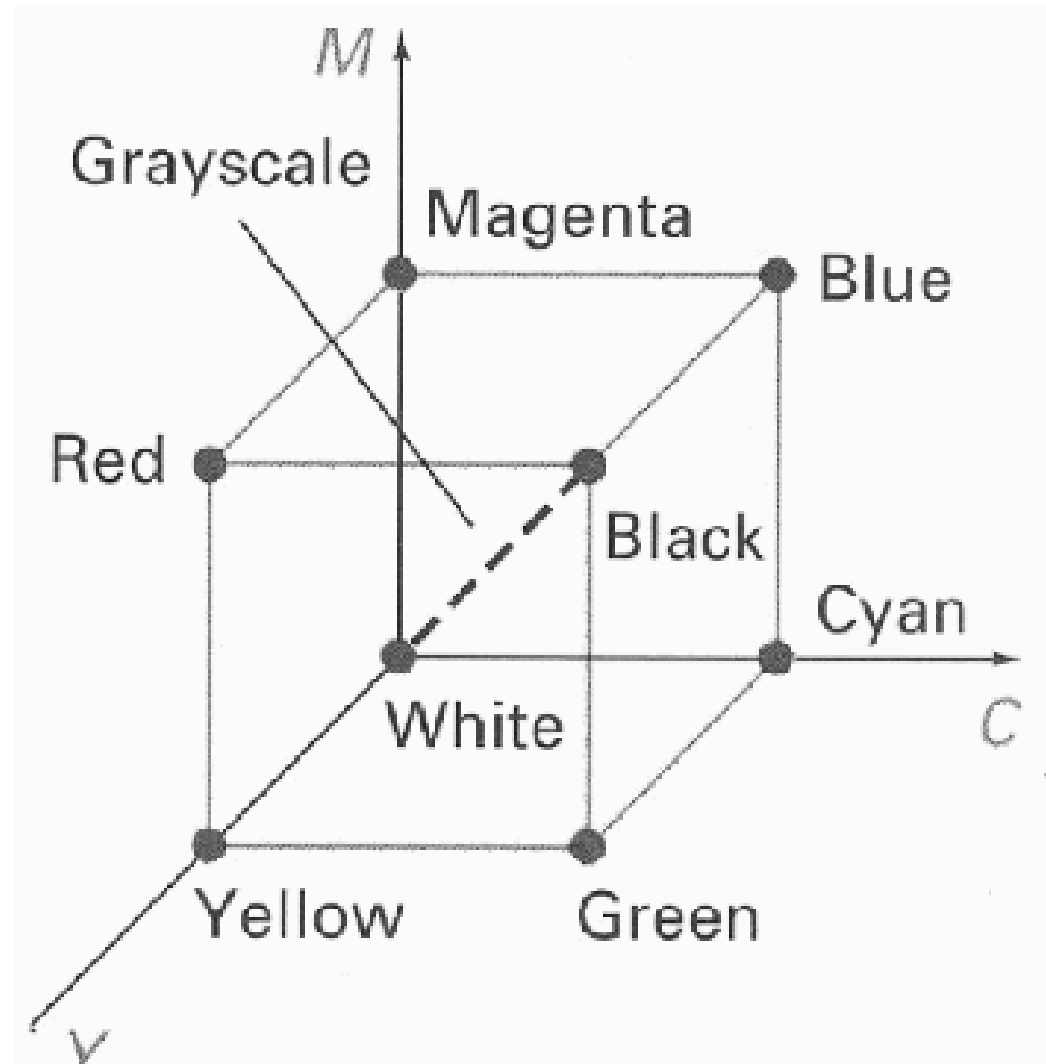


I grigi sono sulla diagonale del cubo unitario; un grigio è rappresentato da una terna con tutti i valori uguali.





Modello CMY





Spazio di Colori: HSI (Hue, saturation, Intensity)

Descrivere un colore attraverso le sue **componenti RGB** è abbastanza **innaturale per un essere umano**, per questo motivo si introduce lo spazio **HSI** (Hue, Saturation, Intensity) che permette di descrivere un colore attraverso i concetti più familiari di tinta (hue), saturazione (saturation) e luminosità (intensity).

Hue (tinta): tonalità , colore dominante così come viene percepito dall'osservatore;

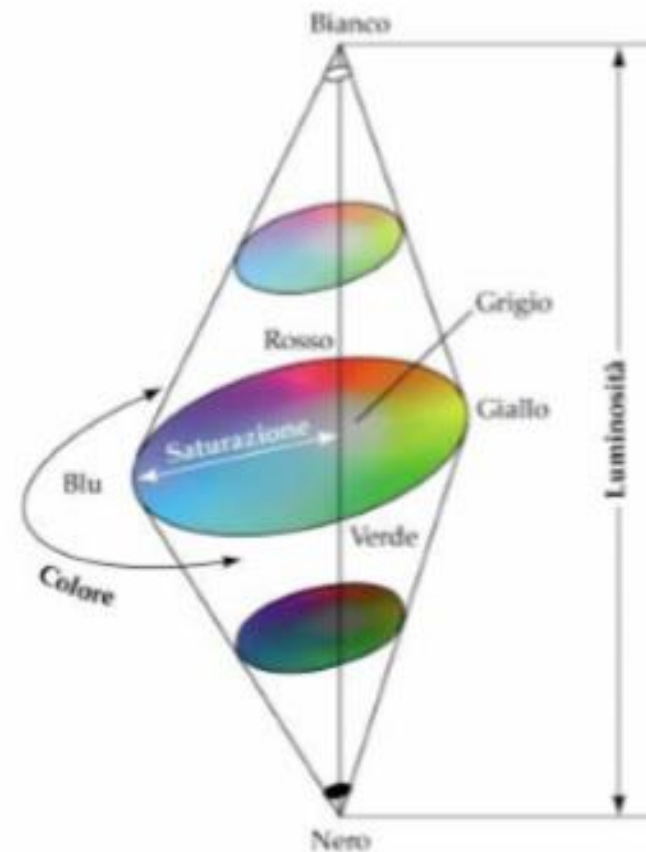
Saturation: purezza del colore o **quantità di luce bianca miscelata** con una certa tinta;

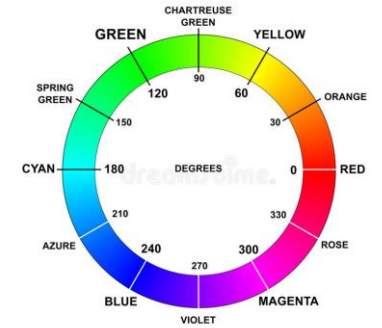
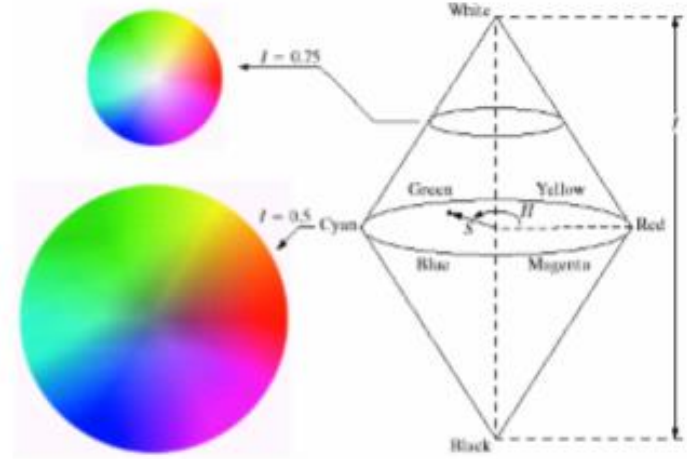
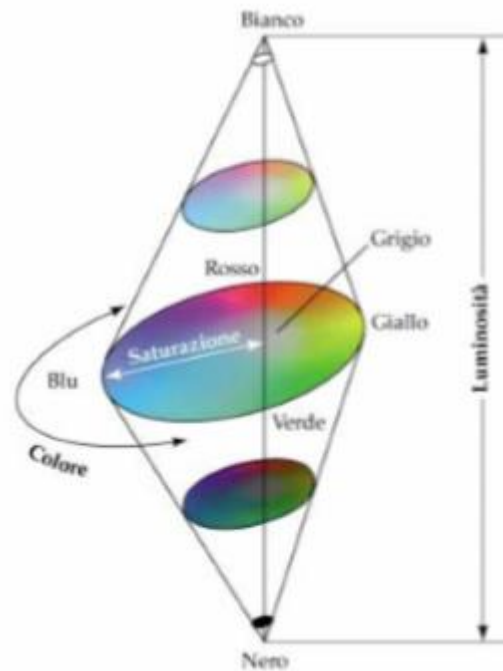
Intensity: luminanza, **luminosità o oscurità di un particolare colore**, varia tra 0 ed 1, 0 significa nero, 1 significa bianco.

Si definisce cromaticità:

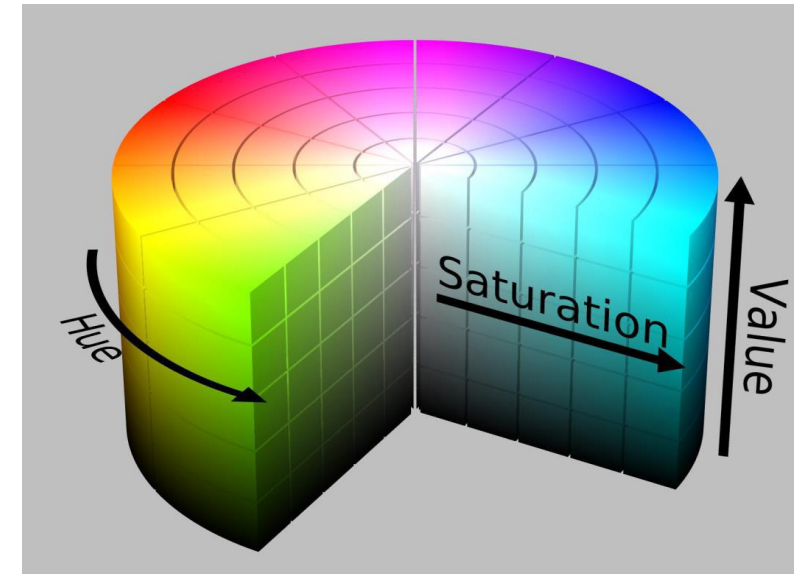
Hue + saturation = cromaticità .

- Si immagini di mettere il cubo RGB in equilibrio sul vertice $(0,0,0)$, con il vertice $(1,1,1)$ in alto e quindi mantenendo verticale la linea dei grigi.
- Il cubo viene ad assomigliare così ad un doppio cono, con la base a mezz'altezza e i due vertici in basso e in alto.





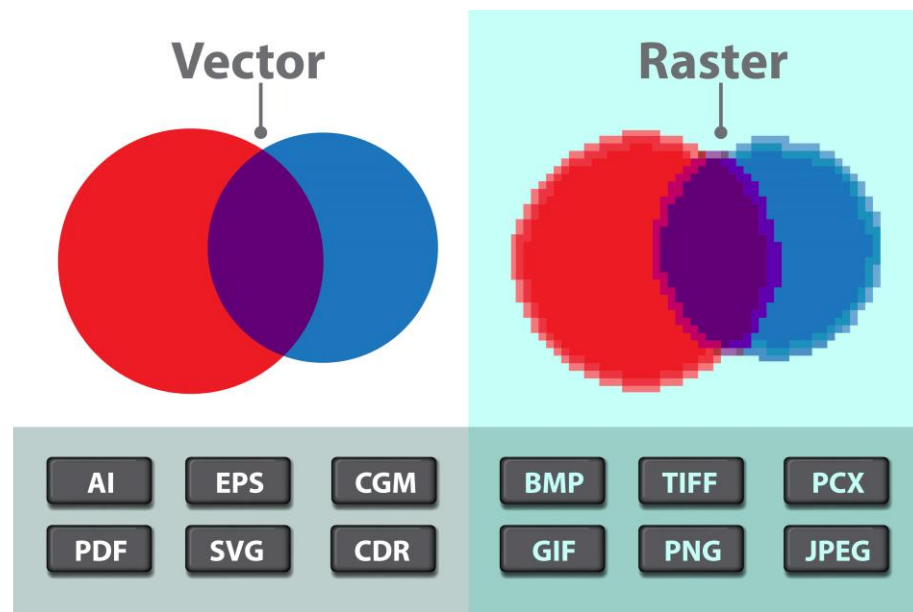
La tonalità **H** viene misurata da un angolo intorno all'asse verticale, con il rosso a 0 gradi, il verde a 120 e il blu a 240. La saturazione (**S**) (purezza del colore, quantità di luce miscelata) vale zero sull'asse del modello: una tinta non satura ha un colore che tende ad un grigio chiaro, una tinta satura ha un colore vivo e forte. L'altezza del modello rappresenta l'intensità (**I**) (luminosità o oscurità della tinta) con lo zero che rappresenta il nero e l'uno il bianco.





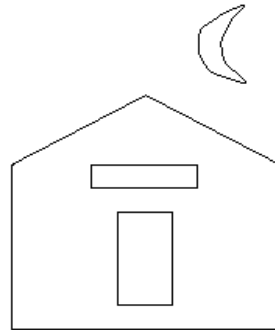
Immagini raster ed immagini vettoriali

- Nella grafica raster, l'immagine è una griglia rettangolare (raster) di pixel colorati. Nella memoria del computer vengono conservati i singoli pixel dell'immagine bitmapped. Un file bitmap contiene i valori dei pixel (.bmp, .gif)
- Nella grafica ad oggetti, detta **anche grafica vettoriale**, un'immagine consiste di oggetti grafici (punti, linee, rettangoli, curve, e così via) ognuno dei quali è definito, nella memoria del computer, da un'equazione matematica. Il file contenente un'immagine vettoriale contiene comandi e dati (.dxf, .svg)

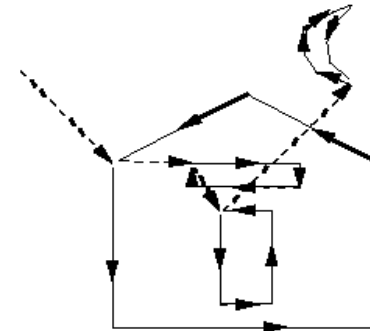


Scanner- stampanti sono dispositivi che fanno uso della rappresentazione **raster**,
il **plotter** è un dispositivo **vettoriale**

- Immagine vettoriale

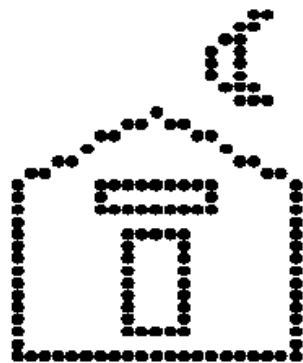


Ideal Drawing

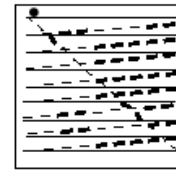


Vector Drawing

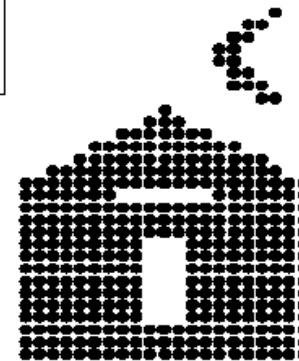
Immagine raster



outline primitives



Raster



filled primitives



-
- Poichè ogni oggetto nella grafica vettoriale è rappresentato (in memoria) da **un'equazione matematica**, per riprodurre l'immagine su un dispositivo raster questa **va trasformata in pixel**, operazione che si dice **rasterizzazione**.
 - Per le **immagini bitmap** ovviamente non esiste il concetto di rasterizzazione, in quanto sono già per definizione una **griglia di pixel colorati**.
-

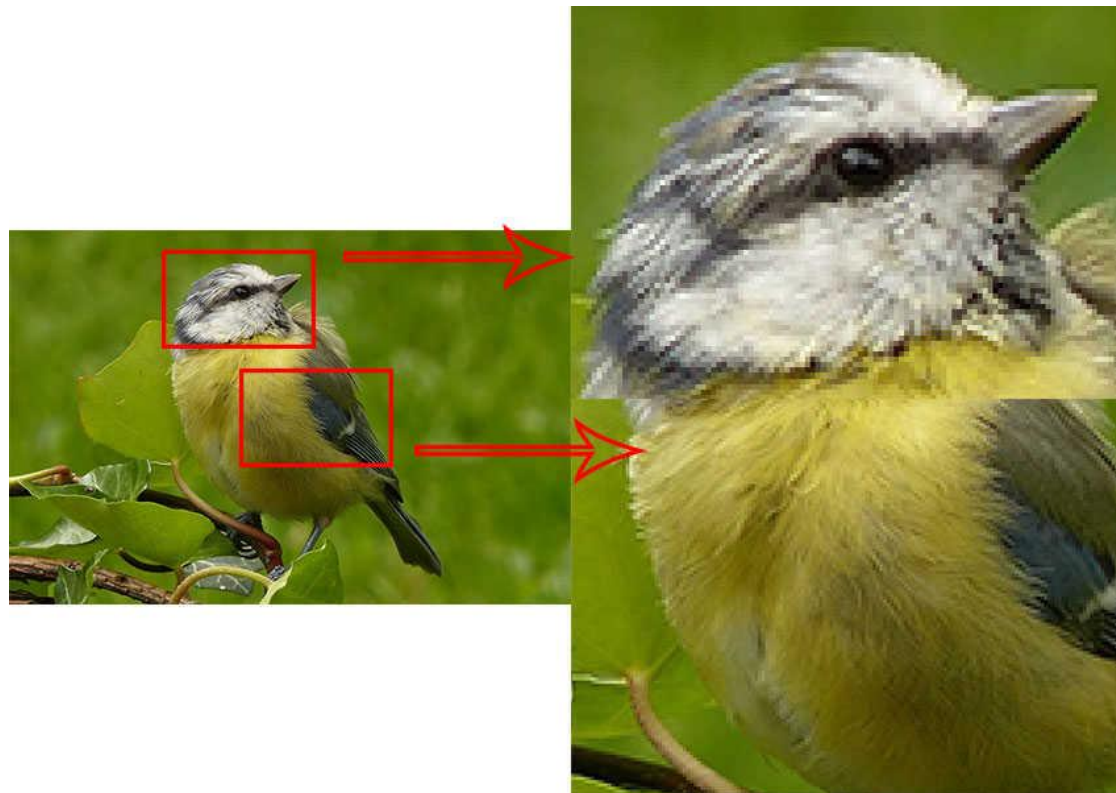
- La **grafica a punti (grafica bitmap o raster)** trova applicazione soprattutto nell'elaborazione di immagini fotografiche e nelle illustrazioni pittoriche. La loro natura a mosaico costituisce il punto di forza e anche di debolezza delle immagini bitmap.

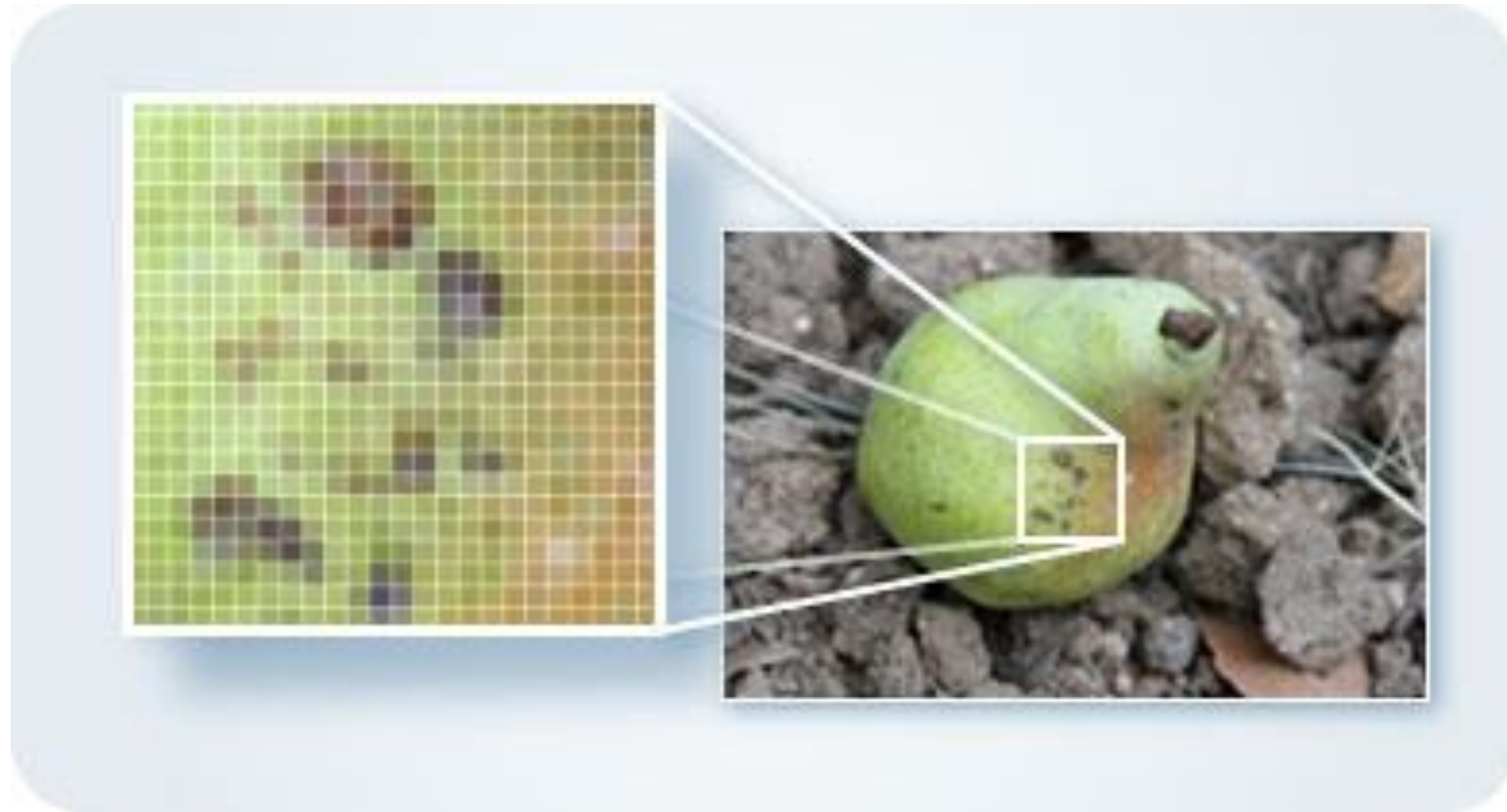




-
- **Vantaggi**
 - lavorando con i singoli pixel si possono ottenere effetti simili a quelli della pittura e grafica tradizionale (pennello, aerografo, matita, carboncino).
 - I programmi di fotoritocco, per esempio, funzionano con immagini raster, e i ritocchi sono possibili punto per punto.
-

- **Svantaggi**
- l'immagine si può ingrandire (su monitor o in stampa) solo ingrandendo la dimensione del pixel, che può diventare visibile, fino a creare effetti sgradevoli (pixelizzazione).







-
- Inoltre per **elaborare** (spostare, modificare, cancellare) una parte dell'immagine occorre letteralmente selezionare alcuni pixel e spostarli, indipendentemente da quello che rappresentano. La zona in cui erano rimane vuota, e i pixel vanno a sovrascrivere permanentemente quelli che si trovavano nella posizione di arrivo.
 - Nel **ruotare** un'immagine bitmap, i pixel vengono risistemati in modo che l'immagine appaia ruotata; ma, a meno che la rotazione non sia stata di 90° o multipli, la rotazione è solo un'approssimazione dell'immagine originale.
 - Nello stampare un'immagine bitmap, la stampante riproduce l'immagine punto per punto esattamente come i punti sono. Ciò indipendentemente dalla risoluzione della stampante.
-





GRAFICA VETTORIALE

- La grafica a oggetti è un'evoluzione della grafica a punti.
 - I programmi che consentono di creare immagini ad **oggetti non memorizzano il disegno come insieme di punti, ma con formule matematiche che descrivono i singoli oggetti;**
 - le formule matematiche sono scritte in qualche linguaggio (PostScript è il più noto). Ogni oggetto del disegno viene memorizzato in un database interno di oggetti grafici descritti matematicamente.
-



-
- Grazie a questa tecnica gli oggetti si possono ingrandire, rimpicciolire, ruotare, ridimensionare, colorare (bordi e contenuto) senza nessuna perdita di qualità.
 - Inoltre, gli oggetti si possono trattare in modo indipendente, come se ognuno fosse tracciato su un foglio trasparente.
 - Gli oggetti possono essere messi uno sull'altro, eventualmente quello sopra può nascondere quello sotto, che comunque non viene permanentemente cancellato.
-



**IMMAGINE
VETTORIALE**



**IMMAGINE
RASTER O BITMAP**



- In fase di stampa, **invece** di indicare alla stampante **dove vanno stampati i singoli pixel**, alla stampante **arriva la descrizione** (matematica, descritta in qualche linguaggio, per esempio **PostScript**) e la stampante (che può contenere un interprete PostScript) stampa l'immagine alla sua migliore risoluzione.
- Quindi, contrariamente alla grafica a punti, la qualità di una **illustrazione ad oggetti è device independent**: l'output uscirà alla miglior risoluzione della stampante.
- Generalmente poi la descrizione di un'immagine ad oggetti occupa meno spazio della descrizione di un'immagine a punti.



- I formati ad oggetti possono naturalmente contenere delle parti a punti, poichè il punto è anch'esso un oggetto grafico.
- In modalità bitmap la rappresentazione di un cerchio avverrà con pixel neri su sfondo bianco.
- In modalità vettoriale il cerchio verrà rappresentato con una formula, per esempio con le coordinate del centro e la lunghezza del raggio.



-
- Uno dei principali **svantaggi** consiste invece nella **natura intrinsecamente discreta della rappresentazione tramite pixel**: le primitive grafiche (quali linee, cerchi, poligoni), devono essere **convertite nelle matrici di pixel** che meglio le rappresentano, per poter essere visualizzate su un display di tipo raster.
 - Questa operazione è chiamata **scan conversion**.
-



• Scalable Vector Graphics (SVG)

- SVG è uno standard definito in un file testo XML per la rappresentazione di immagini vettoriali, non solo per l'uso destinato al web.
- E' un linguaggio per descrivere grafici 2D in XML.
- L'elemento html `<svg>` (introdotto in html5) è un container per grafici svg,
- `<svg width="100" height="100">`
- `<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" / >`
- `< /svg>`





Raster Scan Display System

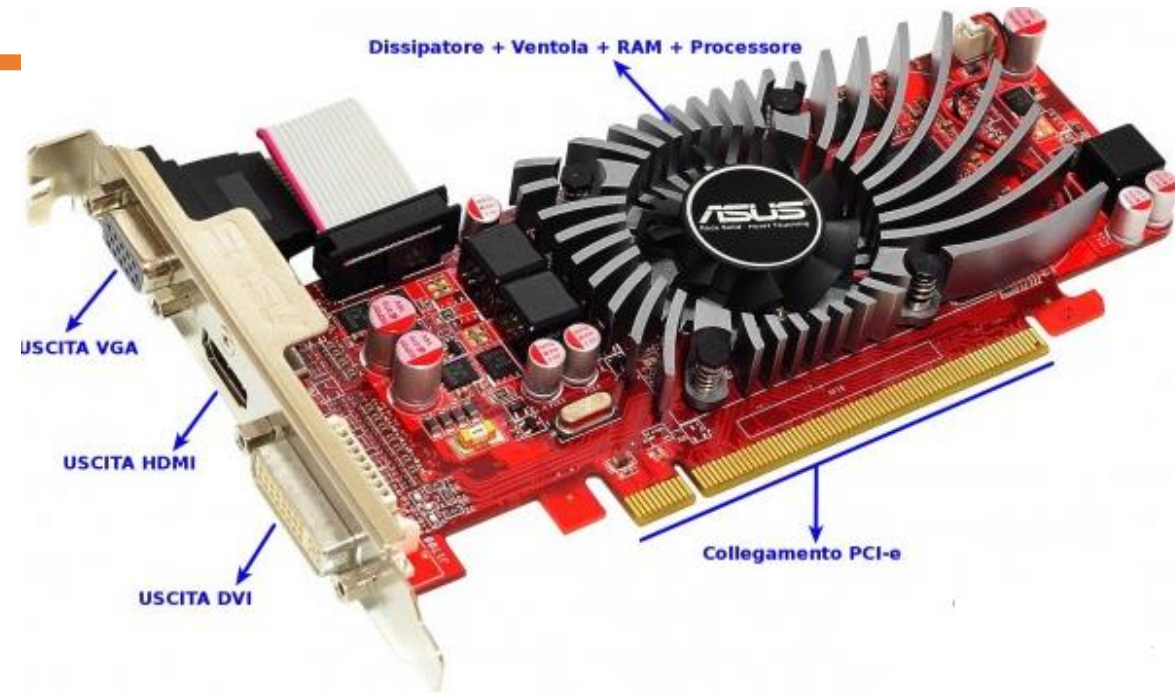
- I sistemi grafici a **display raster, Raster Scan Display System**, consistono essenzialmente di tre componenti:
 - **Scheda grafica** (detta anche video card, graphics card), al cui interno si trova **la GPU o PROCESSORE GRAFICO** e una **speciale memoria RAM detta Frame Buffer**
 - **MONITOR** (**display raster**) **connesso via cavo alla scheda grafica**
 - Un **device driver** mediante il quale il sistema operativo controlla la scheda video.
-



-
- Le immagini sul monitor sono costituite da pixel.
 - La scheda video è responsabile del rendering dell'immagine (dati binari dalla CPU) sullo schermo del computer: traduce i dati binari prodotti dalla CPU in immagini sullo schermo.
 - La CPU, lavorando insieme ad applicazioni software, invia informazioni sull'immagine alla scheda video.
 - La scheda video decide come utilizzare i pixel sullo schermo per creare l'immagine.
 - Quindi invia tali informazioni al monitor tramite un cavo.
 - • il computer deve eseguire questo processo circa sessanta volte al secondo.
-

Una scheda video è costituita da un circuito stampato su cui sono montati i componenti:

1. GPU,
2. Memoria video,
3. BIOS video,
4. Connessione monitor,
5. Connessione scheda madre per dati e alimentazione



'Unità di Elaborazione Grafica' – o GPU – è il “cuore” della scheda video moderna, quello che permette di **elaborare** le immagini eseguendo rapidamente gran parte dei complessi calcoli matematici e geometrici richiesti e **di coordinare** l'invio dell'immagine al monitor nel momento giusto.

In genere la GPU è praticamente invisibile in quanto, data l'elevata velocità di elaborazione e la conseguente attitudine a produrre molto calore, è “nascosta” sotto una grande ventola o un dissipatore.



LA RAM

componente in grado di memorizzare le informazioni elaborate dalla GPU ed eventualmente le immagini completate nell'attesa di inviarle al monitor. Ciascuna locazione di memoria della RAM si occupa di memorizzare i dati relativi a ciascun pixel, inclusi il colore e la sua posizione sullo schermo; si parla di RAM in modalità frame-buffer quando una parte di essa è dedicata a memorizzare le immagini complete finché queste non debbano essere visualizzate. La memoria RAM agisce a gran velocità, è dual-ported (si può leggere e scrivere su di essa contemporaneamente) ed è direttamente collegata al DAC (per le schede video dotate di uscita VGA).

IL BIOS

il BIOS (sistema di input/output di base) è un chip che memorizza le impostazioni della scheda video e, al momento del suo avvio, esegue tutti i test di diagnostica sulla memoria, sull'input e sull'output della scheda stessa.

Il DAC è una componente in grado di convertire il segnale digitale generato dalla GPU in un segnale analogico leggibile ed utilizzabile dai monitor VGA. (il DAC è utilizzato soltanto per la "sincronizzazione" della palette in caso di collegamento DVI/HDMI, poiché in tal caso il monitor è in grado di leggere il segnale digitale e non c'è bisogno di conversione).

Il DAC invia le immagini al monitor tramite un cavo di collegamento disponibile ad oggi in diversi formati.



Il **collegamento Scheda – Monitor** (le uscite video)

Sono diverse le tipologie di collegamento tra scheda video e monitor.

Le più utilizzate ad oggi sono:

VGA: si tratta di uno standard del 1987, compatibile con i monitor a tubo catodico (CRT) ma usato ancora oggi per compatibilità in una gran quantità di monitor LCD. Essendo uno standard abbastanza vecchio soffre di alcuni problemi strutturali, i più noti sono collegati alla risoluzione ed alla distorsione dell'immagine;

DVI: introdotto grazie ai monitor LCD, risolve i problemi collegati alla risoluzione e riscontrati nello standard VGA; può essere utilizzato da monitor LCD, proiettori e televisori al plasma;

HDMI: si tratta di uno standard del 2003 in grado di supportare risoluzioni ad alta definizione (e di conseguenza monitor HDMI) e che cerca ancora oggi di imporsi a tutti gli altri standard.

PCI-e (PCI-express): introdotto nel 2004 come un'evoluzione del PCI, ha soppiantato rapidamente AGP per la maggiore potenza erogata ed è diventato lo standard di fatto per il collegamento tra scheda video e scheda madre.



Modalità 2D-accelerata

Le schede grafiche 2D-accelerate sono quelle che siamo abituati a vedere da almeno una decina d'anni a questa parte, ovvero quelle **dotate di GPU in grado di “aiutare” il processore del computer ad eseguire parte delle funzioni** (tracciare linee, archi e forme semplici) e dei calcoli necessari per la visualizzazione delle immagini..

Modalità 3D-accelerata

Questo tipo di scheda dispone delle funzionalità delle accelerazione 2D e vi si affianca la modalità 3D, in cui **la GPU esegue ancora più calcoli, e l' unico compito della CPU resta fornire i dati geometrici tramite i quali la GPU stessa calcolerà i pixel fotogramma per fotogramma.**

Per questa modalità si richiedono una grossa quantità di RAM – che conterrà le texture da applicare ai pixel – ad alta frequenza e potenzialità di calcolo parallelo (con più “core” per ogni GPU): il processore grafico deve infatti essere in grado di calcolare rapidamente i dettagli della superficie (visibilità, coordinate, texture e relativi pixel) a partire dai dati geometrici forniti dalla CPU.



scheda grafica integrata e dedicata nei computer

La grafica integrata **integra l'hardware video** nella scheda madre, non utilizza la propria memoria. Condivide le risorse con la CPU.

La grafica dedicata utilizza un hardware video separato per eseguire la grafica.

La grafica integrata è adatta per attività semplici come l'elaborazione di testi, la navigazione e la modifica, mentre la grafica dedicata è adatta per applicazioni grafiche pesanti.

La grafica integrata consuma meno energia e aumenta la durata della batteria, mentre la grafica dedicata fornisce prestazioni più elevate.



Display

- Il display (monitor) utilizzato con il personal computer è un **dispositivo raster**: lo schermo cioè consiste di una matrice rettangolare di pixel (picture element). Ogni pixel del monitor può assumere un colore tra quelli disponibili. **L'intensità (colore) di ogni pixel viene letta dalla memoria frame buffer.**
 - Le principali tecnologie con le quali sono realizzati i monitor sono:
 - tubo a raggi catodici (CRT, cathode ray tube);
 - cristalli liquidi (LCD, liquid crystal display);
 - al plasma (PDP, plasma display peripheral).
-

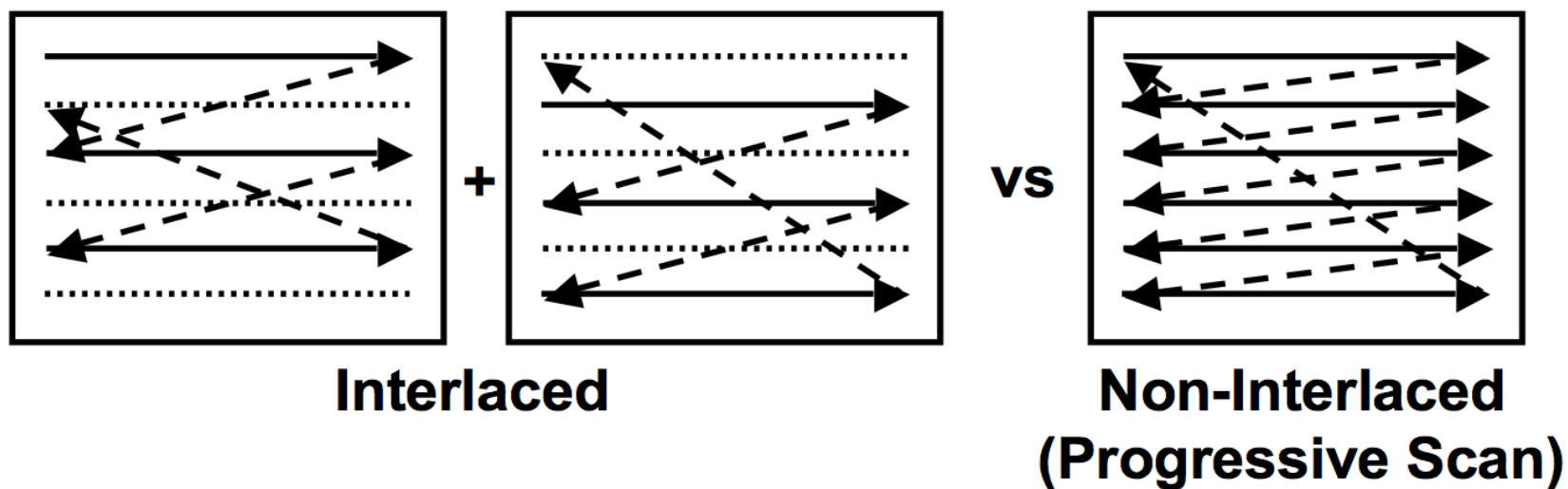


Refresh rate e frame rate

- **Refresh rate**, (misurato in hertz) numero di frame che vengono visualizzati sullo schermo al secondo.
 - **frame rate: (anche noto come frames per second (FPS))** misura quante immagini differenti al secondo sono visualizzate.
 - **Attenzione: il refresh rate include il disegno ripetuto di frame identici.**
 - Per esempio, un tipico valore di frame rate di un proiettore è 24 volte al secondo. Ma ogni frame è illuminato due o tre volte prima che il frame successivo venga proiettato. Quindi il proiettore gira a 24 frame al secondo, ma ha un refresh rate di 48 o 72 Hz.
-

I display LCD sono nativamente a **scansione progressiva**: avviene attraverso la scansione di tutti i frame immediatamente, una linea alla volta dall'alto verso il basso e da sinistra verso destra

Nella **scansione interlacciata** la scansione avviene sulla divisione di un frame in due semiquadri: un semiquadro contiene tutte le linee dispari dell'immagine, l'altro contiene tutte le linee pari. Il video interlacciato visualizza le linee di scansione pari e dispari come campi separati. Le **linee di scansione pari vengono disegnate sullo schermo**, **successivamente vengono disegnate le linee di scansione dispari** e si ottiene così un fotogramma video.



Poiché viene visualizzata solo mezza immagine alla volta, anche se a un intervallo di $1/60$ di secondo, le conseguenze dell'uso dell'interlacciamento sono prevedibili. Poiché si cattura solo metà dell'immagine alla volta, il movimento all'interno dell'inquadratura (dovuto al movimento della fotocamera o al movimento del soggetto) può causare artefatti da movimento, sfocature, **tearing** (strappatura dell'immagine) e generalmente fornisce una qualità dell'immagine ridotta rispetto alla scansione progressiva.



Tearing: un singolo fotogramma visualizzato sullo schermo contiene informazioni provenienti da due o più fotogrammi



Frame Buffer

- Il frame buffer è una porzione della **memoria RAM presente nella scheda** video, dove vengono memorizzate le immagini prima di essere visualizzate sul display.
 - Il frame buffer è una **collezione di diversi buffer** (piani di bit) dedicati.
 - Il principale tra questi buffer è il **color frame buffer**.
 - Altri buffer disponibili sono lo **Z-buffer**, anche chiamato depth buffer, che gestisce la visibilità.
 - Il **color frame buffer** contiene le componenti del colore per ogni pixel.
 - La sua forma a matrice riproduce la struttura a griglia del display raster; ogni elemento (x,y) contiene informazioni per il pixel corrispondente nella posizione (x,y) nel display
-



- Il **double buffer** (front buffer + back buffer) è comunemente usato per velocizzare le animazioni .
- Se l'immagine finale mostrata a video è contenuta nel front buffer, nel back buffer è contenuta la scena che sta per essere mostrata.
- I due buffer vengono poi scambiati dal processore grafico, dopo lo scambio i ruoli di back e front buffer si scambiano, e così succede dopo ogni frame visualizzato.



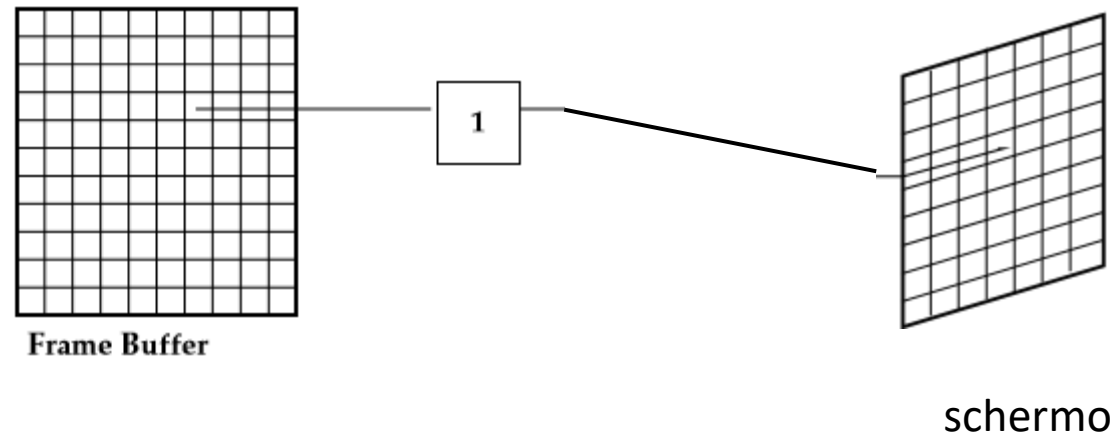
• Color Frame Buffer

- Nel color frame buffer **ogni posizione** di memoria indica il **colore di un pixel sul display**.
 - Poichè il **colore del pixel** di un monitor si forma in **sintesi additiva** a partire da tre primari **R, G e B**, ogni posizione sarà divisa in tre zone per i tre valori R, G e B.
 - Se per ogni colore primario sono riservati otto bit (1 byte),
 - ogni pixel richiede 24 bit (3 byte)
 - per esempio, un monitor di 1024x768 pixel richiede $1024 \times 768 \times 3$ byte = 768 x 3 Kbyte = 2304 Kbyte = 2.25 Mbyte di video RAM.
-



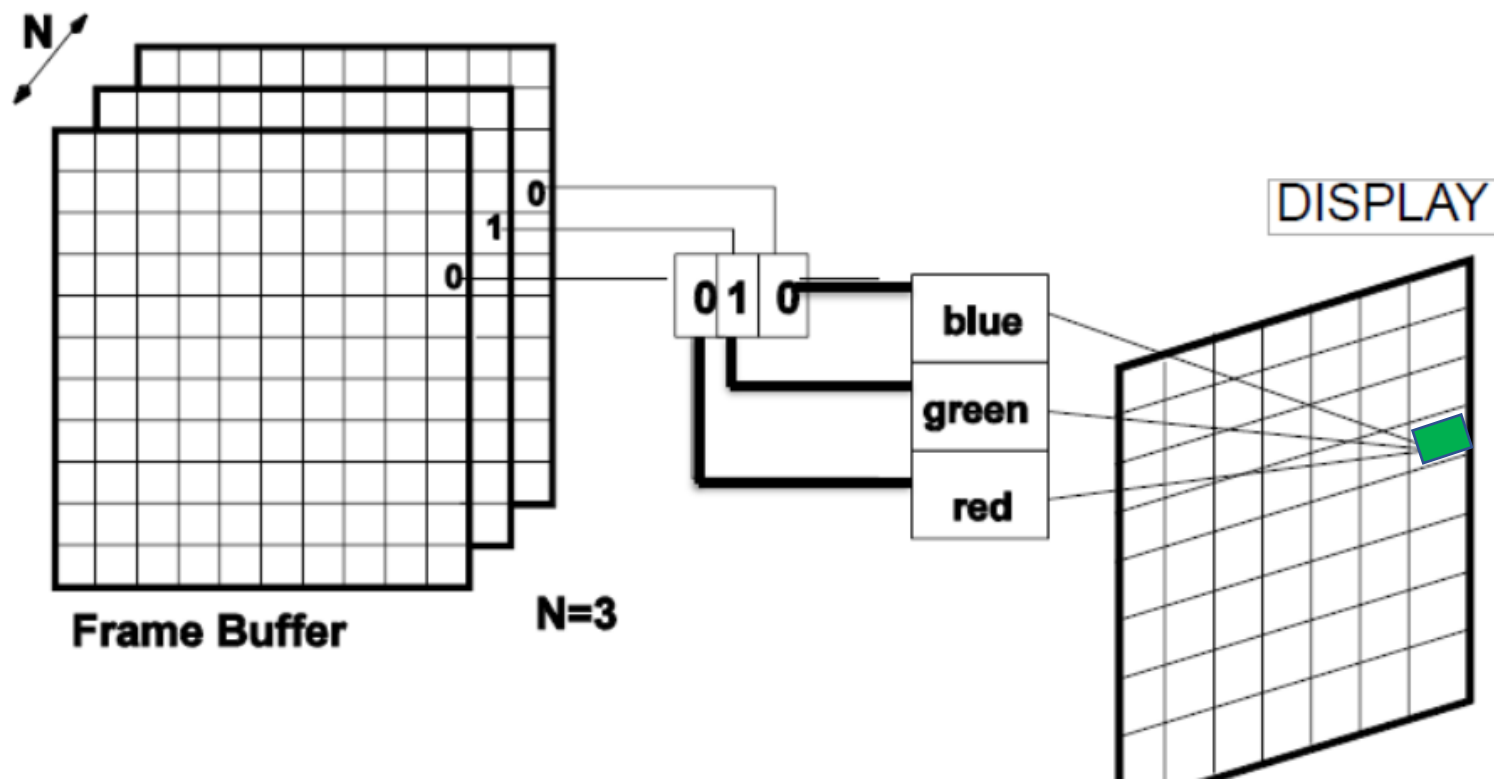
Metodi per codificare immagini a colori nel frame buffer:

- **Video monocromatico** il color buffer ha 1 bit per locazione, quindi basta 1 bit per pixel e quindi il color buffer ha un solo piano di bit





-
- Il color buffer per un monitor a colori può essere gestito in **modalità true color o pseudocolor.**
 - **Modalità True Color** I valori memorizzati nel color frame buffer rappresentano i valori effettivi delle componenti colore da visualizzare. Quindi un color buffer a N piani di bit è in grado di gestire 2^N valori di colore differenti.
-



Color buffer in modalità true color: Con tre piani di bit utilizzati ciascuno per una componente colore diversa.

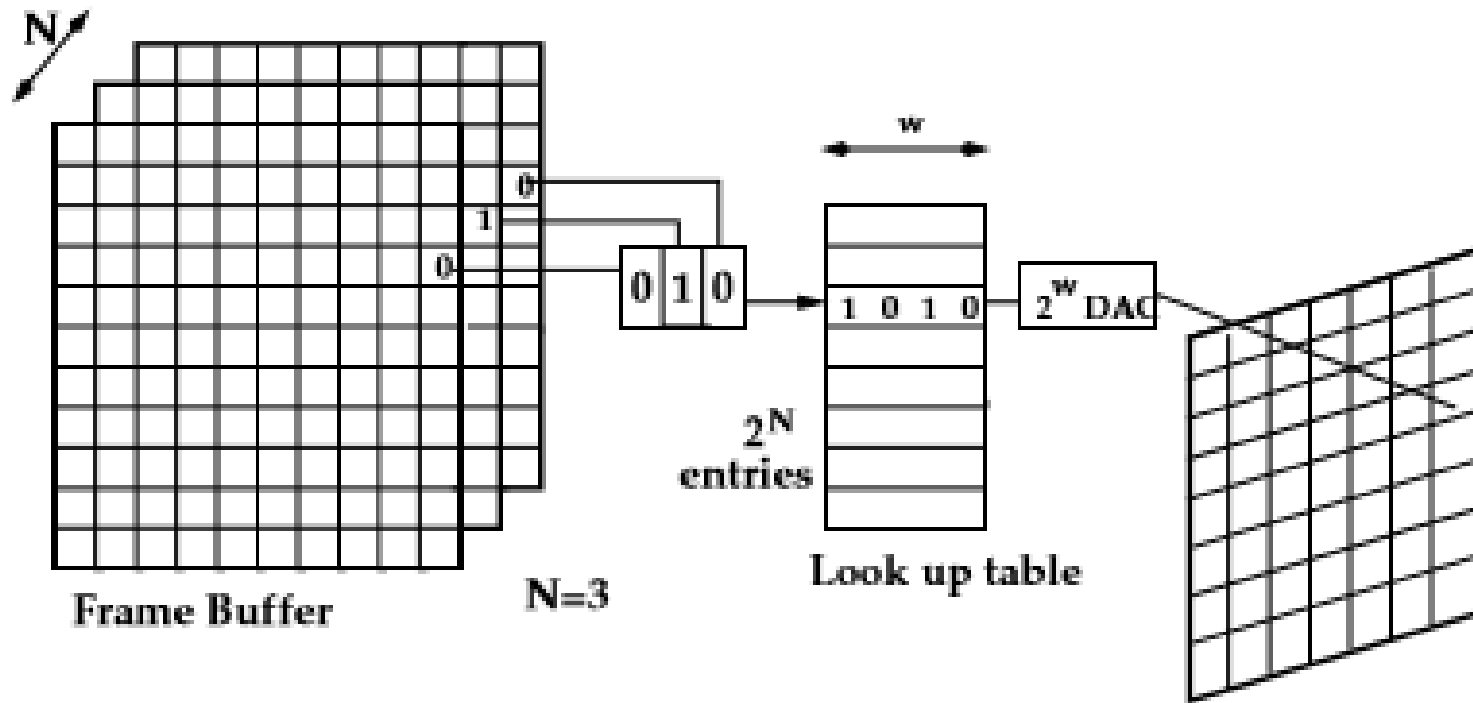


- Tipicamente si fa uso di **24 piani di bit** per le tre componenti colore (1 byte R, 1 byte G e 1 byte B) e **un quarto byte per il canale alfa** (che gestisce la trasparenza, **RGBA**).
- se un pixel ha il **canale alpha è 0**, il pixel dovrà essere considerato **trasparente** e quindi **non visualizzato**. Naturalmente avviene il contrario per i pixel con canale alpha uguale ad 1.
 $A = 0$ trasparente
 $A = 1$ visibile
- Quindi **32 bit** per pixel.
- Con 8 bit a disposizione, il valore di luminosità di una componente primaria può variare da 0 a 255.
- Quindi ci sono 256 possibili livelli di rosso, 256 possibili livelli di verde e 256 possibili livelli di blu, e quindi in totale $256 \times 256 \times 256 = 16777216$ possibili colori che un pixel può assumere.



Modalità pseudocolor

- I valori memorizzati nel frame buffer sono trattati come **indirizzi ad una tabella dei colori**, chiamata **look up table o colormap (LUT)** che contiene **2^N** elementi ciascuno dei quali composto da **w** bit
 - La colormap memorizza 2^w intensità di colore differenti, contemporaneamente, ma solo 2^N colori differenti possono essere disponibili su Frame Buffer.
 - Nella figura è mostrato un esempio di modalità pseudocolor che gestisce 8 colori contemporaneamente visualizzabili, scelti da una gamma di 16 colori disponibili (2^4).
-



Esempio: Frame Buffer con 3 bit-planes ($N=3$), $w=4$. (8 colori differenti da una gamma di 16 colori).

l'intensità a 3 bit per ogni elemento nel frame buffer è un indice che punta a una voce nella tabella dei colori.



-
- L'idea alla base della mappa dei colori è che invece di memorizzare un colore definito per ogni pixel in un'immagine, ad esempio in formato RGB a 24 bit, **il valore di ogni posizione nel frame buffer viene invece trattato come un numero di indice nella mappa dei colori**. Un vantaggio di questa modalità è **la possibilità di risparmiare spazio di archiviazione**: è possibile fare in modo che il numero di indice utilizzi meno bit del valore di output.
 - Ad esempio, è possibile utilizzare **un numero indice a 8 bit per cercare un valore di colore RGB a 24 bit** nella LUT. Poiché per ogni posizione nel frame buffer deve essere memorizzato solo il numero di indice a 8 bit, una tale rappresentazione occupa meno spazio di un'immagine a 24 bit completa della stessa dimensione. Ovviamente l'immagine può contenere solo 256 colori diversi (il numero di voci in una LUT a 8 bit), ma questo è sufficiente per molte applicazioni e di solito la degradazione dell'immagine osservabile è piccola
-

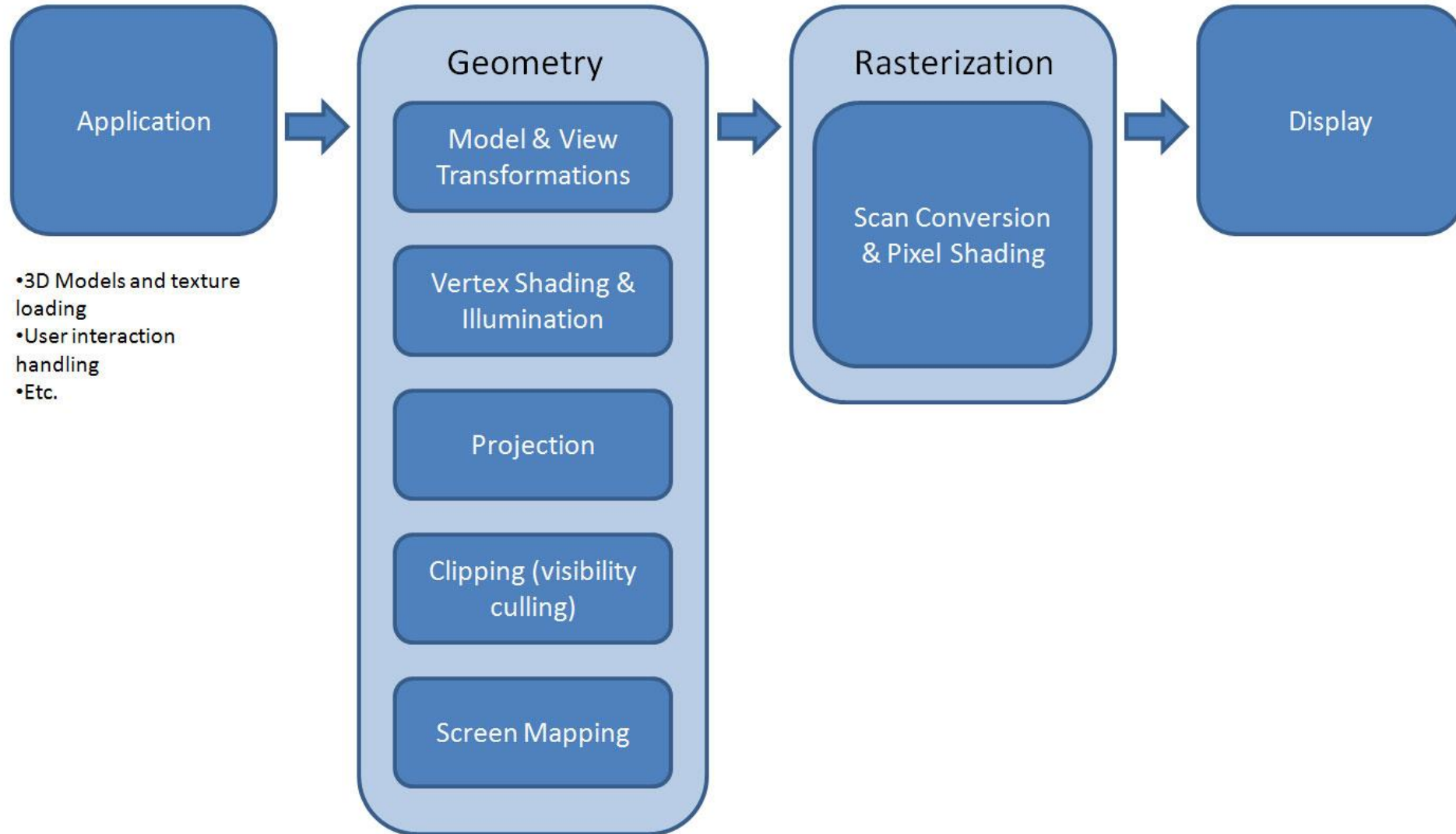


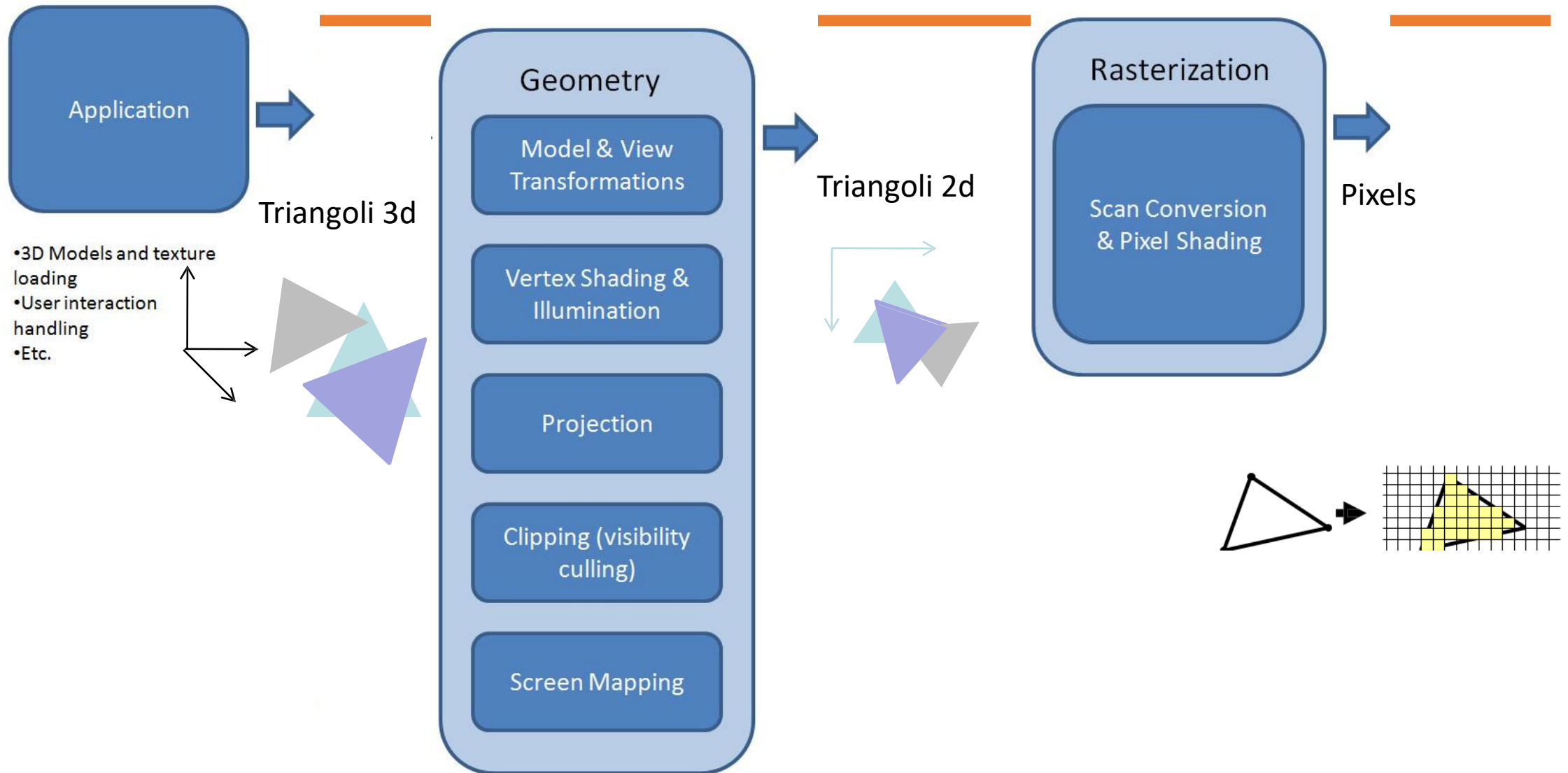
Rendering Graphics pipeline

- La GPU ha architettura a pipeline poichè progettata per soddisfare **la graphics rendering pipeline**.
 - **In input** alla pipeline vi è un insieme di primitive geometriche 3D, camera virtuale, risorse luminose, texture ed altro.
 - I vari stadi della pipeline possono essere globalmente accorpati in tre principali stadi: **application**, **geometry** e **rasterizer**.
-



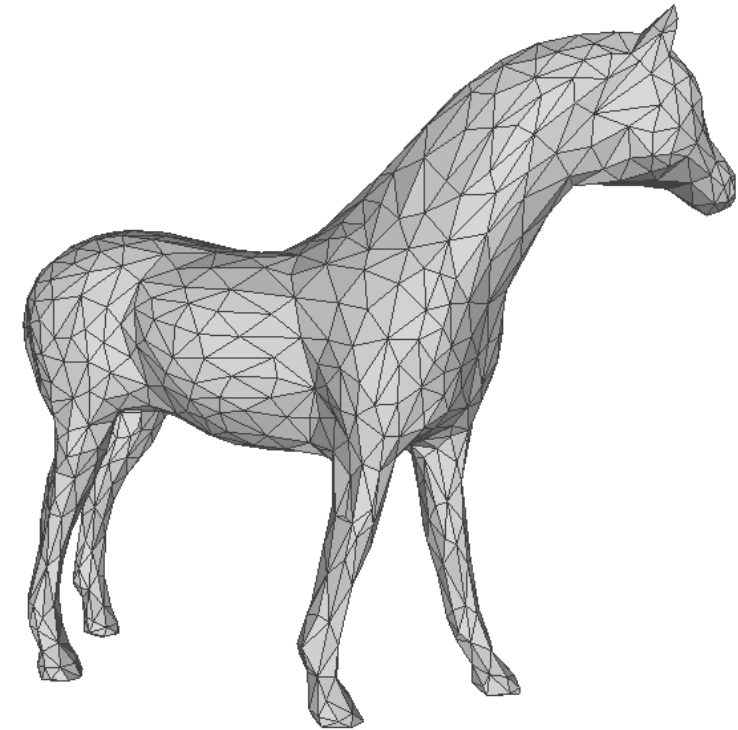
Real-Time Graphics Pipeline

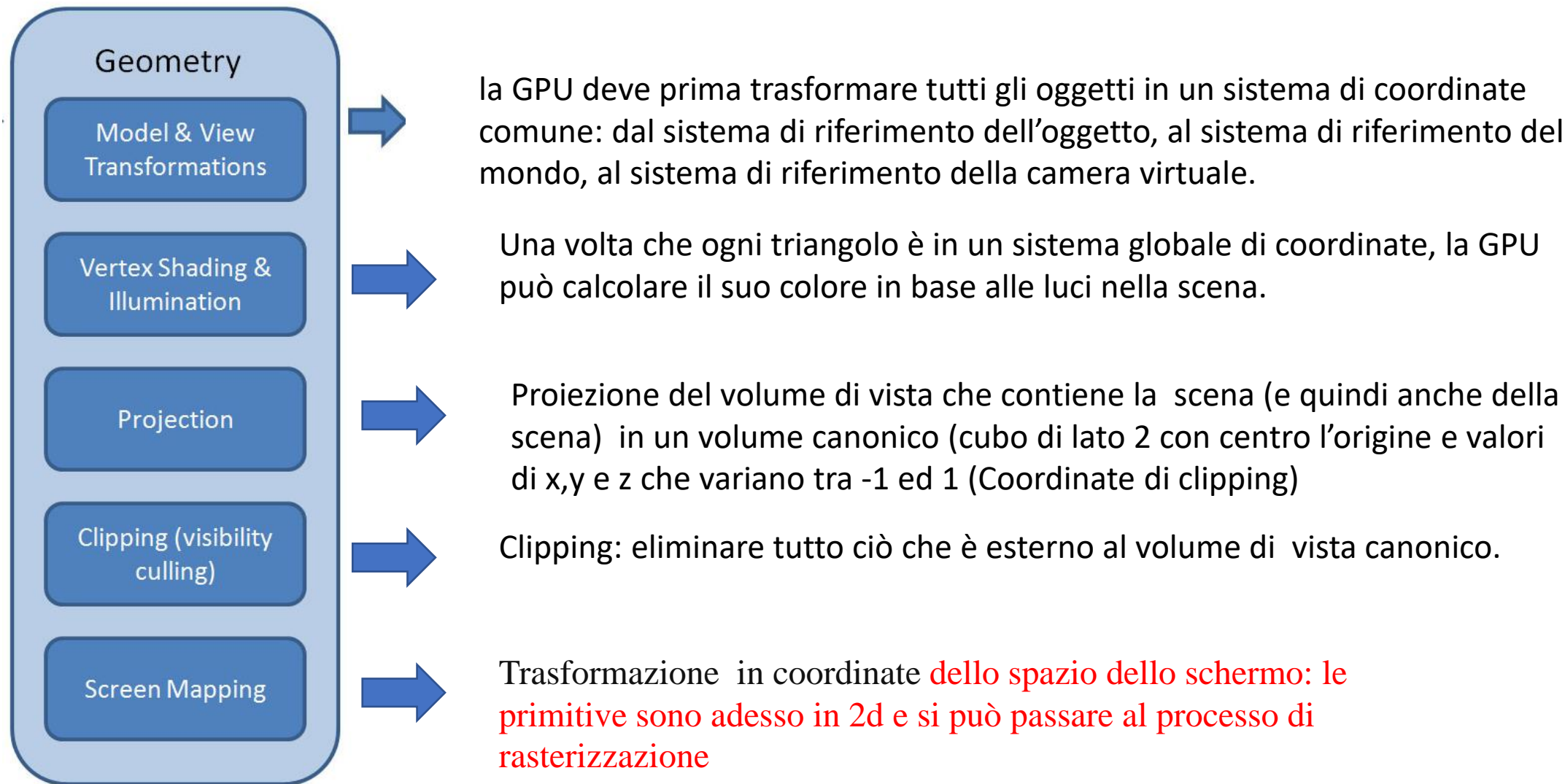




L'application stage gira su una, costruisce una geometria in base ad input forniti dall'utente e produce in output poligoni, o mesh di poligoni.

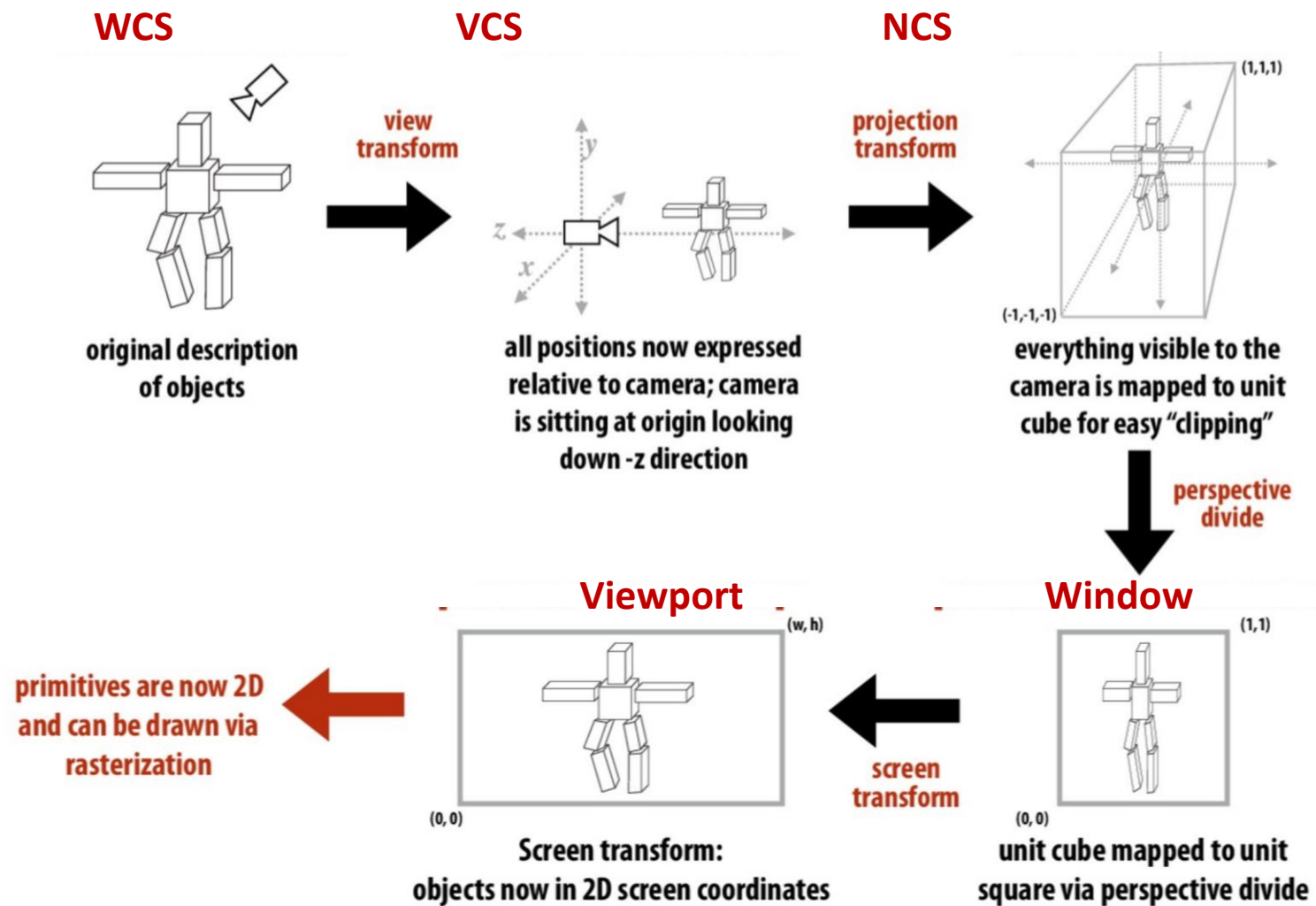
La maggior parte dei sistemi grafici in tempo reale presuppone che **tutto sia fatto di triangoli**, e prima suddividono forme più complesse, come quadrilateri o patches di superficie curva, in triangoli. Lo sviluppatore utilizza una libreria di grafica per computer (come OpenGL o Direct3D) per fornire ogni triangolo alla pipeline grafica un vertice alla volta



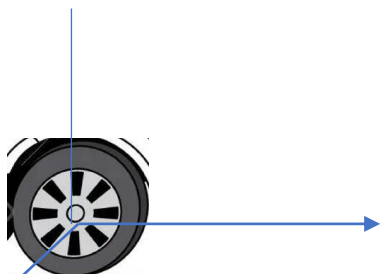
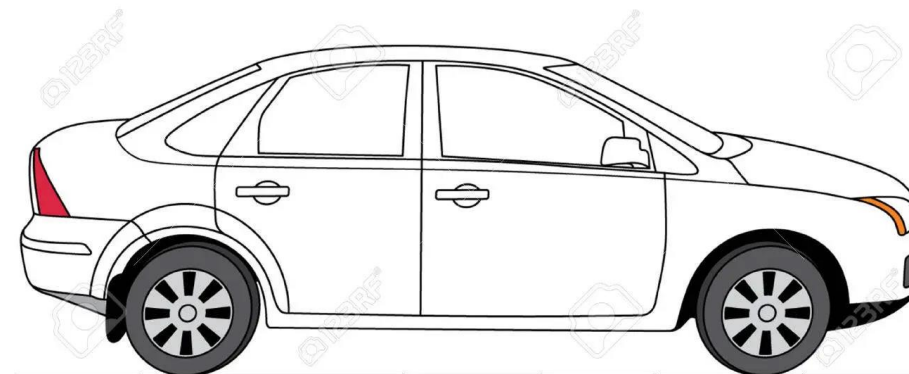
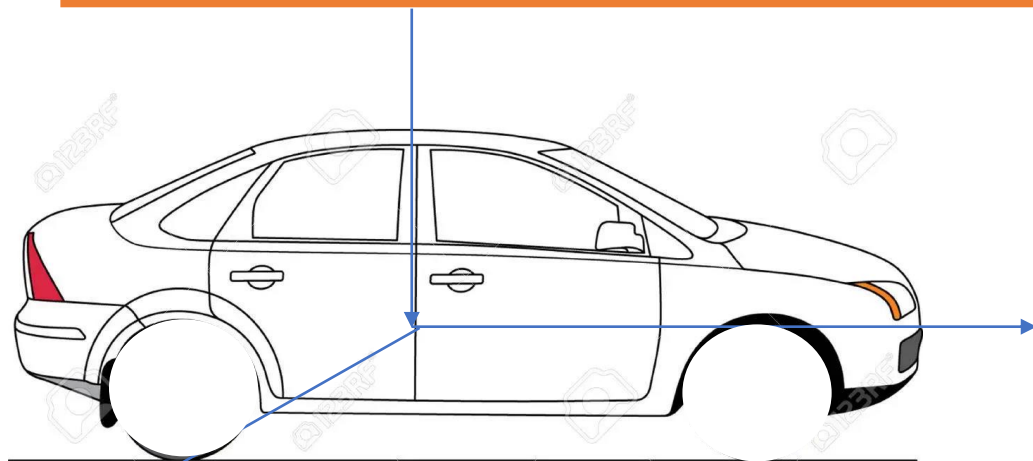




Geometry stage

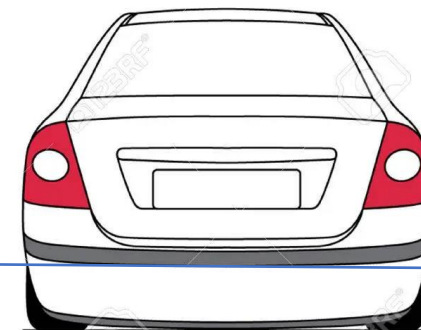
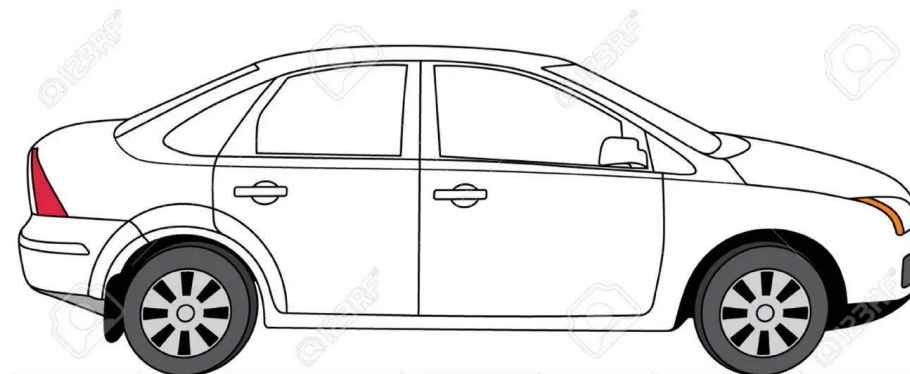
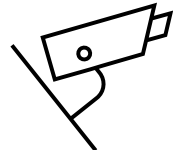


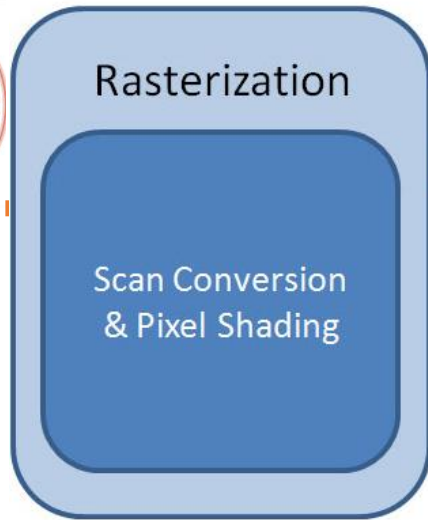
Sistema di riferimento
comune: sistema di
riferimento del mondo



Ogni oggetto della scena viene modellato in
un suo sistema di riferimento (Sistema di
riferimento dell'oggetto)

Sistema di riferimento della
telecamera

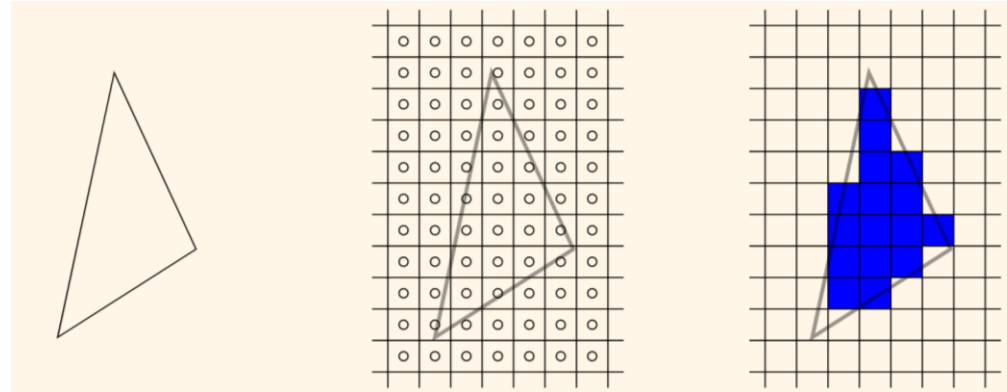




Il processo di **scan conversion** consiste nel determinare i pixel dello schermo interni ad ogni triangolo visibile sullo schermo.

Genera i «frammenti»

Primitive in
coordinate 2D



Frammenti

Fragment processing

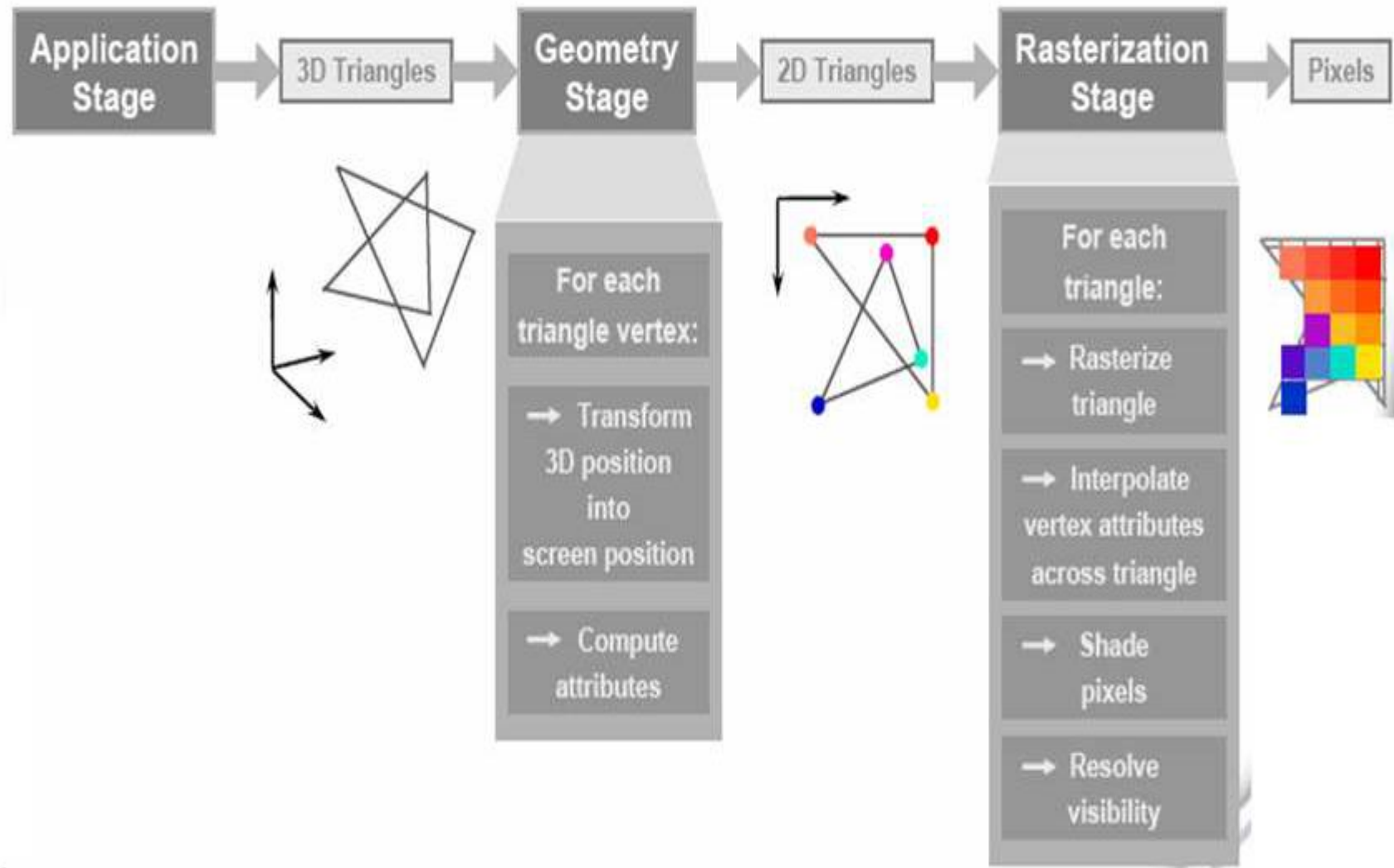
I frammenti hanno una posizione (pixel location) e altri attributi, come colore e coordinate di texture che vengono determinati interpolando i valori sui vertici.

Il colore effettivo di ogni pixel può essere preso direttamente dai calcoli di illuminazione, ma per un maggiore realismo, è possibile mappare delle texture sui triangoli.

Z-buffer e test di visibilità, Blending

Nella maggior parte delle scene, alcuni oggetti oscurano altri oggetti. Tutte le moderne GPU forniscono un buffer di profondità, una regione di memoria che memorizza la distanza da ogni pixel al visualizzatore.

Oppure hanno un livello di trasparenza ed il loro colore si mescola al colore degli oggetti che stanno dietro di essi (blending)





Evoluzione delle schede grafiche

- Nei primi general purpose computer (anni '60) la CPU era responsabile anche dell'intera fase di rendering compreso il refresh dello schermo.
 - Verso la fine degli anni 70 alle CPU vennero affiancati dei dispositivi special-purpose chiamati Display Processor Unit (DPU), che si occupavano del refresh del monitor ed erano collegati a CPU e video.
 - Nel 1987 IBM introdusse il VGA (Video Graphics Array), un nuovo controller hardware che oltre alle funzioni dei precedenti DPU , offriva alcune istruzioni per visualizzare primitive a video. Era il primo passo verso la prima generazione di GPU (Graphics Processor Unit).
 - Nel tempo gli ingegneri hardware hanno trasferito gli algoritmi per le trasformazioni delle primitive e gli algoritmi di rasterizzazione per produrre le immagini 2D su hardware grafico 3D specializzato ad alte prestazioni.
 - Questi multichip specializzati per la grafica 3D erano molto costosi e quindi integrati solo in costose workstation UNIX, come quelle prodotte nei primi anni 90 dalle compagnie Silicon Graphics International (SGI), e Evans Sutherland.
 - Con la diffusione dei personal computer, però, il peso economico del mercato dei videogiochi spingeva affinché l'interactive rendering arrivasse sui PC a prezzi contenuti.
 - Questa spinta si concretizzò nella seconda metà degli anni 90 con la configurazione di un unico chip grafico a basso prezzo, chiamato Graphics Processor Unit (GPU), facilmente integrabile nei comuni PC e nelle console dei video game.
 - La pipeline grafica, dopo anni di realizzazione software, si ritrovava ad essere in buona parte realizzata in hardware, allo scopo di velocizzare quanto più possibile la resa grafica.
 - In questa direzione si sono mosse la prima e la seconda generazione di GPU
-



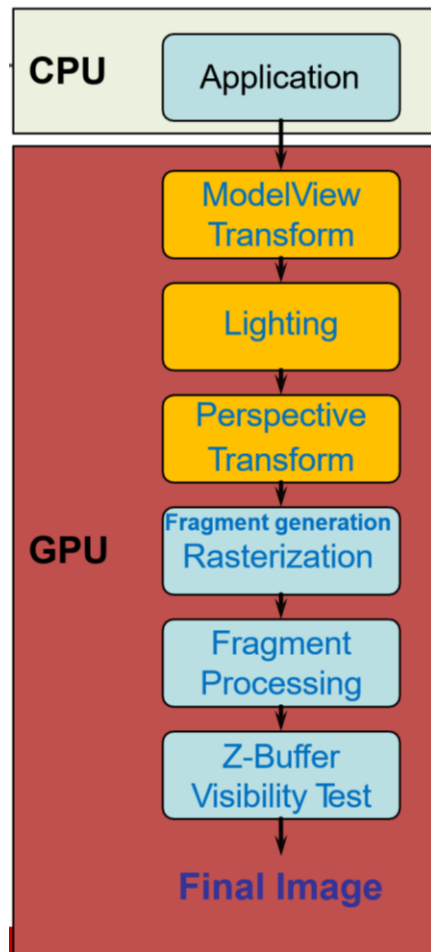
GEN.	PROG.	ANNI	CHIPSET	CARATTERISTICHE
I	NO	'90s	Voodoo di 3dfx TNT 2 nVidia Rage di ATI	single chip GPU. Completa gestione della fase di rasterizzazione da parte delle GPU, trasformazioni dei vertici ancora in CPU, set operazioni su pixel limitato.
II	NO	'99-'00	nVidia GeForce 256 GeForce2, Ati Radeon 7500, S3 Savage3D	La GPU ha completa gestione delle trasformazioni vertici 3D, estese operazioni matematiche per combinare texture, miglior configurabilità, ma non è programmabile.



- In una **fixed-function pipeline**, ovvero in una pipeline non programmabile, il programmatore interagisce con la pipeline via interfaccia software standard quale per esempio OpenGL o DirectX.
- Una **fixed-function pipeline della 3D API** (OpenGL o DirectX) implementa gli algoritmi di rendering direttamente nella scheda grafica, memorizza i dati dei vertici nella onboard video memory, evitando il passaggio dei dati attraverso il system bus.
- Con l'evoluzione dell'hardware grafico, si è arrivati alla progettazione realizzazione di **programmable pipeline**, in cui c'è la possibilità di poter programmare il comportamento di alcune sue fasi.



Fixed Function Pipeline



Implementazione cablata via hardware della pipeline grafica

PRO:

Miglioramento delle prestazioni

CONTRO:

il programmatore ha un controllo limitato su come l'hardware crea l'immagine finale. Per creare effetti non standard, come l'ombreggiatura dei cartoni animati, bisogna ricorrere a molti stratagemmi.



-
- I programmatori potevano accedere alle funzionalità fornite dall'hardware grafico della GPU **tramite opportune interfacce 3D di programmazione (API)**.
 - Le due maggiori **interfacce di programmazione grafica** sono **OpenGL** (sviluppata dalla Silicon Graphics) e **DirectX** (sviluppata dalla Microsoft).
-



-
- La realizzazione della pipeline in hardware, da un lato ha **permesso di migliorare notevolmente le performance delle applicazioni**, e le interfacce API hanno semplificato lo sviluppo di applicazioni, fornendo ai programmatori un livello di astrazione rispetto alle specifiche del sistema, e garantendo una maggiore portabilità del software.
 - **Da un altro lato, però, la standardizzazione degli algoritmi di rendering non permetteva flessibilità negli effetti di resa grafica, limitandone qualità e realismo.**
-



- Sebbene le **performance** guadagnate impiegando hardware grafico dedicato per eseguire vari compiti su vertici e pixel erano **eccellenti** a confronto della sola gestione della CPU, i programmatori 3D **persero** un notevole **controllo sull'immagine prodotta**.
- L'**off-line rendering**, a differenza del **real-time rendering** si avvaleva di standard general purpose CPU che costruivano frame by frame le animazioni impiegando giorni, settimane. **Il vantaggio di usare general purpose CPU era la flessibilità del programmatore di poter usare la CPU per ogni effetto che si poteva immaginare.**



-
- **L'off-line rendering system** peccava di **scarsa velocità**, ma riusciva ad ottenere **rendering di alta qualità e realismo**.
 - La flessibilità e generalità di offline rendering systems sono le caratteristiche chiave che erano state abbandonate nelle precedenti generazioni di hardware grafico 3D. **In altre parole quello che era stato perso era la programmabilità.**
-



Soluzione

- Per poter realizzare real-time rendering con effetti paragonabili a quelli ottenibili già qualche anno prima tramite tecniche di off-line rendering, **era necessario realizzare GPU in cui le diverse fasi della pipeline fossero programmabili**. In questa direzione si sono mosse le successive tre generazioni di GPU.

III	SI'	'01-'02	nVidia GeForce 3 Ge-Force 4, Microsoft Xbox, ATI Radeon 8500	Piena vertex programmability, E' possibile specificare le computazioni effettuate sul singolo vertice. Non si ha programmabilità pixel.
IV	SI'	'02-'03	nVidia GeForce FX, (architettura CineFX), ATI Radeon 9700/9800	Piena vertex e pixel programmability.
V	SI'	>2006	nVidia GeForce 8800, ATI RV770 Fermi INTEL Larrabee Maxwell	Miscrosoft lancia DirectX 10 e nello Shader Model 4.0 introduce il Geometry Shader con il quale è possibile gestire intere geometrie. GPGPU General-Purpose GPU: 32 bit floating point throughout the pipeline,



Cosa si è fatto

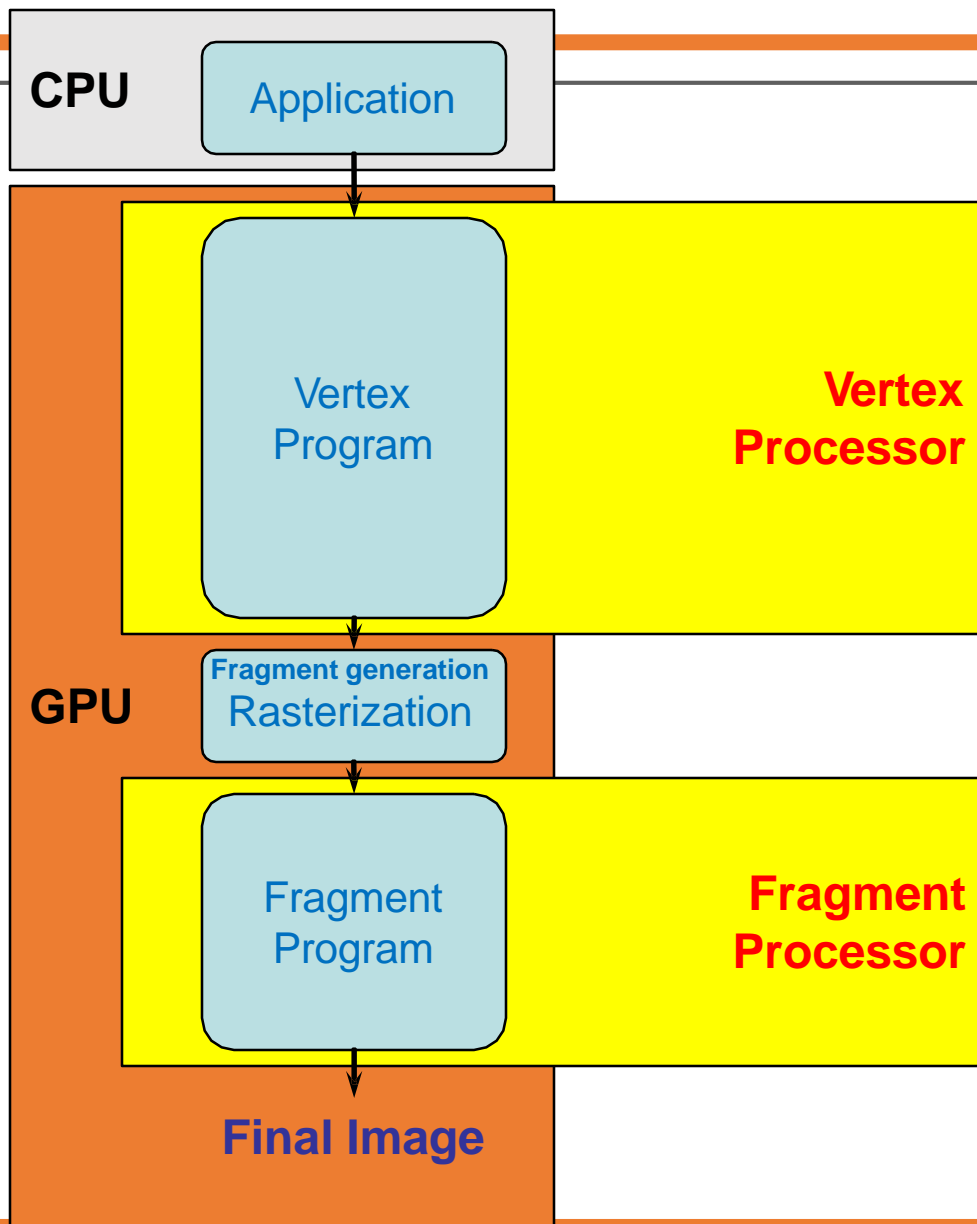
- Si sono cominciate a inserire nelle **GPU** delle unità computazionali, dette **Shader Units**, che **possono essere riprogrammate**, dando così al programmatore la possibilità di **modificare nel dettaglio l'algoritmo di resa impiegato**.
-



-
- **Rendering pipeline programmabile**: significa libertà da parte del programmatore di implementare vari tipi di algoritmi come differenti modelli di luminosità, o generazione di coordinate texture, trasformazioni non lineari.
 - Si hanno due distinte unità programmabili: **vertex processor** e **fragment processor**. Queste rappresentano rispettivamente la manipolazione della geometria e la rasterizzazione dei pixel.
-



Programmable Pipeline



Il programmatore
spedisce semplicemente
i dati alla scheda video e
scrive un programma
per interpretare i dati e
per creare l'immagine



Vertex Shader

- Un Vertex Shader è un programma che prende in ingresso una serie di informazioni su un vertice della scena, e da in uscita la posizione di questo nel sistema di coordinate di vista, e, opzionalmente, il colore, il vettore normale alla superficie e le coordinate texture.
 - La Shader Unit quindi può modificare proceduralmente i dati in uscita variando le matrici di trasformazione, gli attributi relativi alle fonti di luce che illuminano la scena, o altri parametri necessari a specificare le operazioni di trasformazione e illuminazione del Geometry Stage.
-



-
- I Vertex Shader aprono la strada per una vasta gamma di effetti, che prima dovevano essere elaborati nel livello applicativo della pipeline grafica,
 - **come deformazioni della geometria (oggetti semi-rigidi spostati dal vento, movimento dei fluidi, oggetti flessibili) o distorsioni che simulano effetti ottici come la rifrazione della luce attraverso diversi materiali.**
 - Inoltre è possibile implementare un modello di illuminazione diverso da quello di default del Geometry Stage, andando a calcolare opportunamente il colore del vertice in funzione degli attributi della superficie e della sorgente di luce.
-



-
- **Fragment Shaders**
 - Prendono come input gli output del programma vertex shader e le texture map. Producono un colore finale e una trasparenza come output. Sono spesso chiamati shader di frammenti.
-



Linguaggi per gli shader programs

- Linguaggi di programmazione
 - HLSL (Microsoft in DirectX)
 - GLSL (SGI in OpenGL (>2.0))

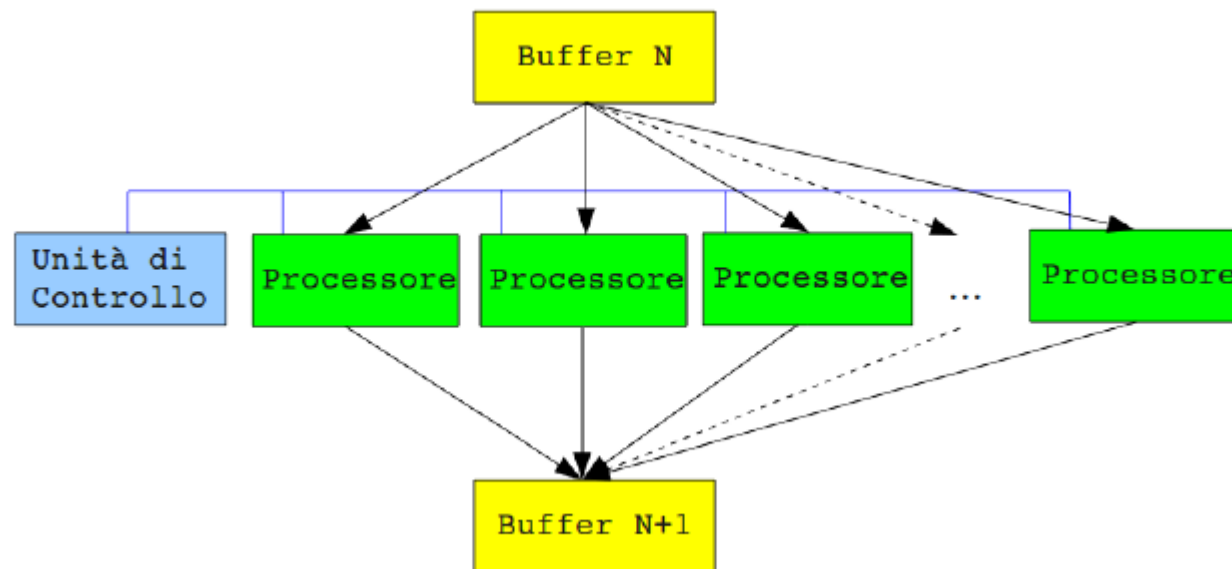


-
- In questo modo **dalla propria applicazione i programmatori di grafica possono accedere e controllare direttamente le funzionalità della scheda grafica.**
 - Le CPU sono di loro natura general purpose, mentre le GPU sono progettate specificatamente per la grafica.
 - Quindi, le **GPU risultano molto più veloci nell'elaborazione di graphic task rispetto alle CPU.**
 - **Le nuove GPU elaborano infatti decine di milioni di vertici per secondo e rasterizzano centinaia di milioni di frammenti per secondo.**
 - Nelle GPU i dati vengono memorizzati nel Vertex Buffer, nella Texture Memory o nel Frame Buffer.
-



- Per ogni fase di elaborazione della pipeline, **le GPU contengono molte (diverse centinaia) unità di elaborazione identiche**, che hanno una **propria area di memoria temporanea e propri registri interni**, ma **che condividono l'unità di controllo**. Queste unità sono dette **(Shader) Processing Unit**.
- A ciascuna Processing Unit è assegnato un diverso elemento presente nel buffer di origine e, quando un numero sufficiente di Processing Unit è pronta, tutti i dati sono elaborati contemporaneamente.
- Ovviamente, siccome l'unità di controllo è unica, unico è **il programma (flusso di controllo) che le Processing Unit parallele possono eseguire**.

- Questa architettura è molto efficiente, perchè l'esecuzione di un numero molto elevato di elaborazioni tutte uguali è molto comune nella pipeline grafica.



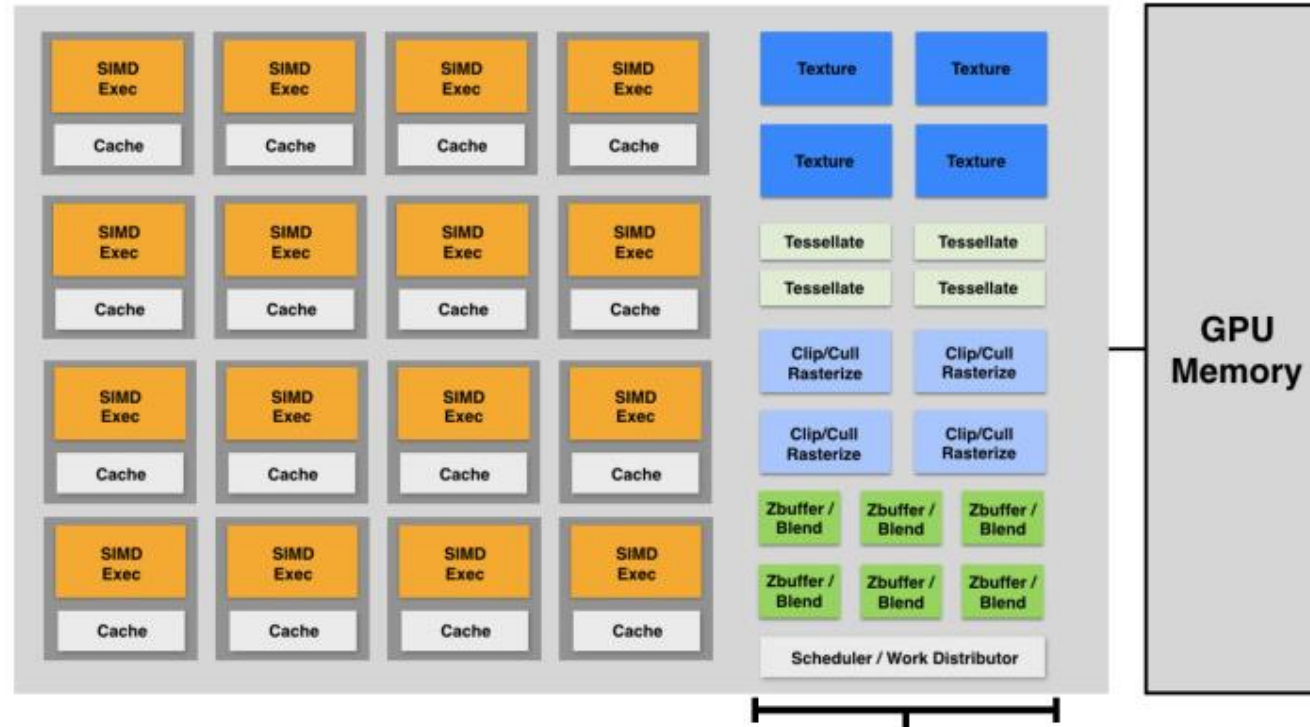
Le shader Processing Unit condividono un'unica unità di controllo



-
- Un'architettura di questo tipo è un'implementazione del paradigma **SIMD (Single Instruction Multiple Data)**. In particolare si parla della GPU come di uno stream processor:
 - dato uno stream di dati in ingresso (le primitive e le loro proprietà), essa applica la stessa operazione (sequenza di operazioni) a tutti i dati.
-



GPU- Multicore processor



Modern GPUs offer ~2-4 Tera-FLOPs of performance for executing vertex and fragment shader programs

Tera-Op's of fixed-function compute capability over here



- L'ultimo passo nell'evoluzione dalla fixed-function pipeline, al rendering programmabile è l'introduzione di shaders unificati.
- **Nelle ultime schede grafiche della quinta generazione invece di utilizzare processori separati specifici per gestire esclusivamente vertex shaders, geometry shaders, e pixel shaders, si è introdotta un'architettura "unified shader" che fornisce una grande griglia di processori paralleli floating-point sufficientemente generali da gestire i carichi computazionali di tutti gli shader.**
- Queste nuove configurazioni hardware hanno proiettato le GPU oltre al loro ruolo nativo di elaborazione di graphic task trasformandole in vere e proprie entità "parallele" alla CPU, utilizzate come **general purpose high performance computing, GPGPU**



-
- **GPGPU (General Purpose GPU)**
 - L'evoluzione tecnologica dell'hardware grafico è giunto oggi alla realizzazione di **dispositivi multithread, multicore, altamente parallelizzati, performanti e largamente programmabili**, progettati in modo tale da offrire la massima flessibilità possibile, a tal punto che **il loro utilizzo si è svincolato anche dal contesto applicativo per cui inizialmente le GPU erano state create: la Computer Graphics.**
 - Da qualche anno si è iniziato a parlare **di GPGPU (General- Purpose GPU)**, termine che in generale indica tutte quelle tecnologie utilizzate per accedere alla potenza di calcolo delle moderne periferiche grafiche, e utilizzarla per effettuare pesanti operazioni su dati matematici/scientifici.
-



- Queste tecnologie hanno proiettato le moderne GPU oltre al loro ruolo nativo di supporto grafico, trasformandole in vere e proprie entità "parallele" alla CPU , tanto da renderle loro concorrenti in diversi settori.
- OpenCL, nVidia CUDA, ATI's CTM
- Queste soluzioni nascono per gestire in modo più o meno trasparente il parallelismo a livello di dati, ma non offrono alcuno strumento che guidi la realizzazione di sistemi grafici basati sull'esecuzione parallela di task, distribuiti tra CPU e GPU.



- L'adozione di una soluzione GPGPU , in generale, offre un grande vantaggio in termini di tempo di calcolo in qualsiasi ambito applicativo che richiede l'analisi e/o la trasformazione di grandi quantità di dati;
- si citano tra i vari settori ad esempio, la previsione della struttura delle proteine, **previsioni meteo**, **analisi delle immagini medicali**, astrofisica, reti neurali, ricerche oceanografiche, fisica delle particelle, chimica quantistica, astronomia, acustica, simulazioni elettromagnetiche, elaborazione di video in alta definizione, crittografia, rendering audio posizionale e in decine di altri settori.



-
- L'estrema flessibilità delle GPU moderne, accompagnata dalla GPGPU , permette oggi una totale libertà nel personalizzare qualsiasi fase della pipeline grafica "classica", riportando potenzialmente l'intero processo grafico ad essere realizzato via software.
-



Cross-Stitching (Punto a croce)







Ocean

