

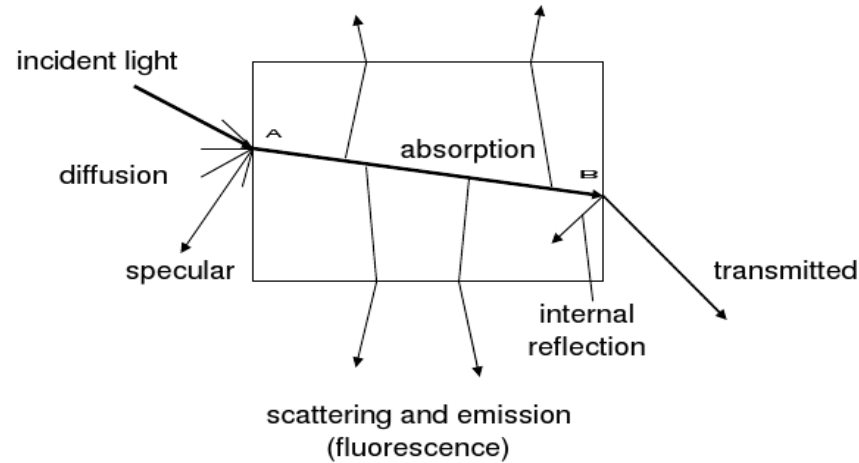
Grafica computazionale

Lezione 2

(slide parzialmente basate su
Computer Graphics - MIT OpenCourseware
Grafica Computazionale - Massimiliano Corsini – Università di Siena)

Interazione luce-materia

- Cosa succede quando un raggio di luce colpisce una superficie?



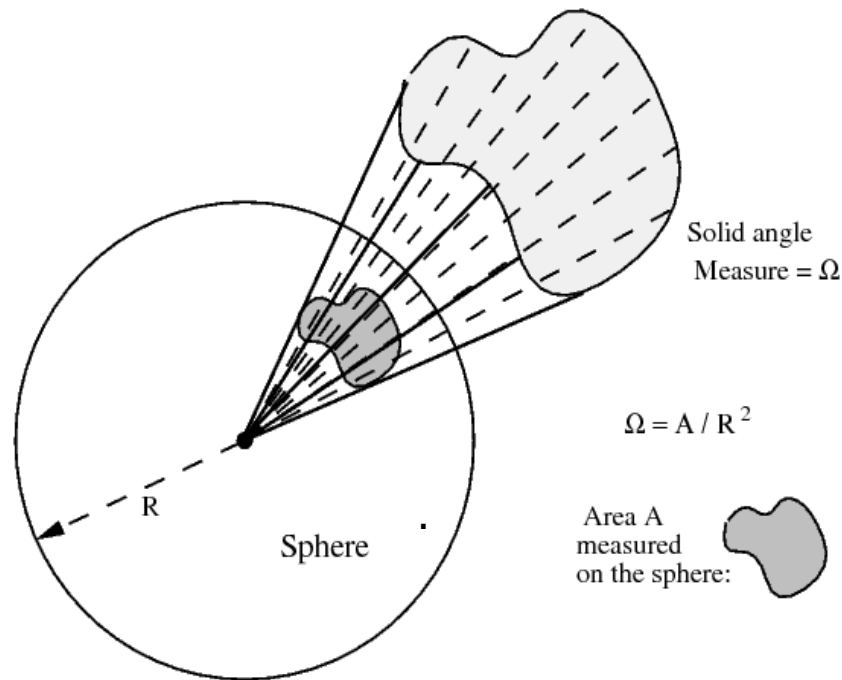
- L'energia luminosa, emessa da una qualsiasi sorgente luminosa (lampada, luce solare), viaggia ed interagisce attraverso la scena fino a stabilizzarsi.
- A noi tutto appare sempre stabile perchè questo processo avviene alla velocità della luce.

Ottica

- **Ottica Geometrica (Ray Optics):** la luce è modellata come raggi che si muovono attraverso la scena seguendo precise regole geometriche. Permette di modellare molti effetti come la riflessione e la rifrazione.
- **Ottica Ondulatoria (Wave Optics):** vede la luce come onda che si propaga, spiega fenomeni come la rifrazione e la diffrazione.
- **Ottica Elettromagnetica (Electromagnetic Optics):** permette di descrivere fenomeni come la polarizzazione e la dispersione non spiegabili dalla wave optics.
- **Ottica Quantistica (Photon Optics):** è l'applicazione della meccanica quantistica alla descrizione dei fenomeni luminosi.

Angolo Solido

- **Angolo Solido** può essere visto come l'estensione allo spazio tridimensionale del concetto di angolo piano.
- L'angolo piano θ si misura (in radianti) come il rapporto s / r dove s è la lunghezza dell'arco di cerchio di raggio r sotteso da θ .
- Analogamente l'angolo solido si misura (in steradiani) come il rapporto A/r^2 ove A è l'area della superficie sferica di raggio r sottesa dall'angolo .



Radiometria

- **Radiometria** è la scienza che misura la radiazione elettromagnetica nello spettro di frequenza ultravioletto, visibile e infrarosso.
- **Flusso radiante** (Watt) è l'energia radiante che attraversa una superficie nell'unità di tempo.

$$\Phi = \frac{dQ}{dt}$$

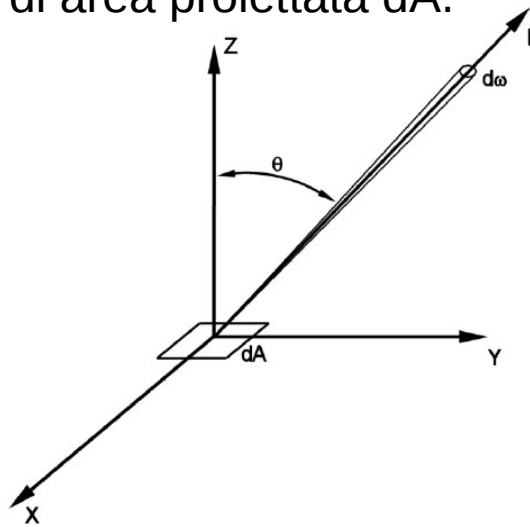
- **Irradianza (E)** (Watt/m²) è il flusso radiante che incide su un elemento di superficie.

$$E = \frac{d\Phi}{dA}$$

- **Radianza (L)** (Watt/m²sr) è il flusso radiante per unità di angolo solido per unità di area

$$L = \frac{d^2\Phi}{\cos\theta d\omega dA}$$

- Questo è il concetto fondamentale in radiometria: è la quantità fisica equivalente al concetto della luminosità osservata dagli umani.
- Misura il flusso elettromagnetico che si muove nel range infinitesimale di direzioni $d\omega$ e che passa attraverso un elemento di area proiettata dA .



Radiometria

- L'irradianza (E) è data dall'integrale della radianza incidente (L) lungo tutte le direzioni.

$$E(x) = \int_{\Omega} L(x, \omega) (\vec{n} \cdot \vec{\omega}) d\vec{\omega}$$

- Il flusso radiante è dato dall'integrale della radianza incidente (L) lungo tutte le direzioni e l'area considerata.

$$\Phi = \int_A \int_{\Omega} L(x, \omega) (\vec{n} \cdot \vec{\omega}) d\vec{\omega} dx$$

- Esempio: Una sorgente di luce puntiforme che emette luce uniformemente in ogni direzione produce (sorgente puntiforme)

$$E = \frac{\phi \cos \theta}{4\pi r^2} \quad L = \frac{\phi}{4\pi r^2}$$

- L'intensità luminosa diminuisce con il quadrato della distanza!

Riflessione e Riflettanza

- Riflessione è il processo con il quale il flusso elettromagnetico incidente ad una superficie lascia la superficie (senza cambio di frequenza)
- Riflettanza è la frazione del flusso incidente che è riflessa
- Ignoriamo:
 - Assorbimento (e Riemissione), Trasmissione, Fluorescenza
 - Diffrazione

Bidirectional Scattering Surface Reflectance Distribution Function

- Quando la luce colpisce una superficie interagisce con essa fino a lasciare la superficie da una differente posizione (scattering).
- La BSSRDF (Bidirectional Scattering Surface Reflectance Distribution Function) è una funzione che descrive il processo di scattering.
- La BSSRDF può essere scritta:

$$S(x_i, \vec{\omega}_i, x_r, \vec{\omega}_r) = \frac{dL_r(x_r, \vec{\omega}_r)}{d\Phi_i(x_i, \vec{\omega}_i)}$$

- Dove L_r è la radianza uscente dal punto x_r nella direzione $\vec{\omega}_r$ e Φ_i è il flusso radiante incidente nel punto x_i proveniente dalla direzione $\vec{\omega}_i$.
- È una funzione di otto parametri!!

Bidirectional Reflectance Distribution Function

- La BRDF (Bidirectional Reflectance Distribution Function) è un'approssimazione della BSSRDF per descrivere matematicamente la riflessione della luce sulla superficie
 - Niente rifrazione e/o riflessione interna => posizione incidente ed uscente coincidono
- In particolare, la BRDF è il rapporto tra la radianza uscente (riflessa) in una certa direzione in un dato punto della superficie (x) e l'irradianza incidente nello stesso punto in una data direzione.

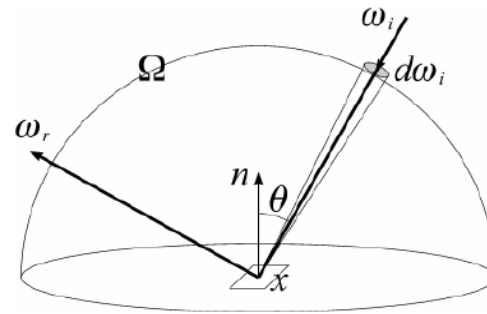
$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{dL_r(x, \vec{\omega}_r)}{dE_i(x, \vec{\omega}_i)} = \frac{dL_r(x, \vec{\omega}_r)}{L_i(x, \vec{\omega}_i)(\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i}$$

L'Equazione di Rendering

- La luce visibile in un punto della scena per una particolare direzione è data dalla luce riflessa più la luce emessa in quella direzione

$$\begin{aligned} L_o(x, \vec{\omega}_r) &= L_e(x, \vec{\omega}_r) + L_r(x, \vec{\omega}_r) \\ &= L_e(x, \vec{\omega}_r) + \int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i \end{aligned}$$

- La luce riflessa è un integrale che somma tutti i contributi luminosi pesati secondo l'angolo di riflessione e la BRDF (che dipende dal materiale)



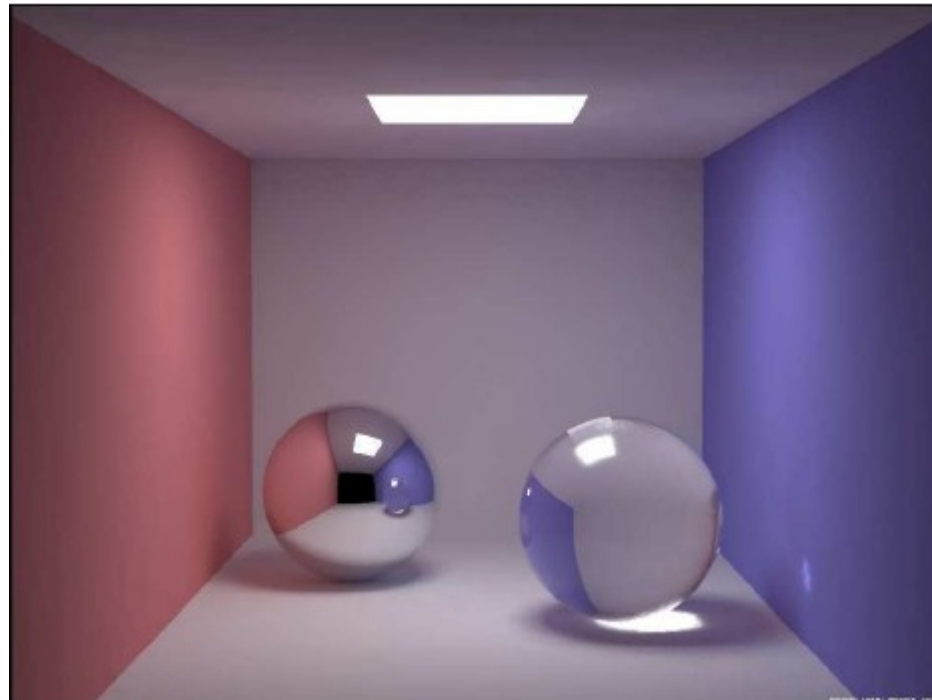
- Calcolo esatto dell'equazione della radianza => operazione complessa e molto costosa
- Sistema di grafica interattiva necessita di una formula utilizzabile per tutti i punti della scena più volte al secondo
 - Semplificazione dell'equazione (approssimazione)
 - Si modella gli effetti della luce soltanto localmente

Modelli di illuminazione

- Modello di illuminazione: formulazione matematica dell'equazione del trasporto dell'energia luminosa
 - Generalmente i modelli di illuminazione sono approssimazioni (locali) della Rendering Equation
- Lighting: calcolo del bilancio luminoso
- Shading: calcolo del colore di ogni pixel dell'immagine

Effetti Globali

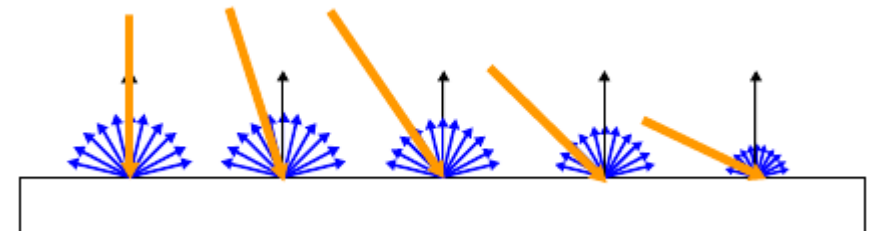
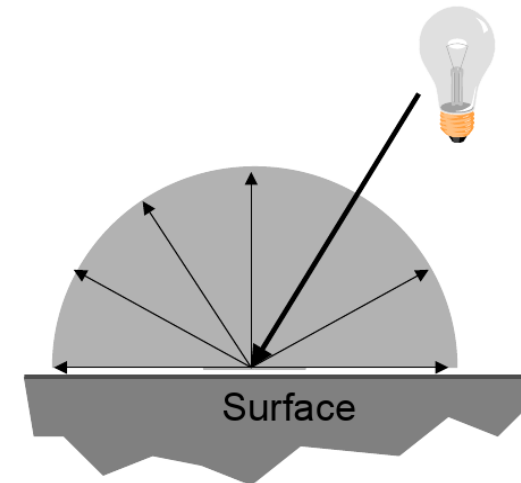
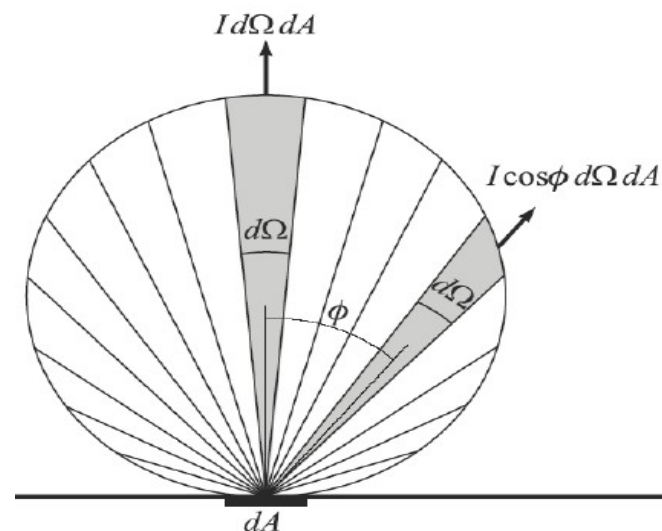
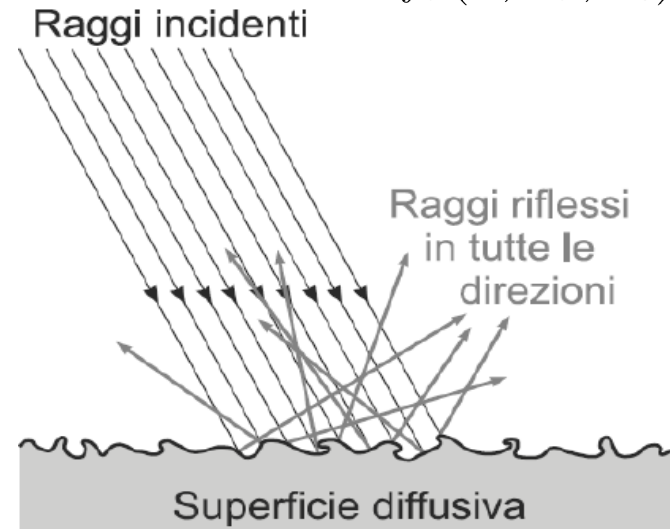
- Alcuni effetti sono dovuti da interazioni multiple tra materiali e non possono essere resi da modelli locali
 - Riflessioni
 - Ombre
 - Color bleeding
 - Caustics



Legge di Lambert (oggetti opachi)

- Materiali molto opachi (es. gesso e legno) hanno una superficie che, a livello microscopico, ha piccole sfaccettature che riflettono la luce in una direzione casuale
- Integrando su scala macroscopica: la luce si riflette uniformemente verso tutte le direzioni, con intensità proporzionale al rapporto tra la direzione del raggio incidente e la normale alla superficie in quel punto

$$f_r(x, \vec{\omega}_r, \vec{\omega}_i) = \rho(\vec{n} \cdot \vec{\omega}_i)$$



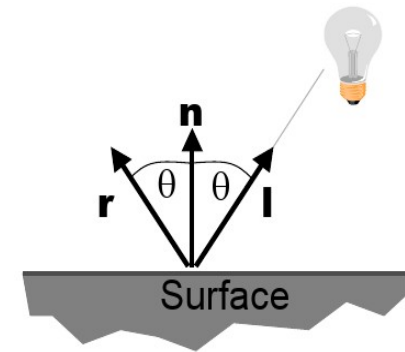
L'intensità della luce riflessa dipende (solo) dal coseno dell'angolo di incidenza $I = I_p k_d (\vec{N} \cdot \vec{L})$

- Per frequenze alte (lunghezze d'onda basse) ogni superficie è scabrosa!

Legge di Fresnel (specchi)

- Quando un raggio di luce passa da un mezzo ad un altro con diverso indice di rifrazione raggiunta la superficie di separazione parte del raggio viene riflessa e parte trasmessa
- La somma delle energie dei due raggi è uguale all'energia del raggio originale
- Se da aria a corpo solido non c'è rifrazione si ha solo riflessione L'angolo di incidenza è uguale all'angolo di riflessione

$$f_r(x, \vec{\omega}_r, \vec{\omega}_i) = \begin{cases} 1 & \text{se } \vec{h} = \vec{n} \\ 0 & \text{altrimenti.} \end{cases} \quad \vec{h} = \frac{\vec{\omega}_r + \vec{\omega}_i}{\|\vec{\omega}_r + \vec{\omega}_i\|}$$



- Vale per materiali molto lisci e lucidi

Modello di Illuminazione di Phong

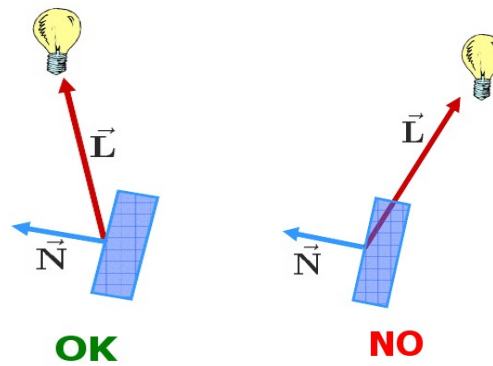
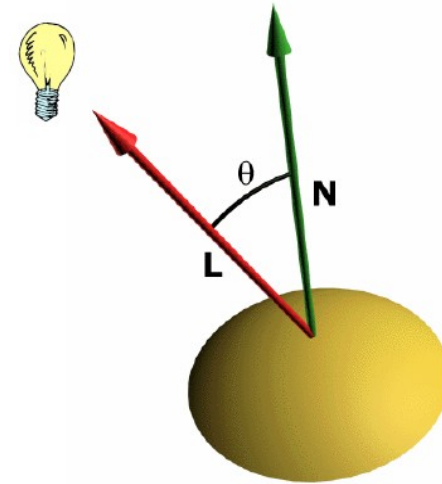
- Modello dovuto a Phong Bui-Tuon, prima metà degli anni '70
- Semplifica lo schema fisico di interazione luce-materia:
 - Solo sorgenti puntiformi
 - No inter-riflessioni
 - Calcolo locale dell'equazione di illuminazione
 - Approssimazione con due costanti della funzione di riflessione
 - Componente diffusiva (Lambert)
 - Componente riflessiva (Fresnel)

Il modello di Phong

- Simula il comportamento di materiali opachi
- Non modella la rifrazione: no materiali trasparenti o semi-trasparenti
- Abbastanza realistico anche se produce immagini un po' "plasticose"

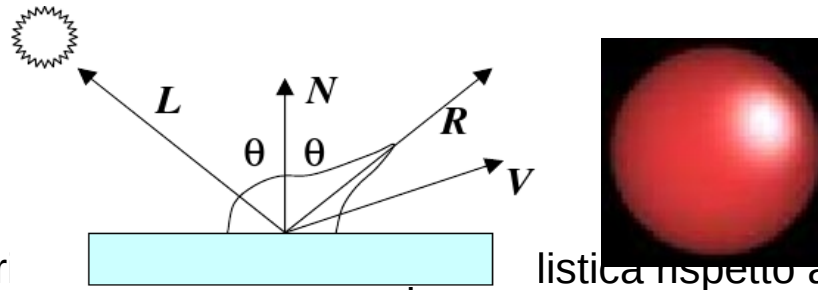
Riflessione Diffusiva

- Sorgenti luminose puntiformi:
 - posizione nella scena
 - intensità della luce emessa
- Per calcolare in P con normale N: $\vec{L} = |L - P|$
- Dipendenza solo da θ : $\cos \theta = \vec{L} \cdot \vec{n}$
- Si approssima la funzione di riflessione diffusa della superficie come una costante k_d dipendente dal materiale
- Equazione di illuminazione (solo diffusiva) $I = I_p k_d \cos \theta$ o meglio $I = I_p k_d (\vec{n} \cdot \vec{L})$
 - Si considera solo per valori di θ compresi tra 0 e $\pi/2$



Riflessione Speculare

- Novità sostanziale: riflettore non perfetto



- Approssimazione empiristica rispetto alla legge di Fresnel

- Conseguenza: specular highlight

- In pratica lo specular highlight è dato dalla luce riflessa nella direzione di vista, quindi la sua posizione sull'oggetto dipende dall'osservatore

- Dipendenza dall'angolo α compreso tra la direzione di riflessione ideale e la direzione di vista

- Riflessione massima per $\alpha = 0$
 - Decadimento più o meno rapido all'aumentare di α

- Questo comportamento si modella elevando alla n il coseno dell'angolo α

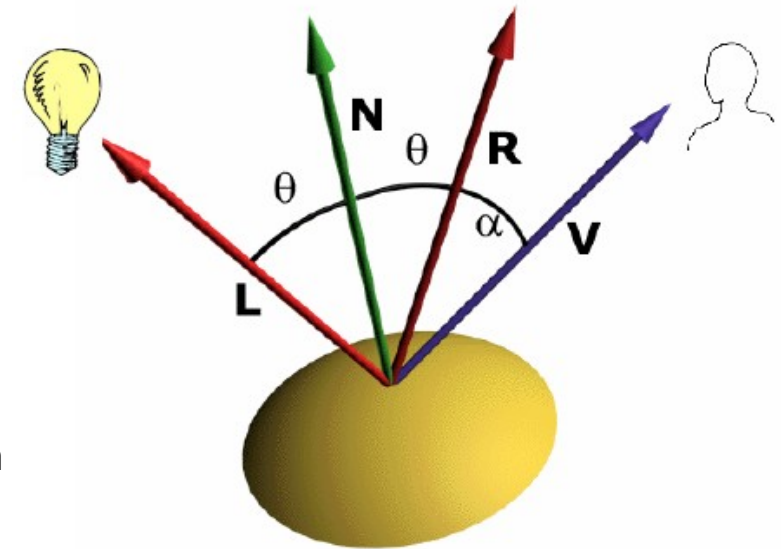
- Il parametro n è detto esponente di riflessione speculare (specular reflection exponent) del materiale

- Equazione di illuminazione (solo speculare)

$$I = I_p k_s \cos^n \alpha = I_p k_s (\vec{\mathbf{R}} \cdot \vec{\mathbf{V}})^n$$

- Parametro k_s modella il comportamento della superficie insieme a n

- Il vettore \mathbf{R} si calcola $\vec{\mathbf{R}} = 2\vec{\mathbf{N}}(\vec{\mathbf{N}} \cdot \vec{\mathbf{L}}) - \vec{\mathbf{L}}$



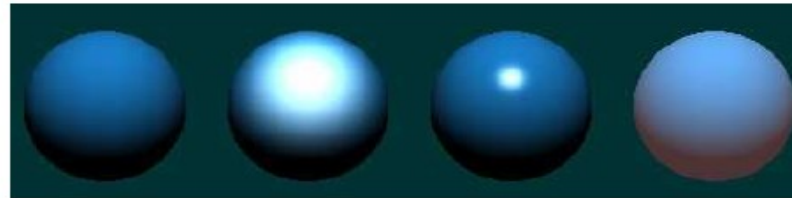
Componente Ambientale

- Le inter-riflessioni tra oggetti diversi nella scena non sono modellate in modo accurato dal modello di Phong
- Sono approssimate dalla componente:

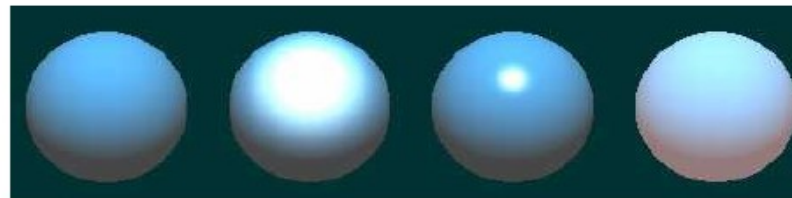
$$I = I_a k_a$$

- I_a modella la radiazione luminosa totale emessa nella scena
 - I_a è costante per tutti i punti di tutti gli oggetti
- k_a modella la riflettività del materiale
- La componente ambientale aggiunge realismo alla scena anche se è una grossolana approssimazione dell'effetto della luce indiretta.
-

Senza →















Con →



Equazione di Illuminazione

- Tutti i contributi descritti si vanno a sommare per calcolare l'equazione di illuminazione
 - Sommatoria su tutte le sorgenti luminose presenti nella scena

$$I = I_a k_a + \sum_p I_p [k_d (\vec{N} \cdot \vec{L}) + k_s (\vec{R} \cdot \vec{V})^n]$$

Phong	ρ_{ambient}	ρ_{diffuse}	ρ_{specular}	ρ_{total}
$\phi_i = 60^\circ$				
$\phi_i = 25^\circ$				
$\phi_i = 0^\circ$				

Fattore di Attenuazione

- Si può tenere conto dell'attenuazione dell'intensità dell'illuminazione all'aumentare della distanza utilizzando il fattore di attenuazione

$$f_{\text{att}} = \begin{cases} \frac{1}{c_1 + c_2 d_L + c_3 d_L^2} & \text{se } \frac{1}{c_1 + c_2 d_L + c_3 d_L^2} < 1 \\ 1 & \text{altrimenti} \end{cases} \quad I = I_a k_a + \sum_p f_{\text{att}_p} I_p [k_d (\vec{\mathbf{N}} \cdot \vec{\mathbf{L}}) + k_s (\vec{\mathbf{R}} \cdot \vec{\mathbf{V}})^n]$$

Rendering a colori

- Finora abbiamo considerato l'intensità luminosa, come passare al colore?
- Quando si utilizza una rappresentazione a colori RGB l'equazione viene calcolata in modo indipendente per ciascuna delle tre componenti cromatiche
 - Ipotesi: riflessione => no cambio di frequenza
- Quindi avremo (luce singola):

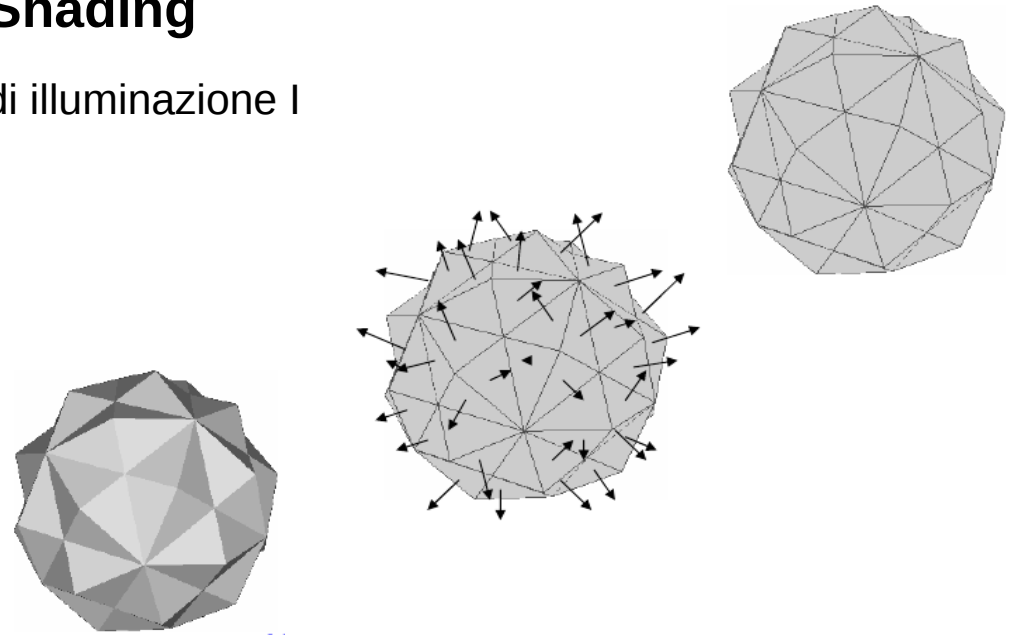
$$\begin{aligned} I_r &= I_{a,r} k_{a,r} + I_{p,r} [k_{d,r} (\vec{\mathbf{N}} \cdot \vec{\mathbf{L}}) + k_{s,r} (\vec{\mathbf{R}} \cdot \vec{\mathbf{V}})^n] \\ I_g &= I_{a,g} k_{a,g} + I_{p,g} [k_{d,g} (\vec{\mathbf{N}} \cdot \vec{\mathbf{L}}) + k_{s,g} (\vec{\mathbf{R}} \cdot \vec{\mathbf{V}})^n] \\ I_b &= I_{a,b} k_{a,b} + I_{p,b} [k_{d,b} (\vec{\mathbf{N}} \cdot \vec{\mathbf{L}}) + k_{s,b} (\vec{\mathbf{R}} \cdot \vec{\mathbf{V}})^n] \end{aligned}$$

Tecniche di Shading

- Il modello di Phong ci dice come calcolare l'interazione tra luce e materia senza utilizzare la Rendering Equation
- Adesso vediamo dove calcolare l'equazione d'illuminazione

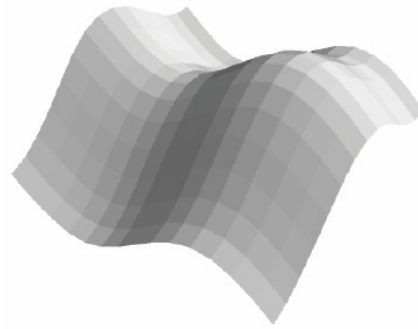
Flat Shading

- Dato l'oggetto per cui calcolare l'equazione di illuminazione I
- calcolare le normali in ogni faccia
- e calcolo I una sola volta per faccia
-
- Nelle seguenti condizioni:
 - sorgenti luminose solo direzionali
 - osservatore a distanza infinita dalla scena
- il flat shading costituisce la migliore approssimazione possibile.

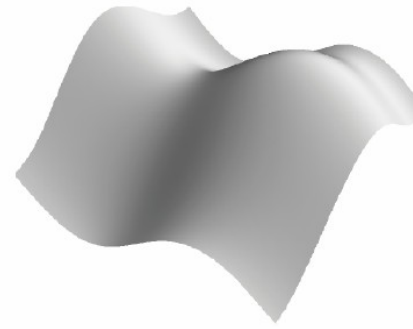


Flat Shading

- Il modello discreto rappresenta in modo approssimato una superficie curva e continua

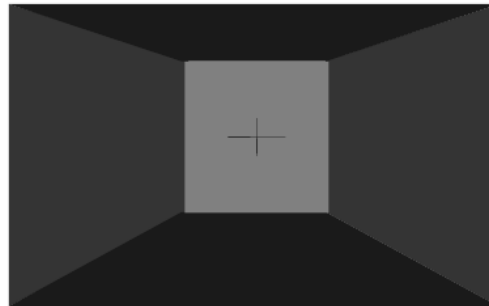
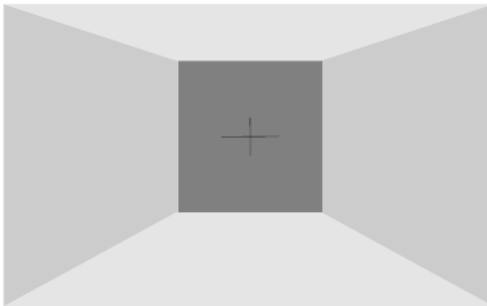


Com'è

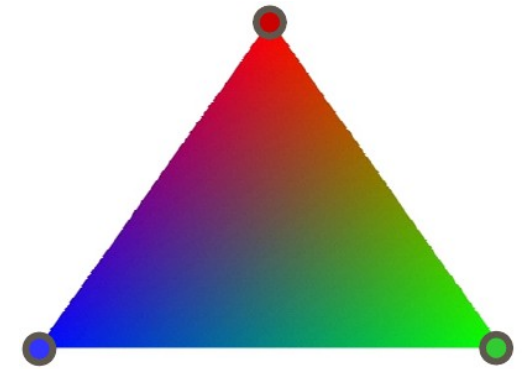


Come dovrebbe essere

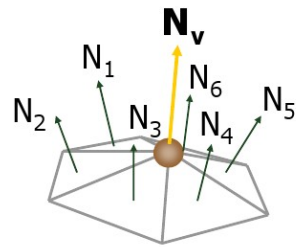
- Possibile soluzione: uso un numero elevato di facce
- Non funziona: si vedono comunque le discontinuità tra una faccia e la vicina.
- Mach Banding (è un effetto percettivo)
 - Un oggetto messo vicino ad uno più chiaro risulta più scuro e messo vicino ad uno più scuro risulta più chiaro



Gouraud Shading

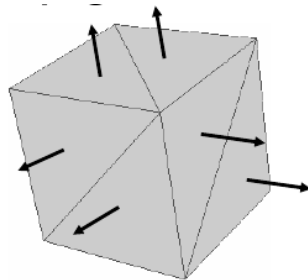


- Calcolare l'equazione di illuminazione solo in alcuni punti nodali
- Interpolare linearmente tra questi valori
 - Aggiungere all'algoritmo di rasterizzazione l'operazione di interpolazione nello spazio colore comporta uno sforzo minimo
 - Il risultato così ottenuto approssima molto il modello di Phong per superfici generiche rispetto allo shading costante
- Che normali utilizzo?
 - La normale alla faccia è bene definita
 - La normale al vertice la calcolo come media delle normali delle facce che incidono sul vertice

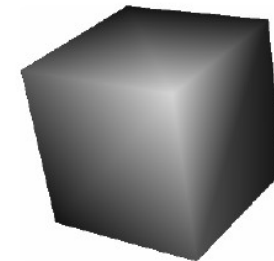
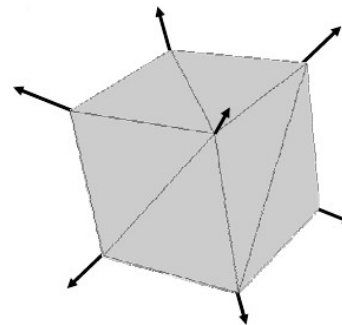


$$\vec{N}_v = \frac{\sum_i \vec{N}_i}{|\sum_i \vec{N}_i|}$$

- Problema: gli spigoli “veri”?



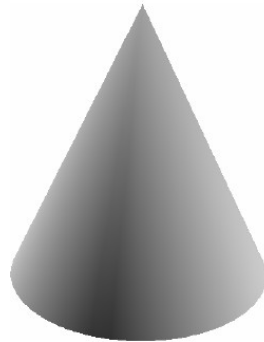
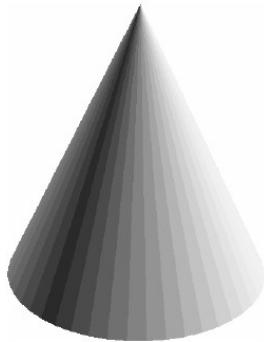
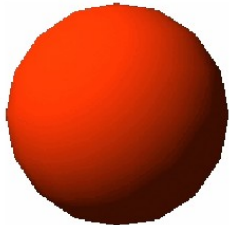
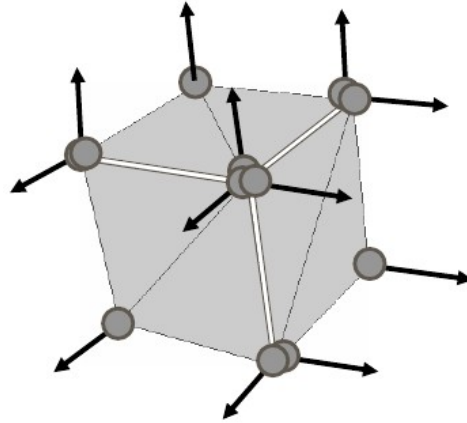
shading costante



Gouraud shading

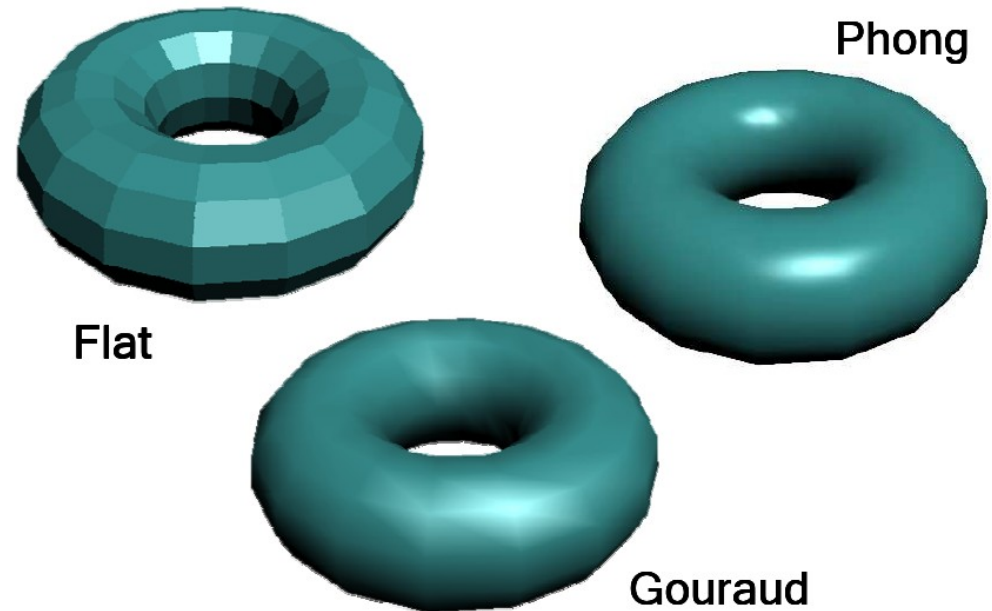
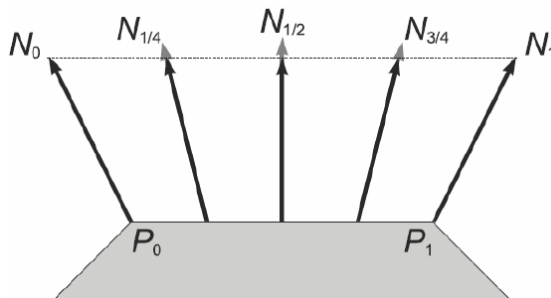
Gouraud Shading

- Soluzione: si utilizzano normali diverse per i due lati dello spigolo
 - La struttura dati deve memorizzare le adiacenze e le diverse tipologie



Phong Shading

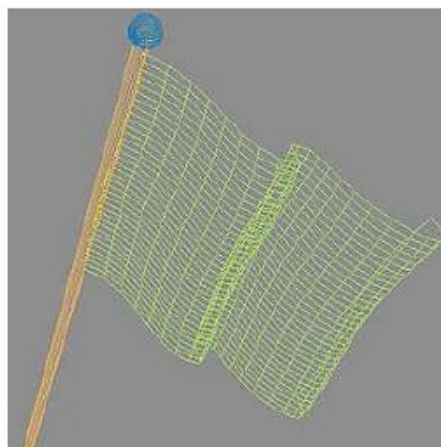
- Gouraud shading:
 - ottimo rapporto complessità/benefici
 - Risultati non eccezionali per superfici dotate di un alto coefficiente di riflessione speculare
- Problema: con un alto indice di riflessione (n) lo specular highlight risulta piccolo,
 - l'highlight può "propagarsi" per tutta la faccia (per interpolazione) se cade su di un vertice
 - Non viene disegnato se cade internamente alla faccia
- Soluzione: si interpola nello spazio delle normali e si calcola l'equazione di illuminazione in ogni pixel



- Fenomeno percettivo: la nostra vista tridimensionale monoculare è sensibile alle variazioni di illuminazioni (3D hints) più che alla posizione assoluta.

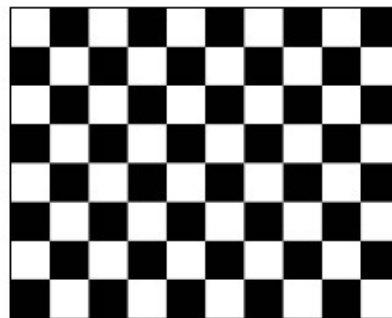
Texturing

- Il concetto di texturing è importante
- Si tratta di “modulare” un qualsiasi attributo del vertice in modo da ottenere l’effetto visivo desiderato
- Attributi modulabili: colore, normali, trasparenza, un parametro del modello di illuminazione...
- L’attributo più immediato per conferire ulteriore dettaglio alla superficie rispetto a quello che abbiamo visto finora è il colore!
 - La modulazione dell’attributo colore prende il nome di texture mapping!
 - Modulazione della trasparenza: la tessitura è una “alpha-map”
 - Modulazione della normale normale (componenti: X-Y-Z): la tessitura è una “normal-map” o “bump-map”
 - Modulazione dei parametri di specularità: la tessitura è una “shininess-map”



geometria 3D
(mesh di quadrilateri)

+



RGB texture 2D
(color-map)

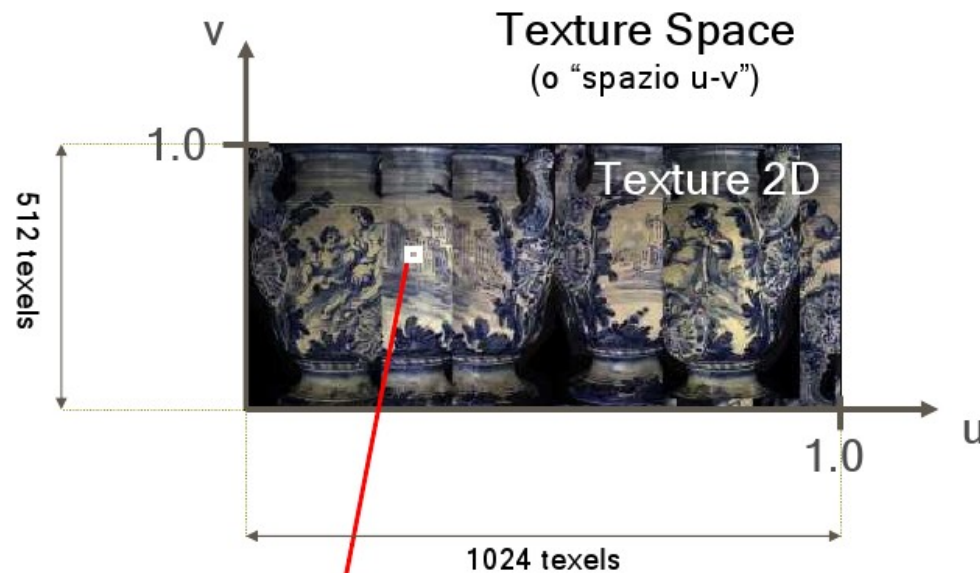
=



Texture Mapping (brevissima storia)

- 1974 introdotto da Ed Catmull
 - nella sua Phd Thesis
- Solo nel 1992 (!) si ha texture mapping in hardware
 - Silicon Graphics RealityEngine
- Dal 92 a oggi ha avuto aumento rapidissimo della diffusione
 - strada intrapresa soprattutto dall'hardware grafico
- Oggi è una delle più fondamentali tecniche di rendering

Spazio Texture (“spazio u-v”)

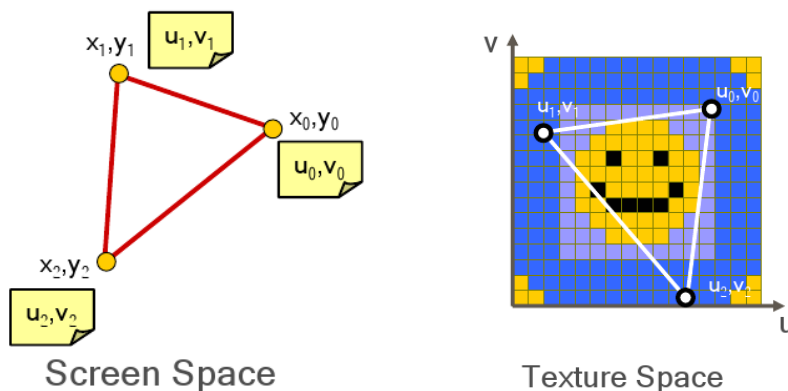


texel

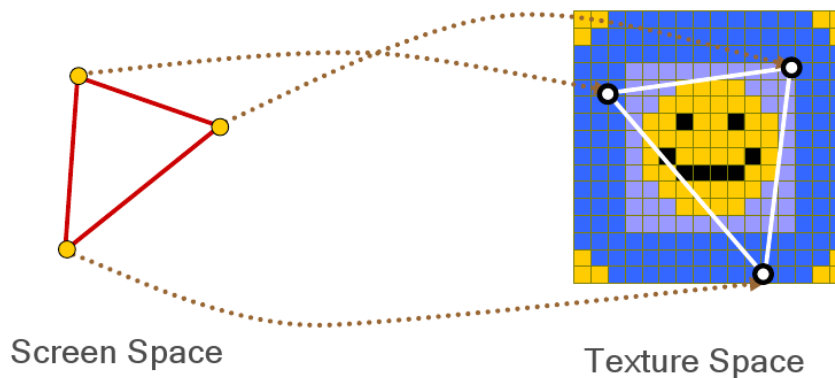
Una Texture è solitamente definita
in coordinate normalizzate $[0,1] \times [0,1]$
nello spazio parametrico della texture

Texture Mapping

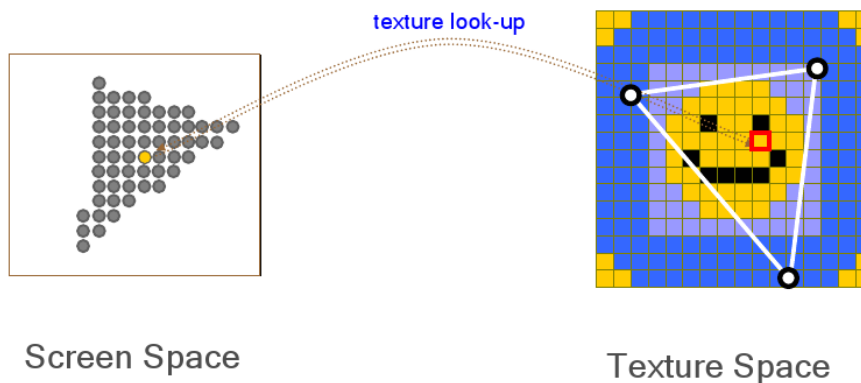
- Ad ogni vertice (di ogni triangolo) assegno le sue coordinate u,v nello spazio tessitura



- Così in pratica definisco un mapping fra il triangolo e un triangolo di tessitura

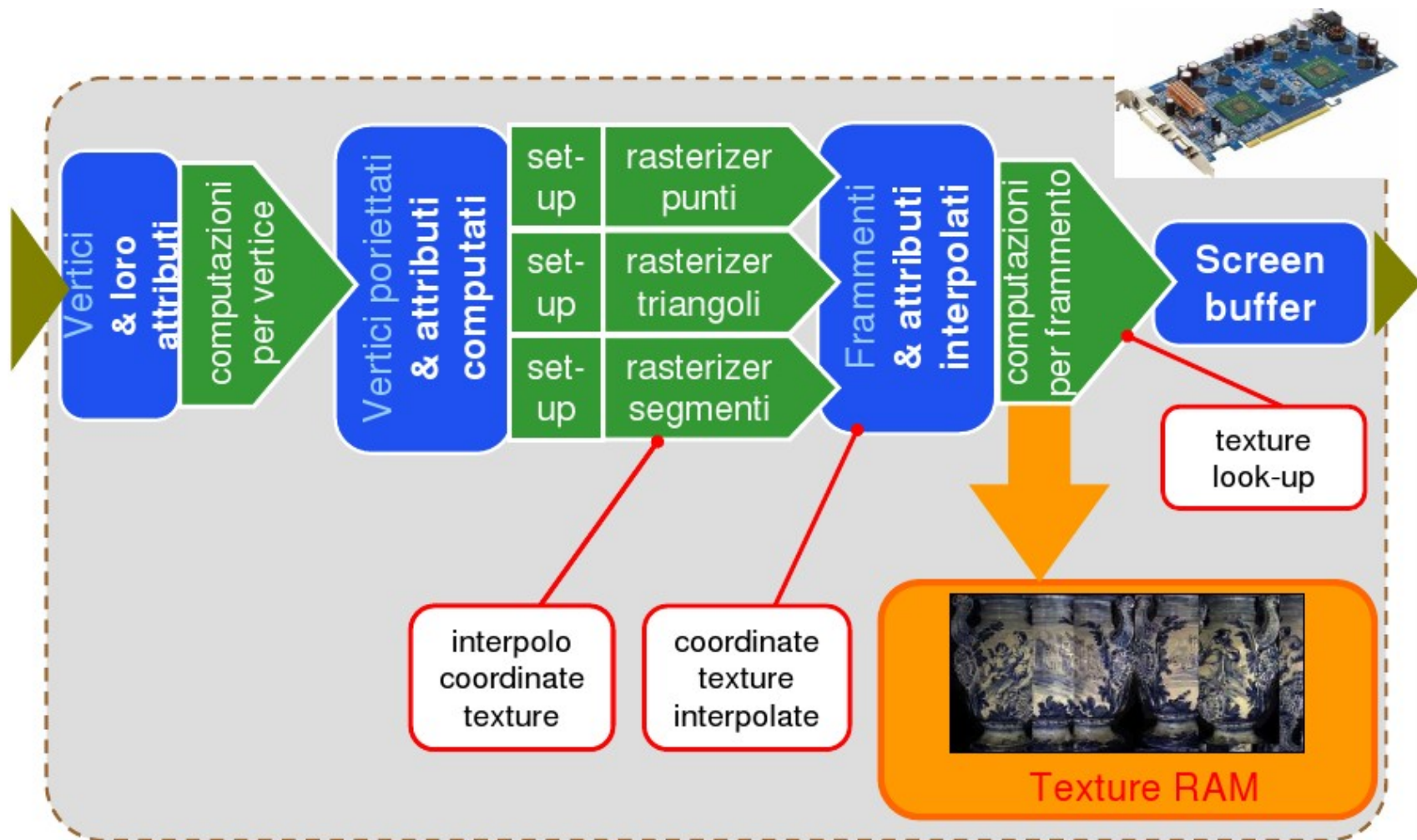


- Ogni vertice (di ogni triangolo) ha le sue coordinate u,v nello spazio tessitura



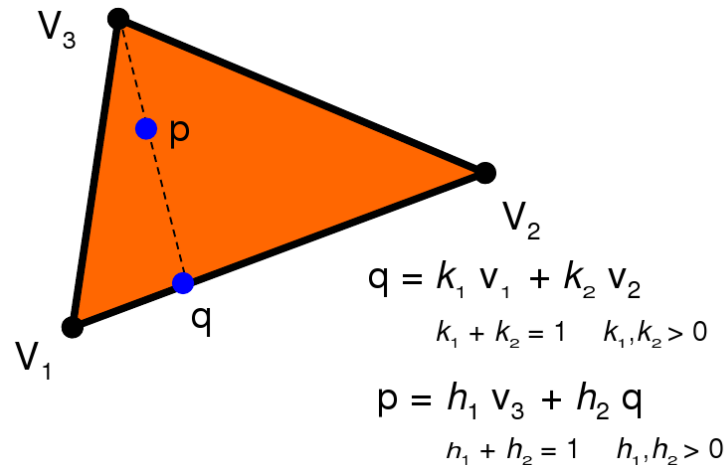
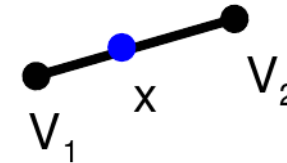
Texture RAM

- Nelle operazioni per frammento si può accedere ad una RAM apposita: la Texture RAM strutturata in un insieme di Textures (“tessiture”)
- Ogni tessitura è un array 1D, 2D o 3D di Texels (campioni di tessitura, prende il nome dai pixels) dello stesso tipo



Interpolazione degli Attributi

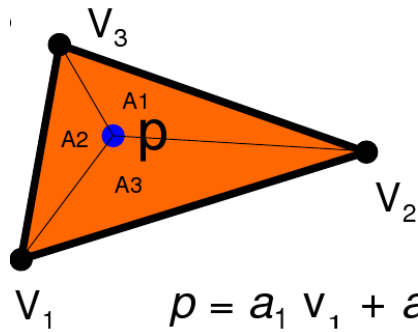
- Di norma si utilizzano le coordinate baricentriche...
- Ma cosa sono le coordinate baricentriche?
- Ricordiamo che un segmento si può scrivere come la combinazione lineare di due punti:
 - $x = a v_1 + b v_2$ con a e b scalari positivi e con $a + b = 1$ (quindi $0 \leq a \leq 1$ e $0 \leq b \leq 1$)
- Lo stesso lo possiamo fare per i punti di un triangolo:



- Un triangolo di vertici $v_1 v_2 v_3$ è l'insieme di tutti i punti x esprimibili come
 - $x = a_1 v_1 + a_2 v_2 + a_3 v_3$
 - a_1, a_2, a_3 scalari positivi con $a_1 + a_2 + a_3 = 1$

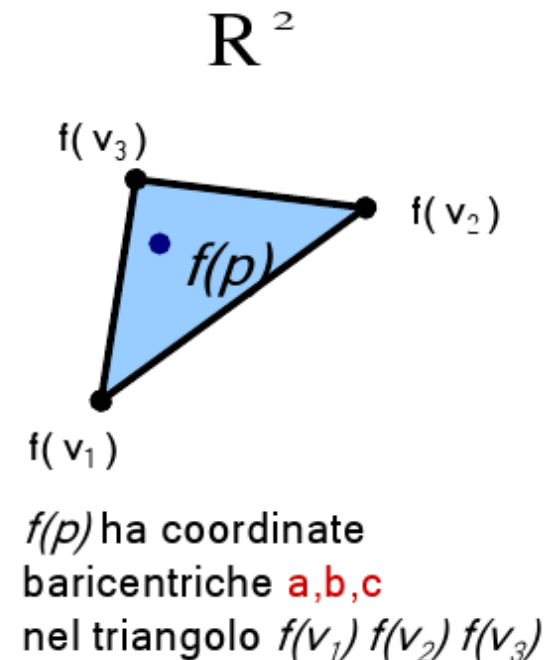
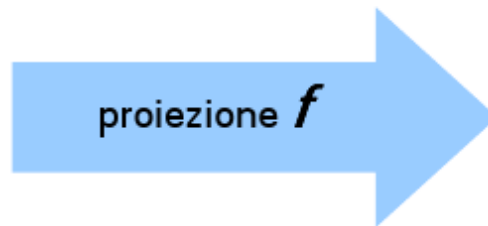
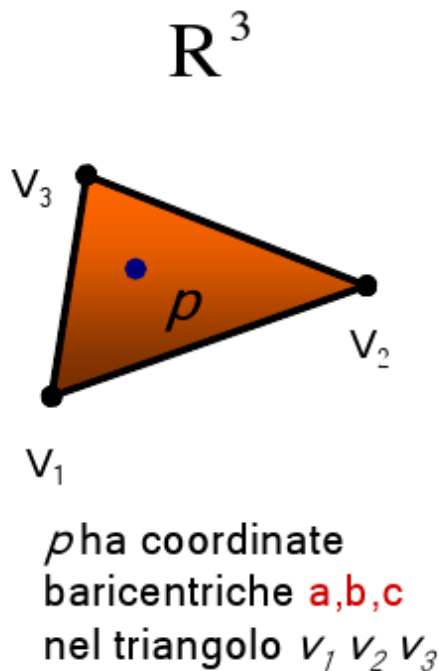
Coordinate Baricentriche

- Quali sono le coord. baricentriche di un punto p ?



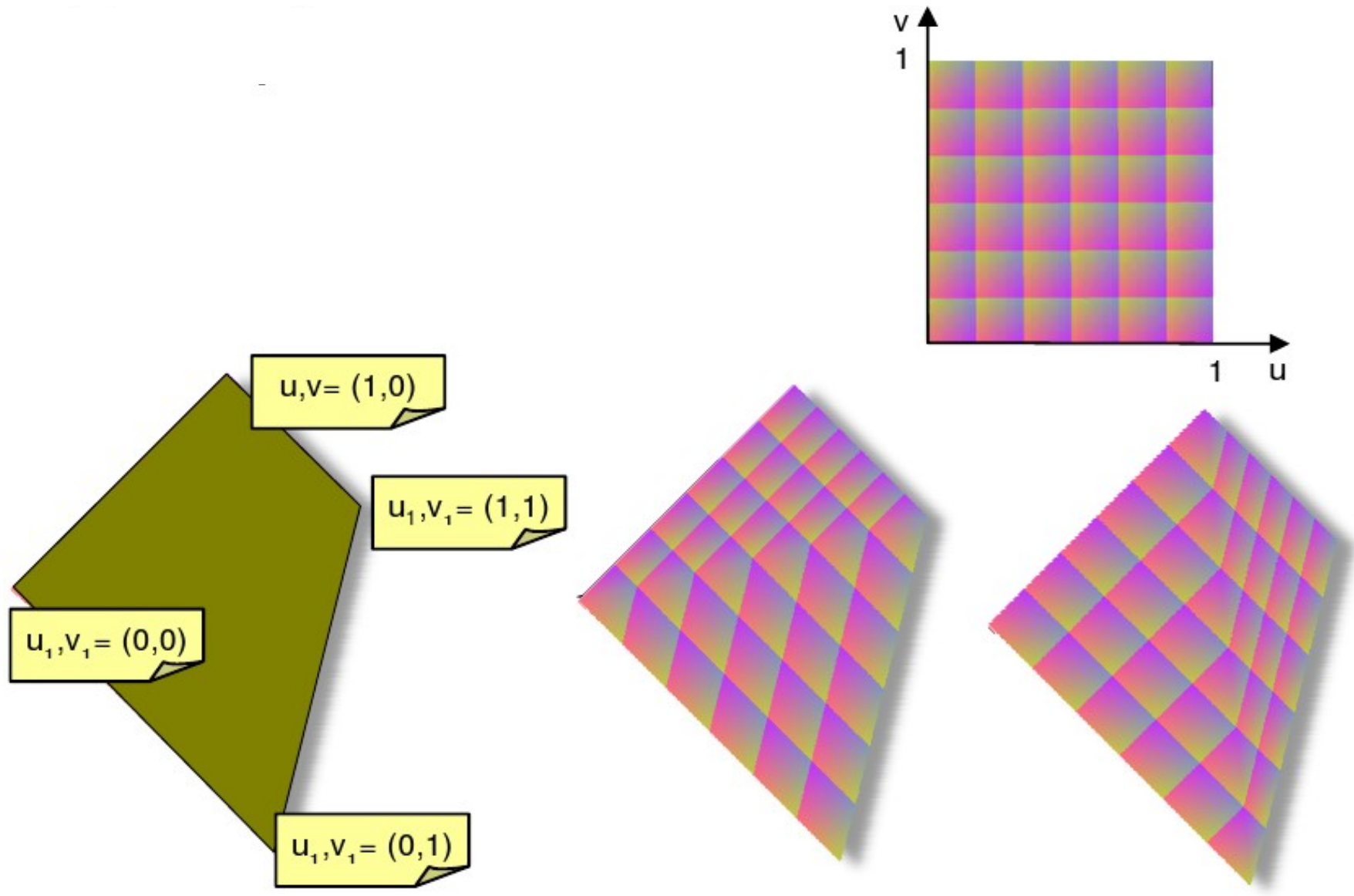
$$a_1 = \frac{A_1}{A_{tot}} \quad a_2 = \frac{A_2}{A_{tot}} \quad a_3 = \frac{A_3}{A_{tot}} \quad A_{tot} = A_1 + A_2 + A_3$$

- Possiamo usare le coordinate baricentriche di un punto per interpolare un attributo definito sui vertici



Deformazione prospettica

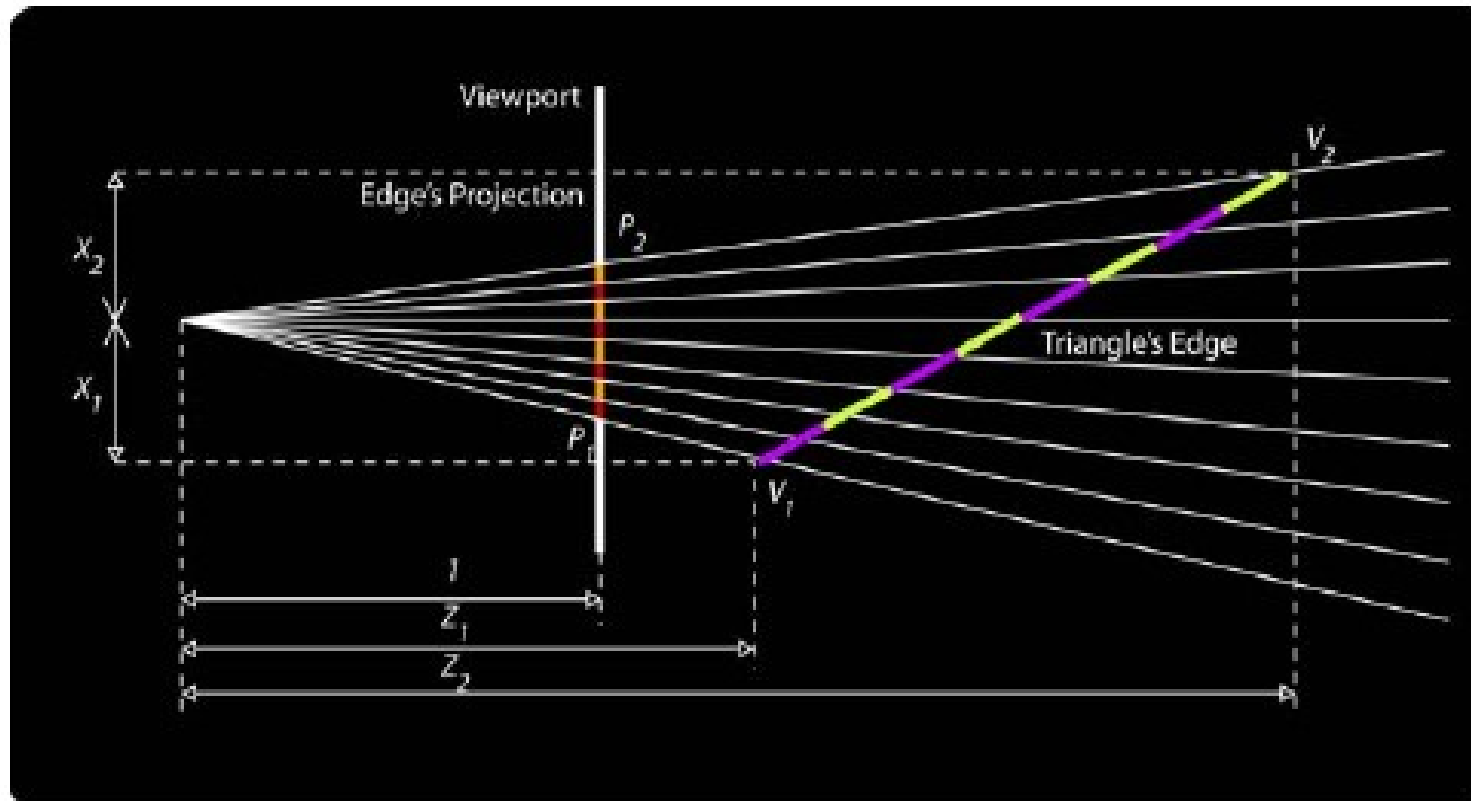
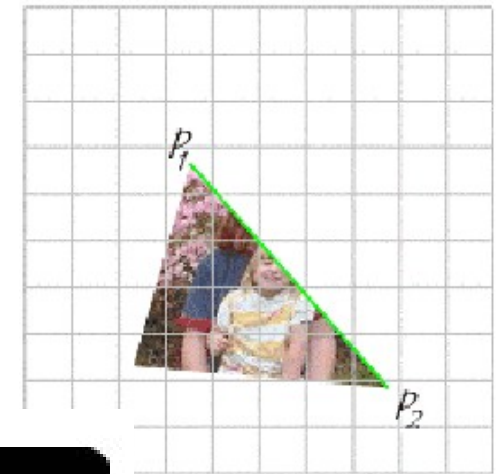
- L'interpolazione secondo le coordinate baricentriche non funziona quando interpoliamo le coordinate texture a causa della proiezione prospettica...



- Funziona però per gli altri attributi (es. normali)

Cosa succede?

- Consideriamo un qualsiasi segmento sulla superficie da mappare con la tessitura
- Intervalli regolari lungo il piano immagine corrispondono a intervalli irregolari nel modello



- Dobbiamo interpolare nello spazio 3D e non nelle coordinate schermo!!

Correzione Prospettica

- p ha coordinate baricentriche c_0 c_1 c_2 ($p = c_0 v_0 + c_1 v_1 + c_2 v_2$)

- $V_0 = (x_0, y_0, z_0, w_0)^T$

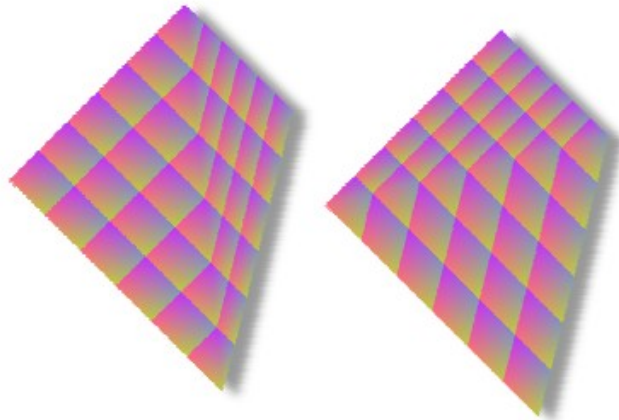
- Attributi di p

- Senza correzione prospettica
(interpolando nello spazio immagine)

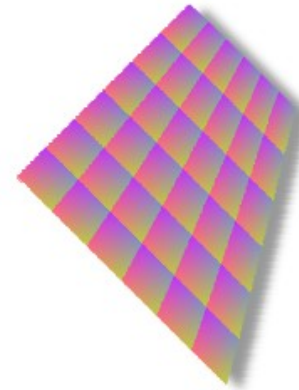
$$A_p = c_0 A_0 + c_1 A_1 + c_2 A_2$$

- Con correzione prospettica
(interpolando nello spazio 3D)

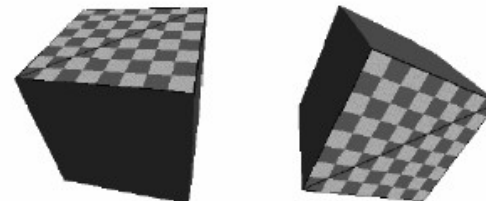
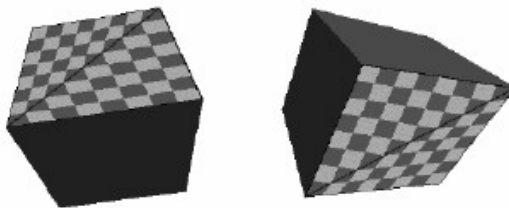
$$A_p = \frac{\frac{c_0}{z_0} A_0 + \frac{c_1}{z_1} A_1 + \frac{c_2}{z_2} A_2}{\frac{c_0}{z_0} + \frac{c_1}{z_1} + \frac{c_2}{z_2}}$$



senza



con

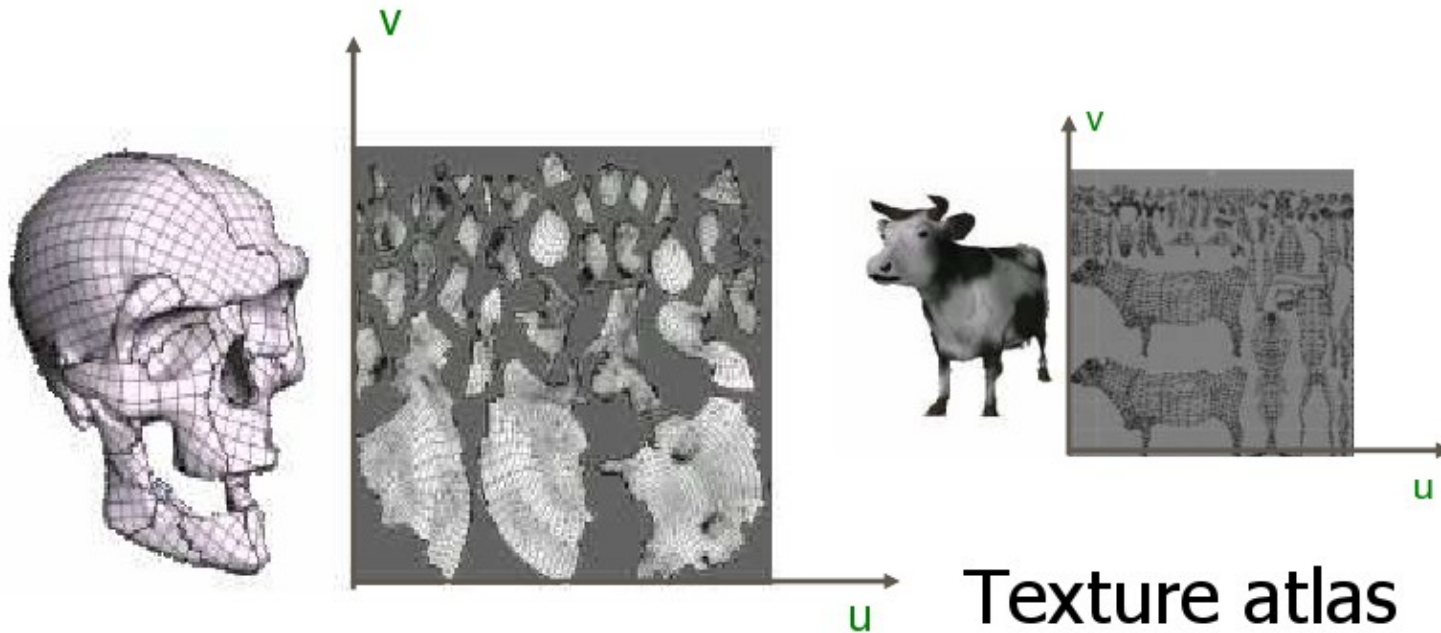


Come assegnare le coordinate texture?

- Soluzioni:
 - Calcolare le coordinate textures on-the-fly durante il rendering...
 - Precomputarle (e salvarle insieme alla mesh)
 - Spesso le assegna il modellatore...
- Non esiste una soluzione ideale, dipende dall'applicazione che stiamo progettando
 - Modelli con una sola texture l'avranno precomputata, per altri che variano dinamicamente l'assegneremo in rendering

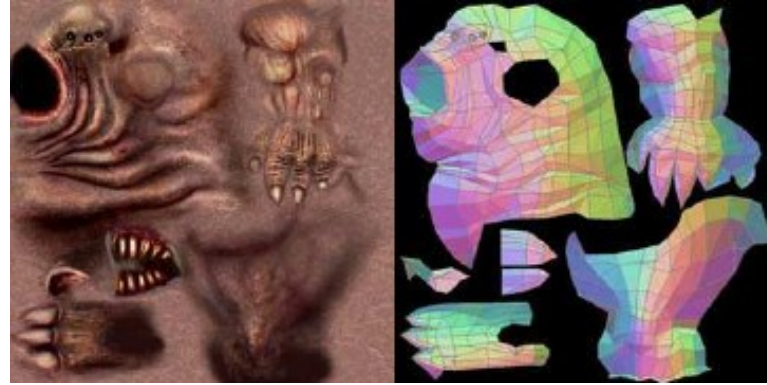
UV Mapping: problema difficile

- Assegnare una coppia di coordinate textures ad ogni vertice della mesh in preprocessing



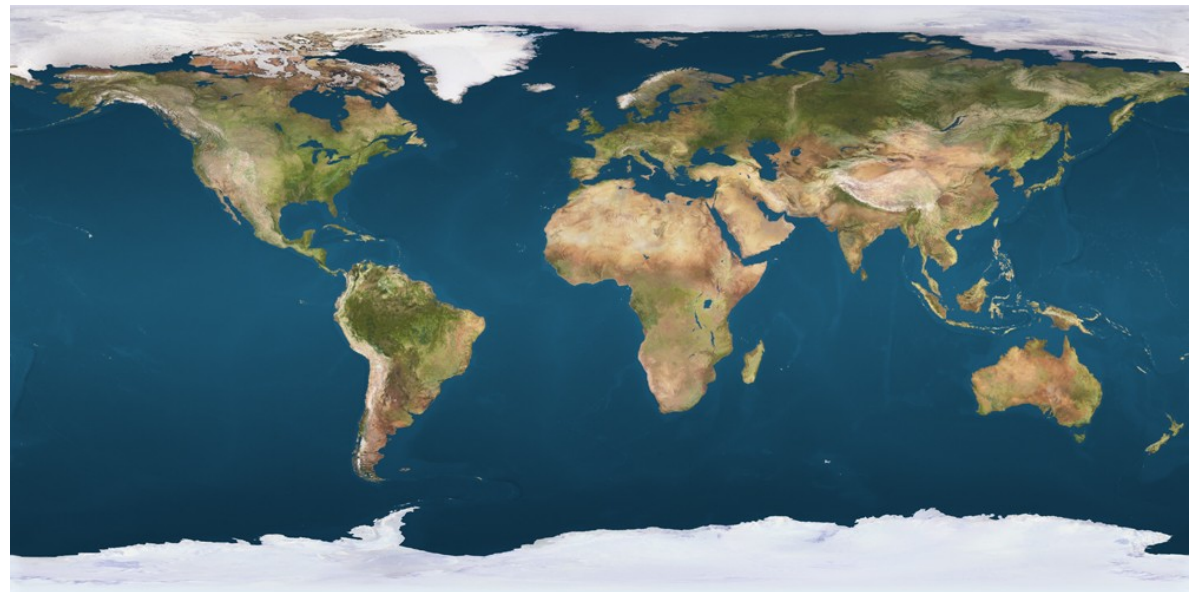
Texture Atlas

- fatto a mano, o automatizzato, componente importante del design 3D



- Ogni mappatura comporta deformazioni

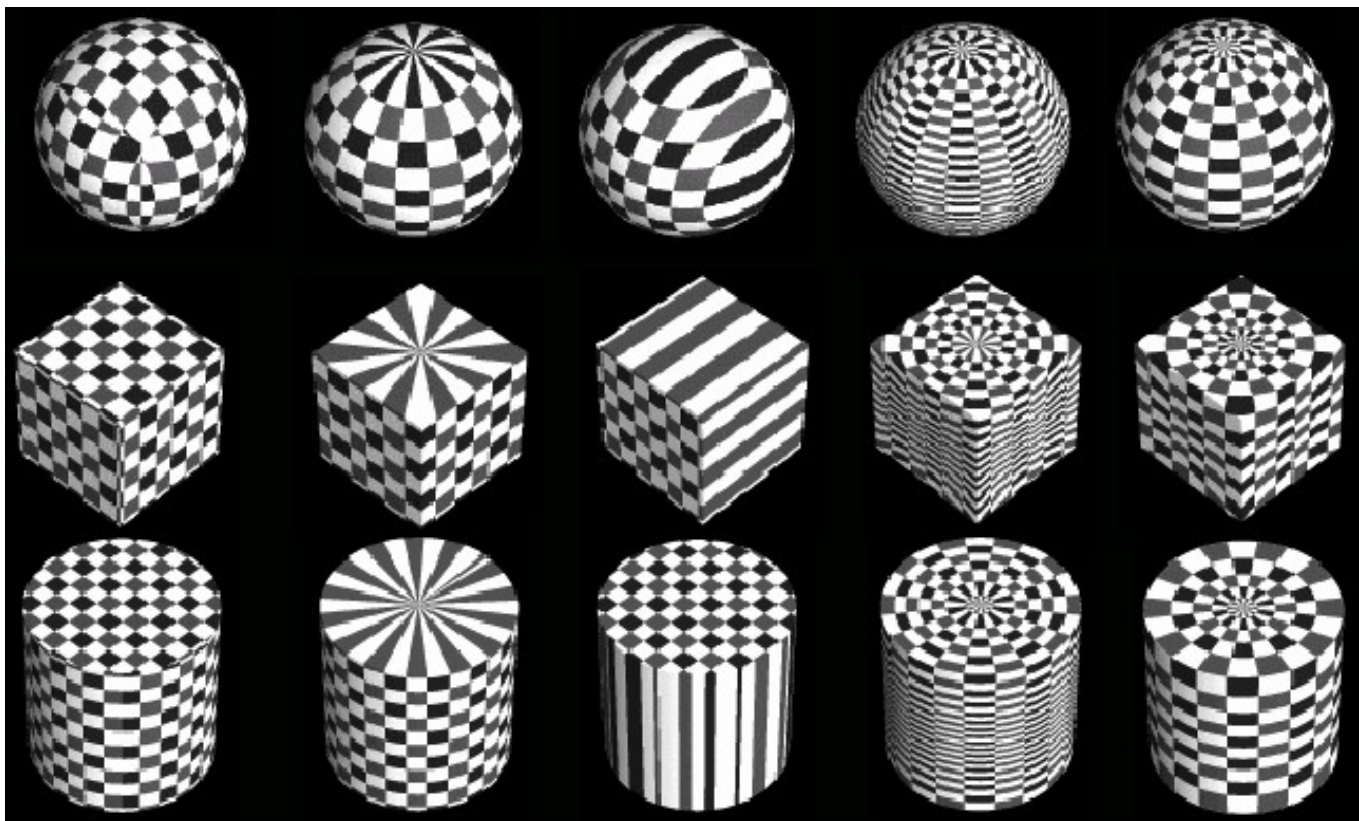
- Devono essere assorbite dalla texture



- Creazione della mappa a bassa distorsione problema aperto studiato da secoli (cartografia)

Creazione automatica delle coordinate texture

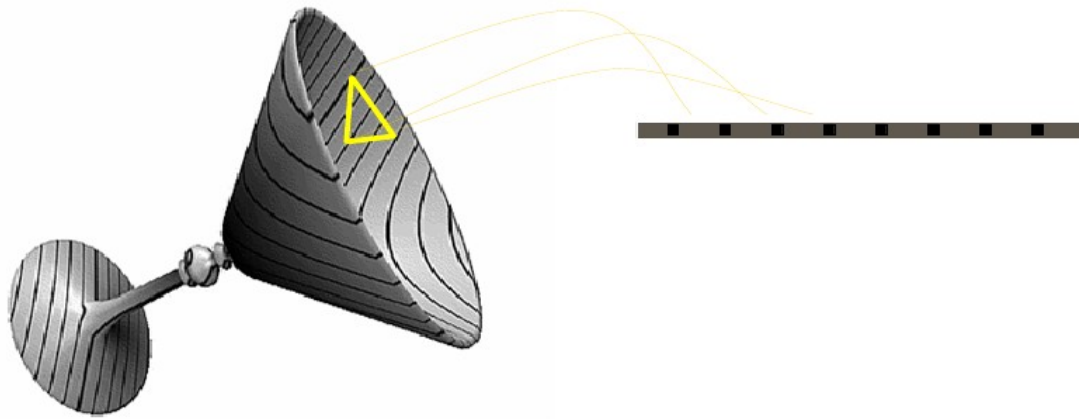
- Si utilizza una funzione di proiezione da (x,y,z) a (u,v) in coordinate oggetto o vista



- Proiezione planare (lungo asse x) $(x, y, z) \rightarrow (u, v)$ con $u = y, v = z$
- Proiezione cilindrica $(x, y, z) \rightarrow (u, v)$ con $u = \text{atan2}(x, z), v = y$
- Proiezione prospettica
 - Modellazione da fotografie
 - Usare foto come textures
- ...

Texture non planari

- Le texture possono anche essere funzioni 1D...



... o 3D

