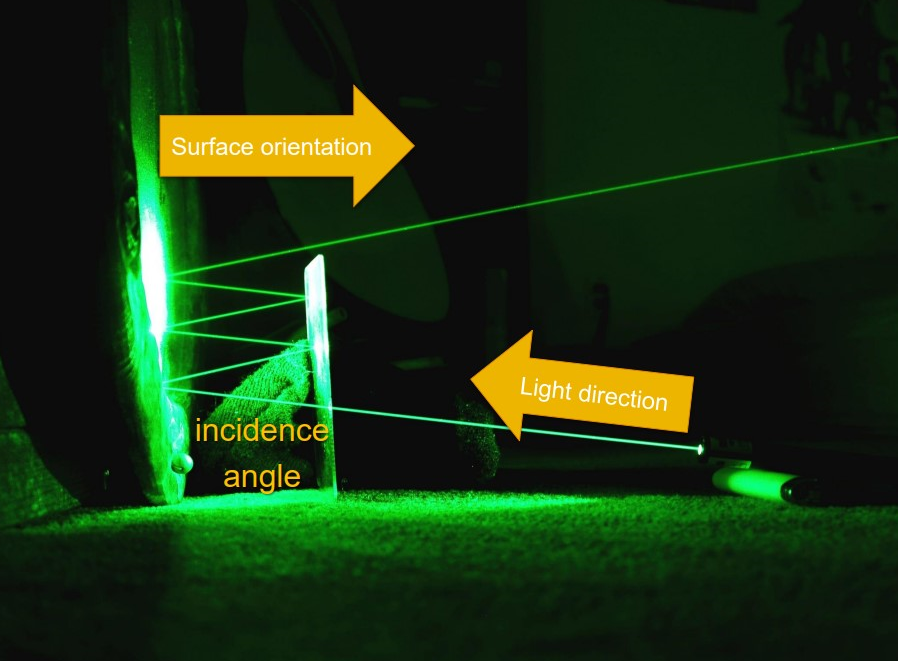
**Lighting**



Per ogni vertice si specificano le coordinate e il vettore normale, che avviene sempre per vertice

Immagine che contiene testo

Descrizione generata automaticamente

I vertici appartengono a un triangolo e quindi appartengono ad un piano.

L’output è un vertice colore (rgb)

Il modello di illuminazione calcola per ogni vertice e non per ogni pixel

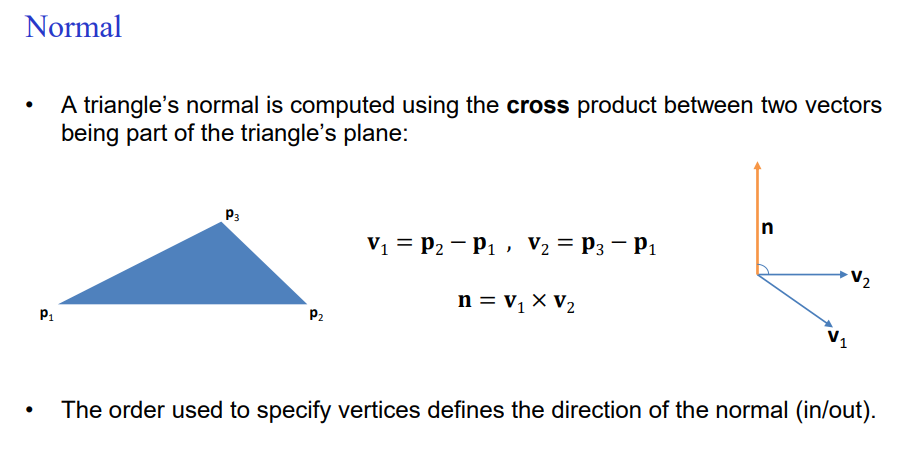


Immagine che contiene testo, tavolo da biliardo, tavolo, tavolo da lavoro

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamentePer ottimizzare il rendering le facce non visibili non vengono renderizzate, questa tecnica è chiamata

**face culling**

glEnable(GL\_CULL\_FACE); il lato opposto delle faccia della mesh non viene renderizzata

glCullFace(flag);

flags are GL\_BACK (default), GL\_FRONT, and GL\_FRONT\_AND\_BACK

**Illuminazione teoria**

glColor non si usa più per renderizzare i vertici, sene occupa openGL in base all’illuminazione

Immagine che contiene testo

Descrizione generata automaticamente

Si fa per ogni vertice, e tutti i vertici che creo avranno come vettore normale l’ultimo settato con glNormal3\*

Il vettore bisogna calcolarlo separatamente, e va normalizzato.

I vettori sono specificati per ogni vertice

openGl ruota, ”scala” anche il vettore della normale in base alla matrice alla matrice passsata con glLoadMatrix.

Lo fa calcolando una matrice inversa trasposta della parte 3x3 della matrice.

Se si fa uno scaling il vettore normale potrebbe diventare un vettore non più normalizzato, per auto normalizzarlo in questa situazione su può abilitare **glEnable(GL\_NORMALIZE)** per farla calcolare automaticamente in fase di scaling, **da fare sempre.**

Il motivo per cui calcolo un vettore normale per vertice è perché il modello di illuminazione calcola la media di tutti i vettori normali di quel vertice, in modo da avere l’illuminazione distribuita sulle varie facce.

Il caso opposto è vedere la mesh con uno scalino per ogni faccia e un colore sconnesso tra le varie facce.

Questa cosa non vale se si vuole visualizzare in modo netto uno spigolo.

Queste operazioni non vanno gestite da codice, ma dipendono dai tool di 3d design.

A livello microscopico tutte le superfice possono essere rappresentate d auna serie di micro specchi chiamati microfacets.

Questo consente la riflessione della luce nella stessa direzione.

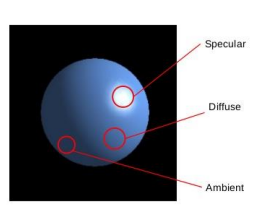


Immagine che contiene testo

Descrizione generata automaticamente

**Materiali**

**Ambient**: è un valore che assume tutto ciò che non è illuminato

Il coseno tra due vettori= prodotto scalare tra due vettori normalizzati

**Diffuse**: indica la luminosità, si basa sulla legge del coseno di Lambert

Coinvolge Vettore normale e un qualunque vertice

Non è importate la posizione dell’osservatore

Viene riflessa in posizioni diverse per via della superfice scatterosa

**Specular**: la luce rimbalza sulla superfice e la riflette in una direzione specifica

I raggi che Rimbalzano nella posizione dell’osservatore sono chiari

**Emissive:**

Sono materiali che emettono luce.

È un glColor con priorità sul modello di illuminazione

Un materiale emissivo non sostituisce una luce, la rasterizzazione mantiene il colore costate.

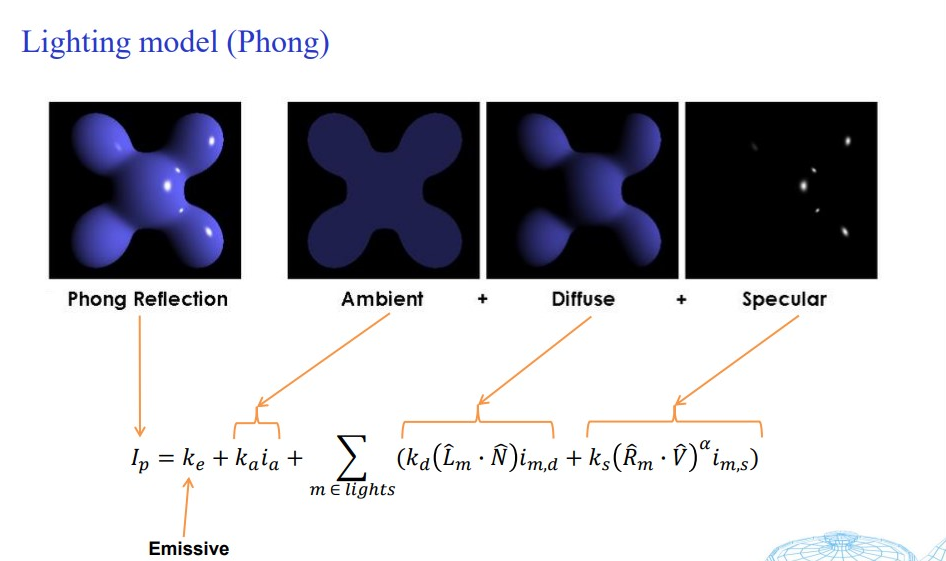
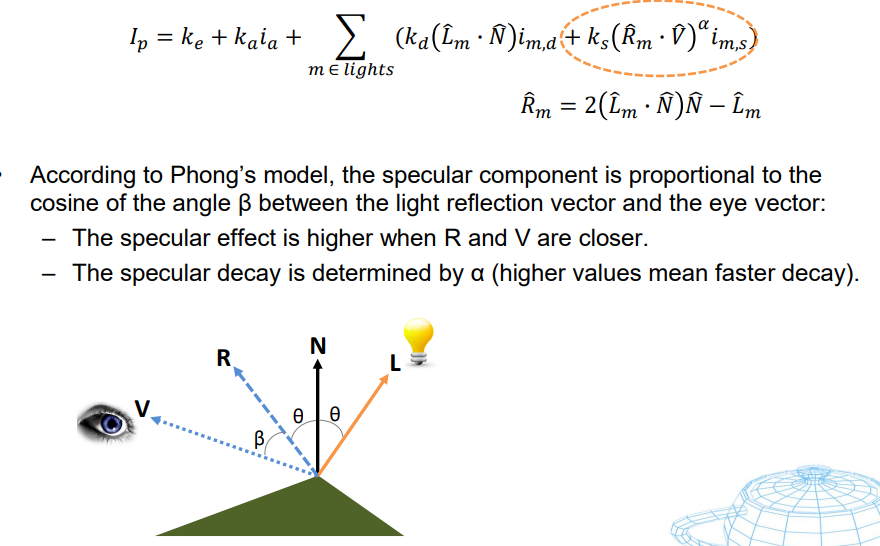


Immagine che contiene testo

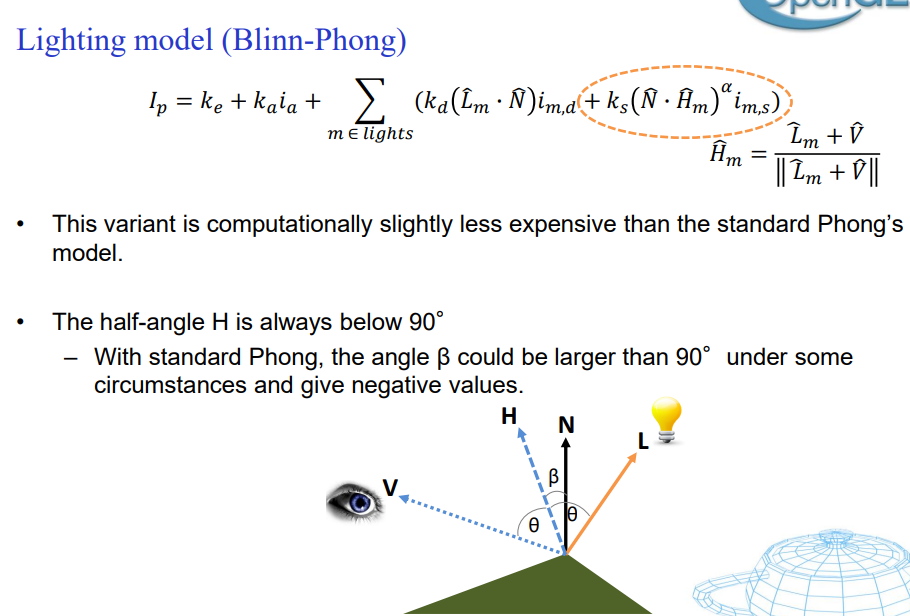
Descrizione generata automaticamente



Il vettore di riflessione perfetto è un vettore che ha lo stesso angolo di arrivo

Immagine che contiene testo

Descrizione generata automaticamente



Il vettore H è il vettore che sta tra L e V, la luminosità è data da quanto è vicino H ad L.

Con Blinn-Phong H non ha mai un angolo maggiore di 90 gradi

openGl usa Blinn-Phong

**Tipi di luci**

**Directional Light**

È una luce con una direzione ma è talmente lontana che i raggi sono paralleli, è detta anche luce infinita.

Quindi fa un’illuminazione globale

Il vettore è uguale per tutti i vertici.

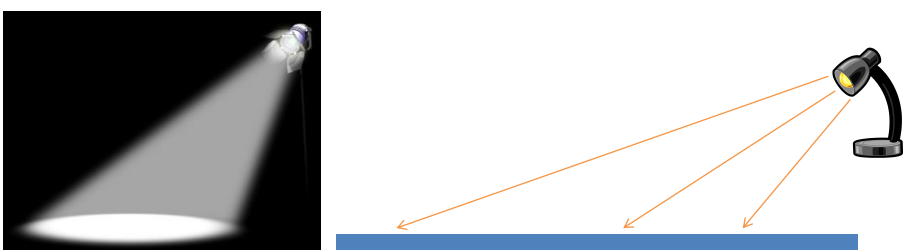
**Omnidirectional light:**

è un punto che illumina a 360 gradi intorno a sé.

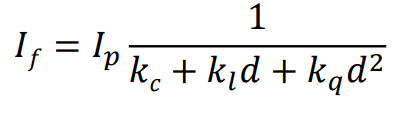
Luce irradiante in un punto (luce puntiforme)

**Spotlight:**

luce con effetto conico, è punti forme, ma illumina in una sola direzione



**Attenuazione**

****

**Immagine che contiene testo, screenshot

Descrizione generata automaticamente**

Le fonti di luce perdono luminosità all’allontanarsi

– glEnable(GL\_LIGHTING); abilita l’illuminazione, da disattivare per il rendering 2D (testo)

– glEnable(GL\_LIGHT#); abilita la singola luce al posto di # metto il numero della luce

Al massimo si possono avere n luci attive che di default è 8

glGetIntegerv(GL\_MAX\_LIGHTS, &maxNrOfLights); You can query the maximum number of supported lights via see

GL\_LIGHT0 di default è una luce direzionale

Immagine che contiene tavolo

Descrizione generata automaticamente

Shiness valore tra 1 e 128 maggiore è più liscia sarà la superfice

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamenteI vettori passati vengono moltiplicati per la matrice di modelview

Immagine che contiene testo

Descrizione generata automaticamente

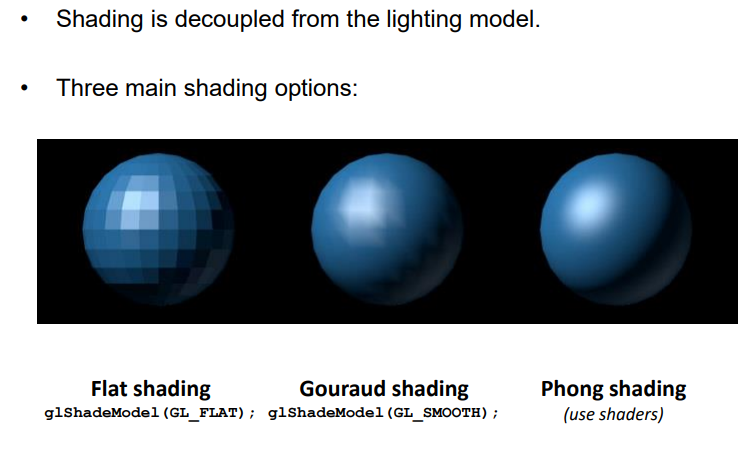
L’angolo di cutoff è l’angolo di apertura del cono

Immagine che contiene testo

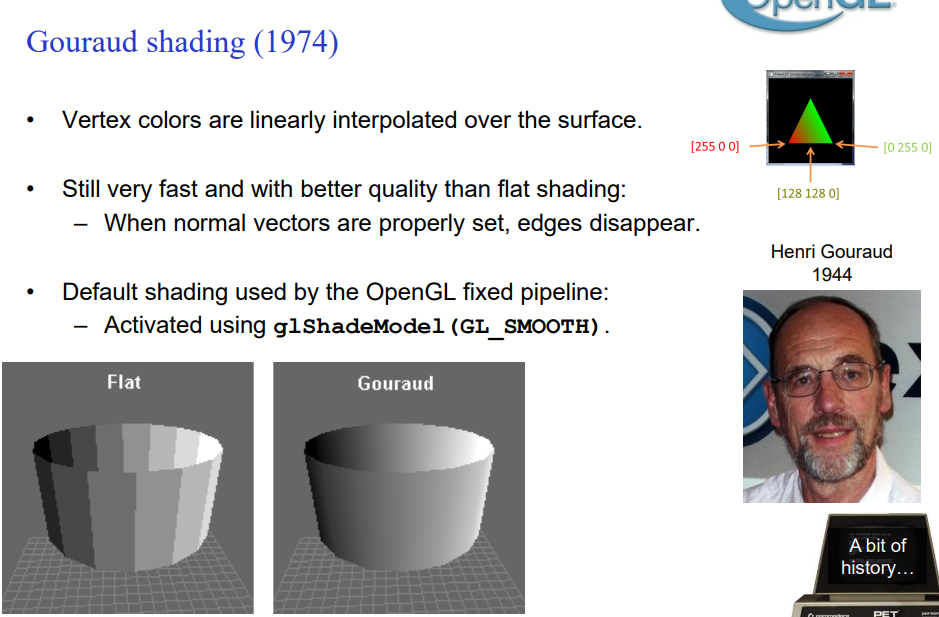
Descrizione generata automaticamente

Immagine che contiene testo

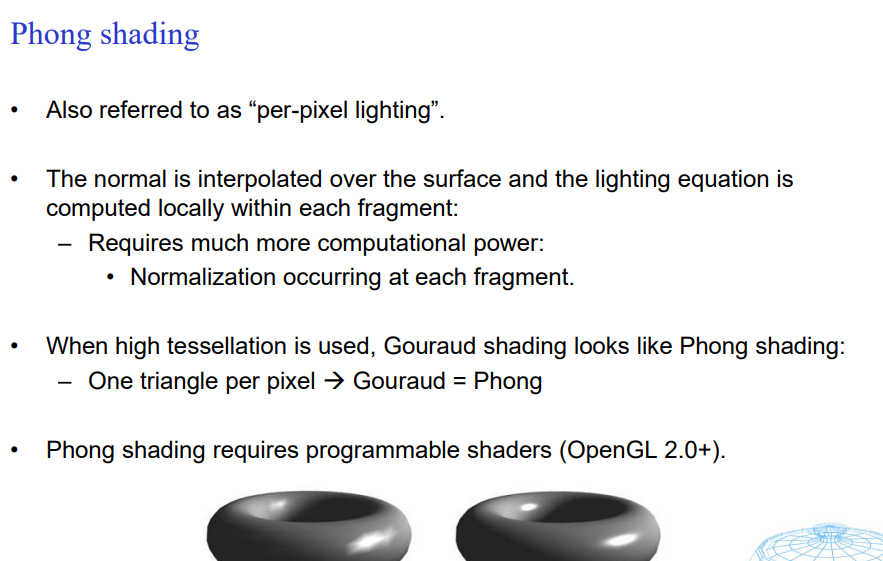
Descrizione generata automaticamente



Nel flat shading il colore dell’ultimo vertice passato sarà il colore usato per riempire il triangolo



Interpola i colori linearmente sulla superfice



Calcola il modello di illuminazione per ogni pixel