

# Grafica Computazionale

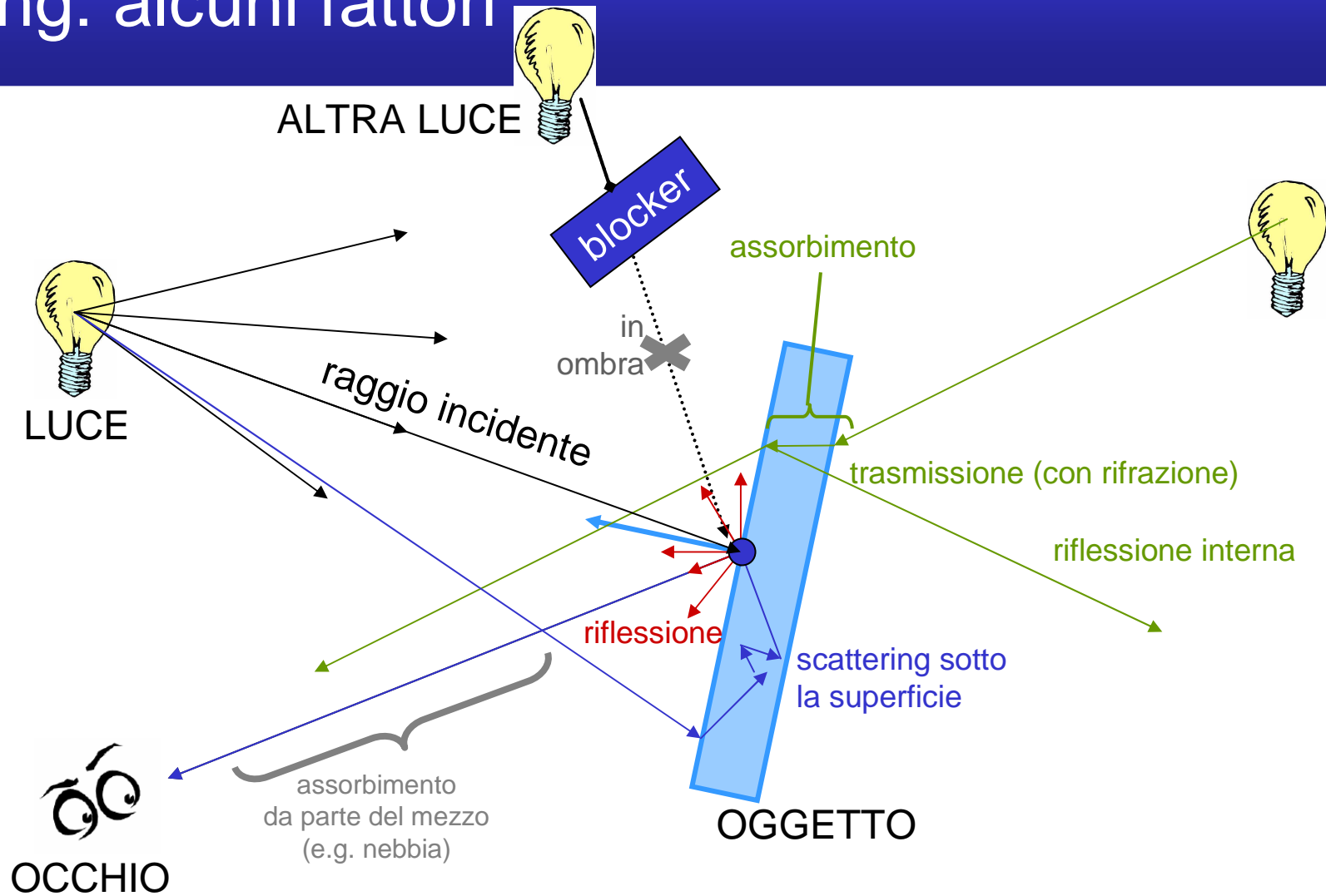
## Lighting e shading

Fabio Ganovelli

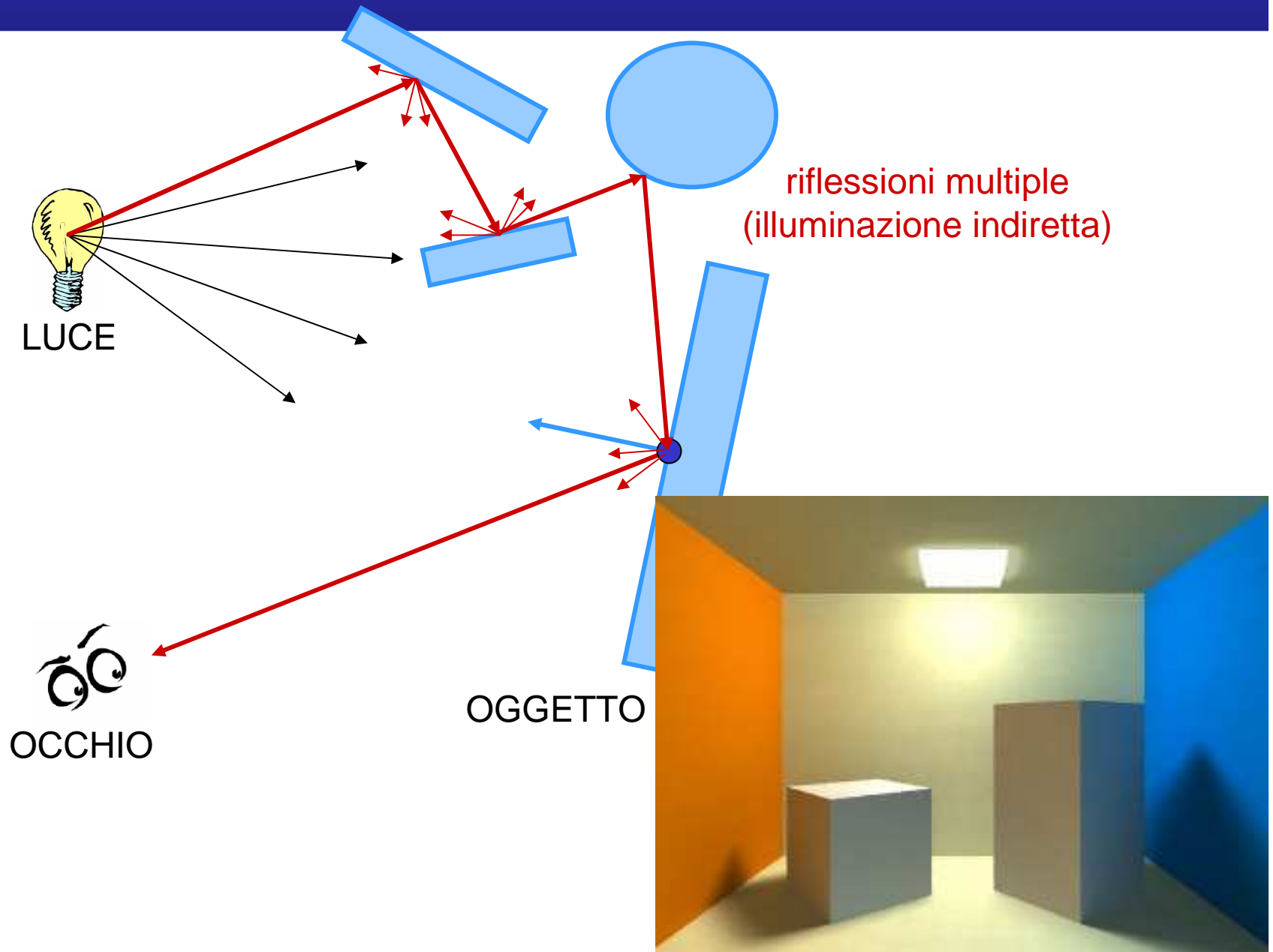
[fabio.ganovelli@isti.cnr.it](mailto:fabio.ganovelli@isti.cnr.it)

a.a. 2005-2006

# Lighting: alcuni fattori



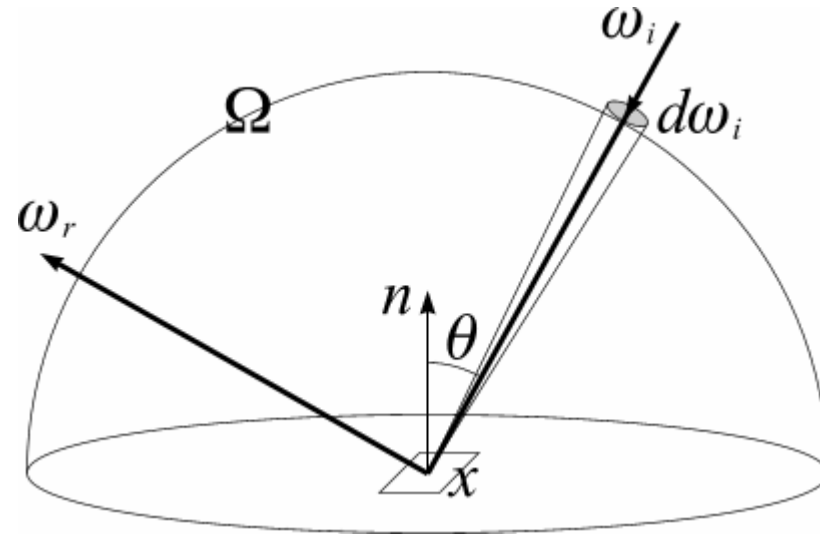
# Lighting: alcuni fattori



# Modelli di illuminazione

- **Modello di illuminazione:** formulazione matematica dell'equazione del trasporto dell'energia luminosa
- L'equazione che risolve questo problema: **equazione di illuminazione**
- **Lighting:** calcolo del bilancio luminoso
- **Shading:** calcolo del colore di ogni pixel dell'immagine

# L'equazione della radianza



$$L_0(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + L_r(x, \Omega, \vec{\omega}_r)$$

$$L_0(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i$$

# L'equazione della radianza

$$L_0(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + L_r(x, \Omega, \vec{\omega}_r)$$

- La luce visibile in un punto della scena è data dalla somma della luce riflessa più la luce emessa

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

- La luce riflessa è un integrale
- Somma i contributi di tutte le sorgenti luminose presenti nella scena e tiene conto dell'angolo di riflessione

# L'equazione della radianza: parametri

$x$  punto sulla superficie in cui si calcola l'equazione;

$\vec{\omega}_r$  direzione che unisce il punto alla posizione dell'osservatore

$\vec{\omega}_i$  direzione da cui proviene il raggio incidente

$f_r$  funzione che determina la frazione riflessa di luce incidente

$\vec{\omega}_i \cdot \vec{n}$  coseno dell'angolo di incidenza rispetto alla normale alla superficie

# L'equazione della radianza

- Calcolo esatto dell'equazione della radianza: operazione complessa e molto costosa
- Sistema di grafica interattiva: formula utilizzabile per tutti i punti della scena più volte al secondo
- Semplificazione dell'equazione



# Lighting: globale VS locale

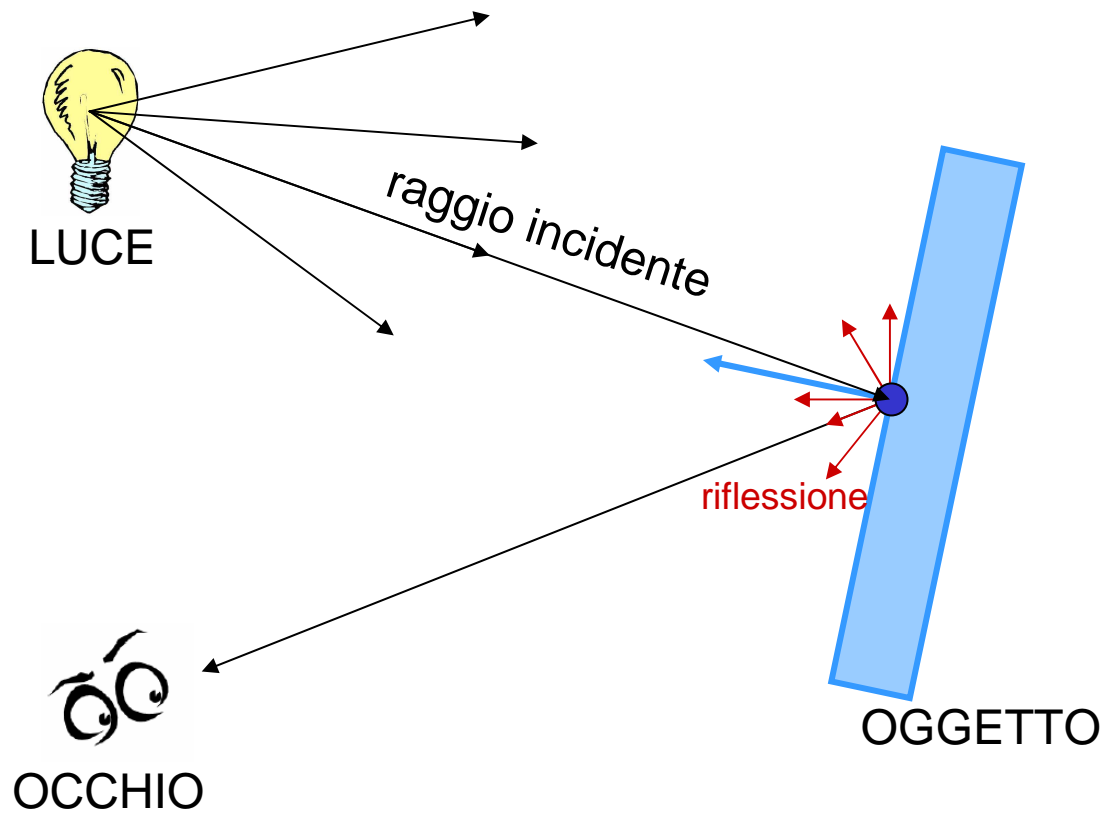
## Illuminazione **locale**

- tiene conto solo di:
  - condizioni di luce
    - N. luci
    - loro pos
    - loro colore
  - pezzetto della superficie da illuminare
    - orientamento (normale)
    - caratteristiche ottiche
      - » per es, colore
- il resto del mondo non c'è

## Illuminazione **globale**

- riflessioni multiple
- ombre
- scattering sottosuperficiale
- rifrazione
- ...

# Lighting locale



# Cosa è facile fare

- Illuminazione **locale**:
  - riflessioni della luce su oggetti
    - con proprietà ottiche molto semplici
  - con multiple fonti di luci
    - ma molto semplici: puntiformi
- Illuminazione **globale**:
  - riflessioni multiple
    - in maniera BRUTALMENTE approssimata
  - assorbimento da parte del mezzo
    - assunzioni semplificanti (nebbia uniforme)
  - tutto il resto solo "a fatica"
    - escogitando algoritmi ad-hoc che si adattano al nostro l'HW

# Il modello di Phong

luce finale

=

ambiente

+

riflessione

+

emissione

per ogni addendo,  
ho una componente R, G e B.

definite sia per l'oggetto,  
(sotto forma di attributi per vertice)  
sia per ogni luce che uso

le proprietà ottiche dell'oggetto,  
(di solito sono attributi per vertice)  
nel loro insieme sono dette  
il suo "materiale"

terminologia OpenGL

## Componente *emissione*

- LEDs, lampadine...
- Non dipende dalle luci
  - solo dall'oggetto
- E' solo una componente additiva
  - costante per R, G e B
- Nota: non manda luce ad oggetti vicini
  - non e' illuminazione globale

# I 3 fattori che consideriamo

luce finale

=

ambiente

+

riflessione

+

emissione

## Componente *ambiente*

- Modella (grossolanamente) la luce che arriva attraverso riflessioni multiple
- Assunzione:  
"un pò di luce raggiunge da tutte le direzioni ogni superficie"  
– anche quelle in ombra
- Piccola **costante** additiva  
– non dipende dalla normale della superficie

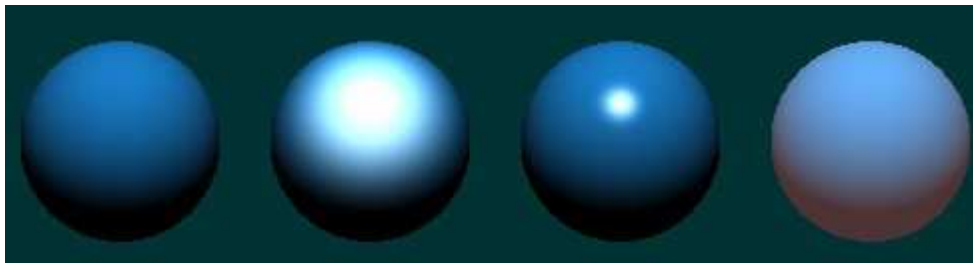
## Componente *ambiente*

- prodotto fra:
  - colore “ambiente” del materiale (  $R_M$   $G_M$   $B_M$  )
  - colore “ambiente” della luce (  $R_L$   $G_L$   $B_L$  )
- Nota: possono essere colori RGB diversi
  - prodotto componente per componente

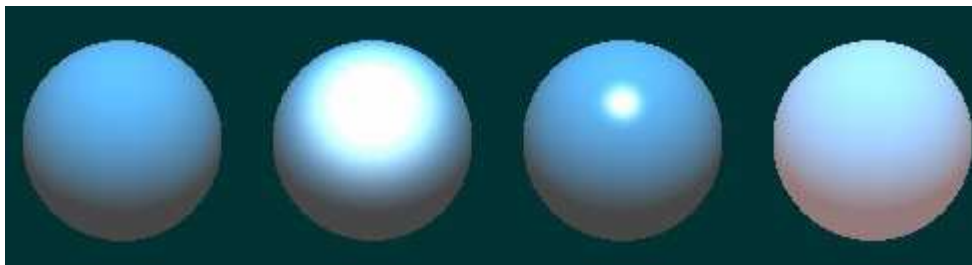


# Componente *ambiente*

- Modella (grossolanamente) la luce che arriva da tutte le direzioni attraverso riflessioni multiple



senza



con

# I 3 fattori che consideriamo

luce finale

=

ambiente

+

riflessione

+

emissione



riflessione diffusa

+

riflessione speculare

# I 4 fattori che consideriamo

luce finale

=

ambiente

+

riflessione diffusa

+

riflessione speculare

+

emissione



# Componente *riflessione diffusa*

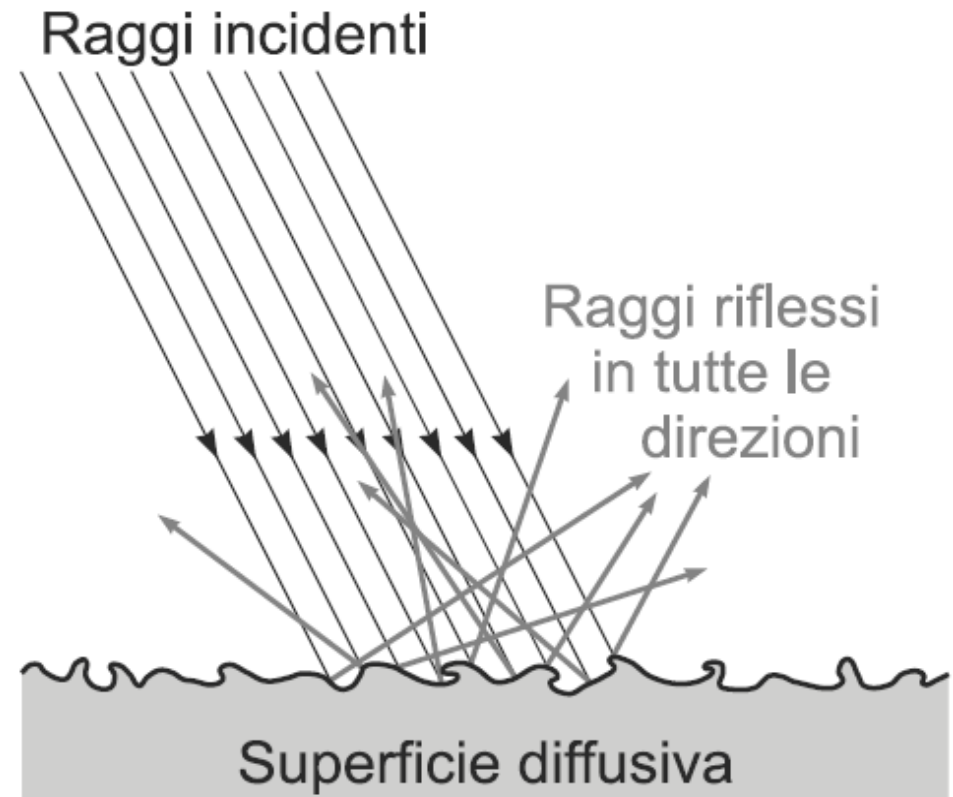
- Esibita nella realtà da (per es):
  - gesso
  - legno (quasi)
  - materiali molto opachi (non lucidi)
- Detta anche
  - diffuse reflection
  - Lambertian reflection



Johann  
Heinrich  
Lambert  
1728 - 1777

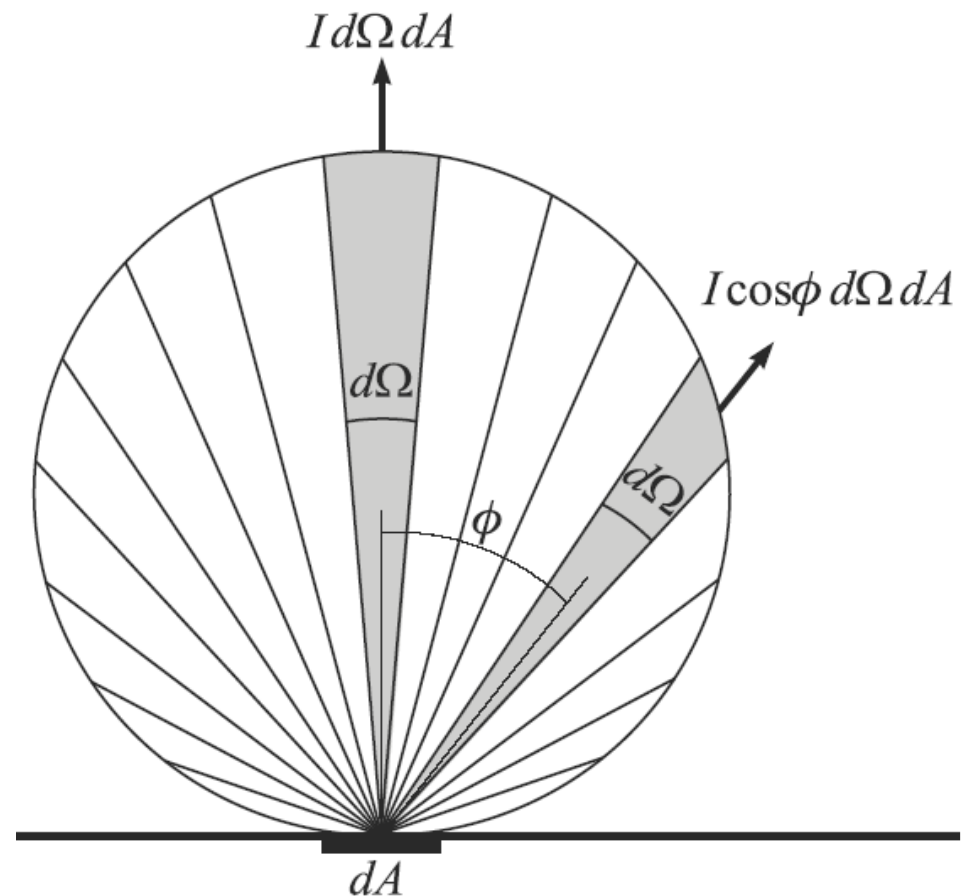
# Legge di Lambert

- 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



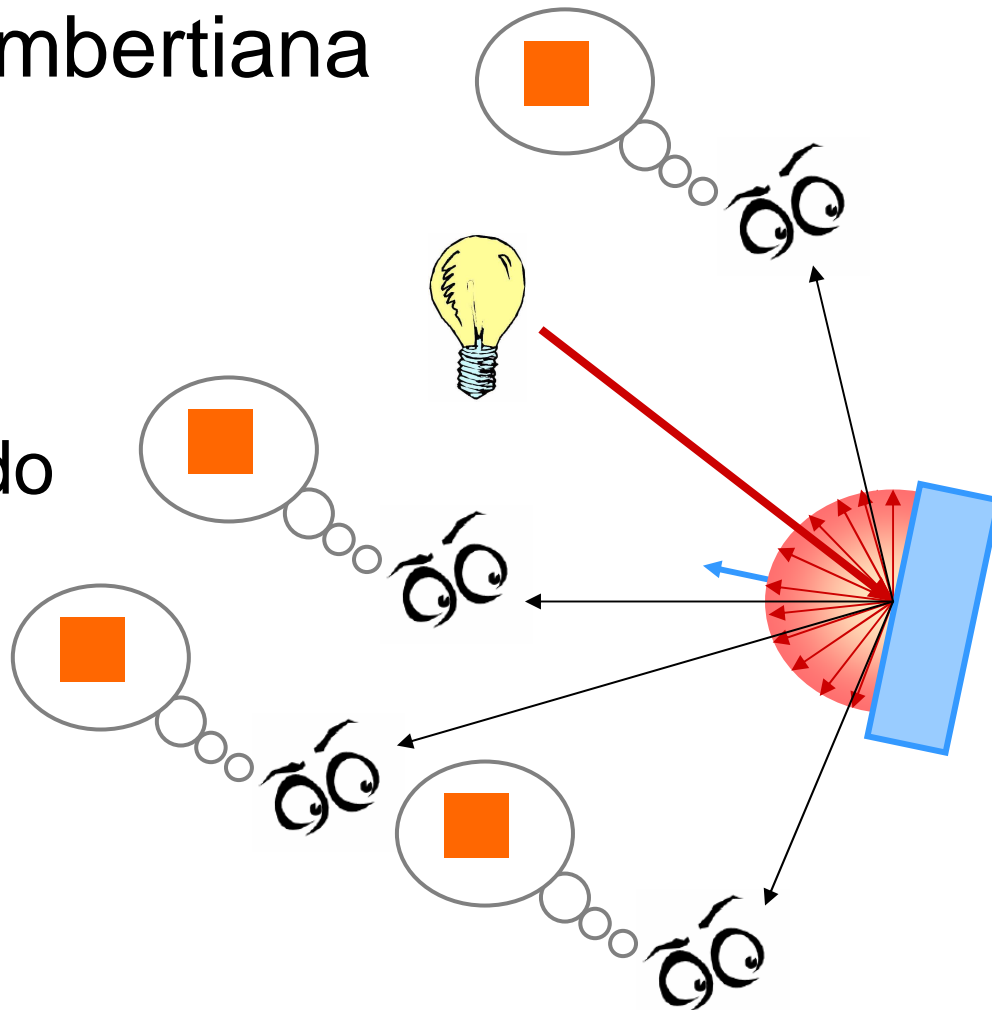
# Legge di Lambert

- 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



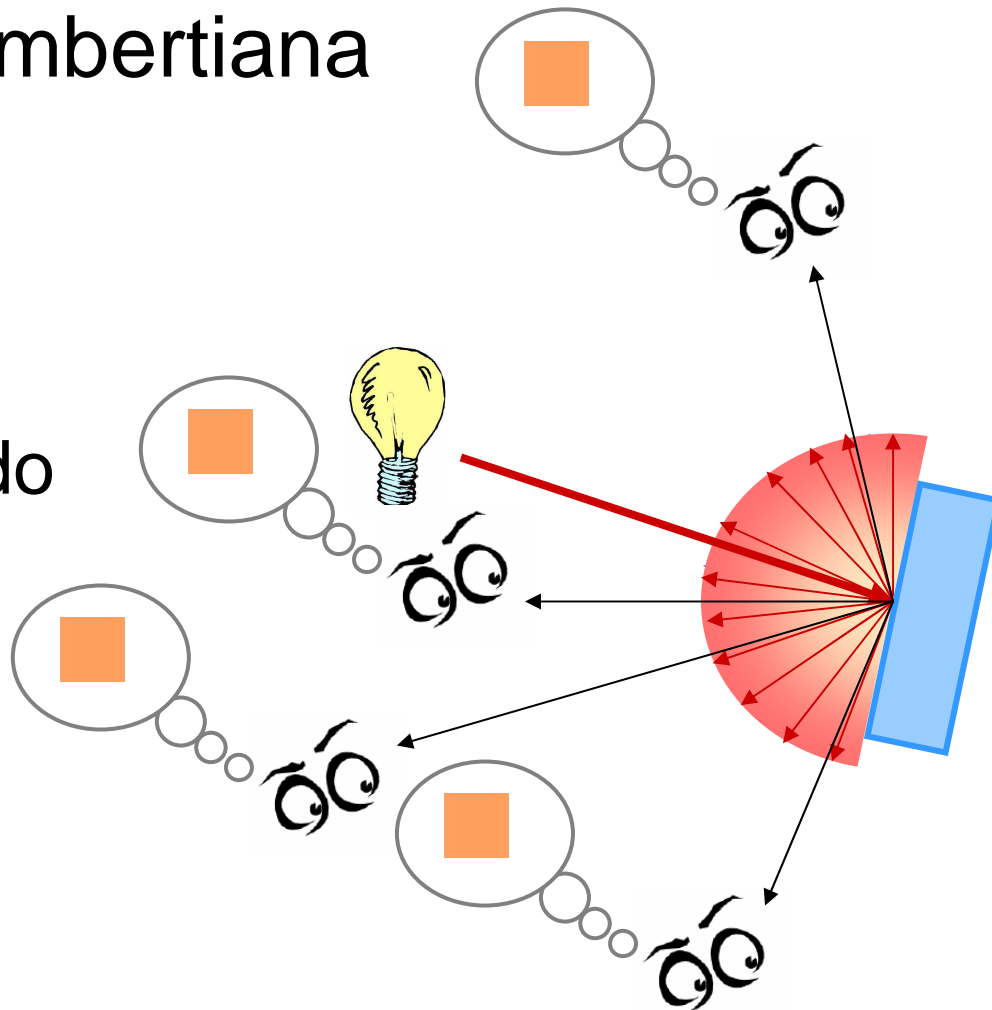
# Componente *riflessione diffusa*

- La luce che colpisce una superficie lambertiana si riflette in tutte le direzioni (nella semisfera)
  - nello stesso modo



# Componente *riflessione diffusa*

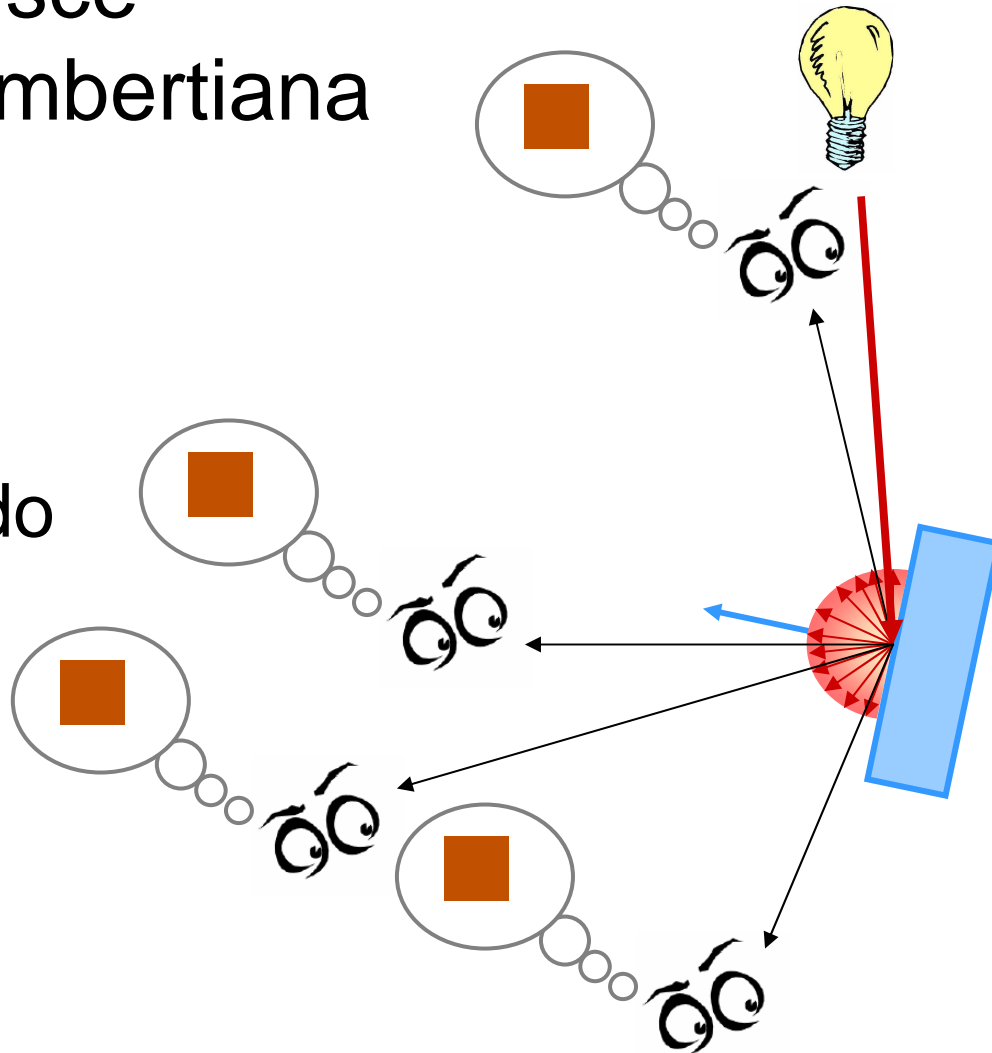
- La luce che colpisce una superficie lambertiana si riflette in tutte le direzioni (nella semisfera)
  - nello stesso modo





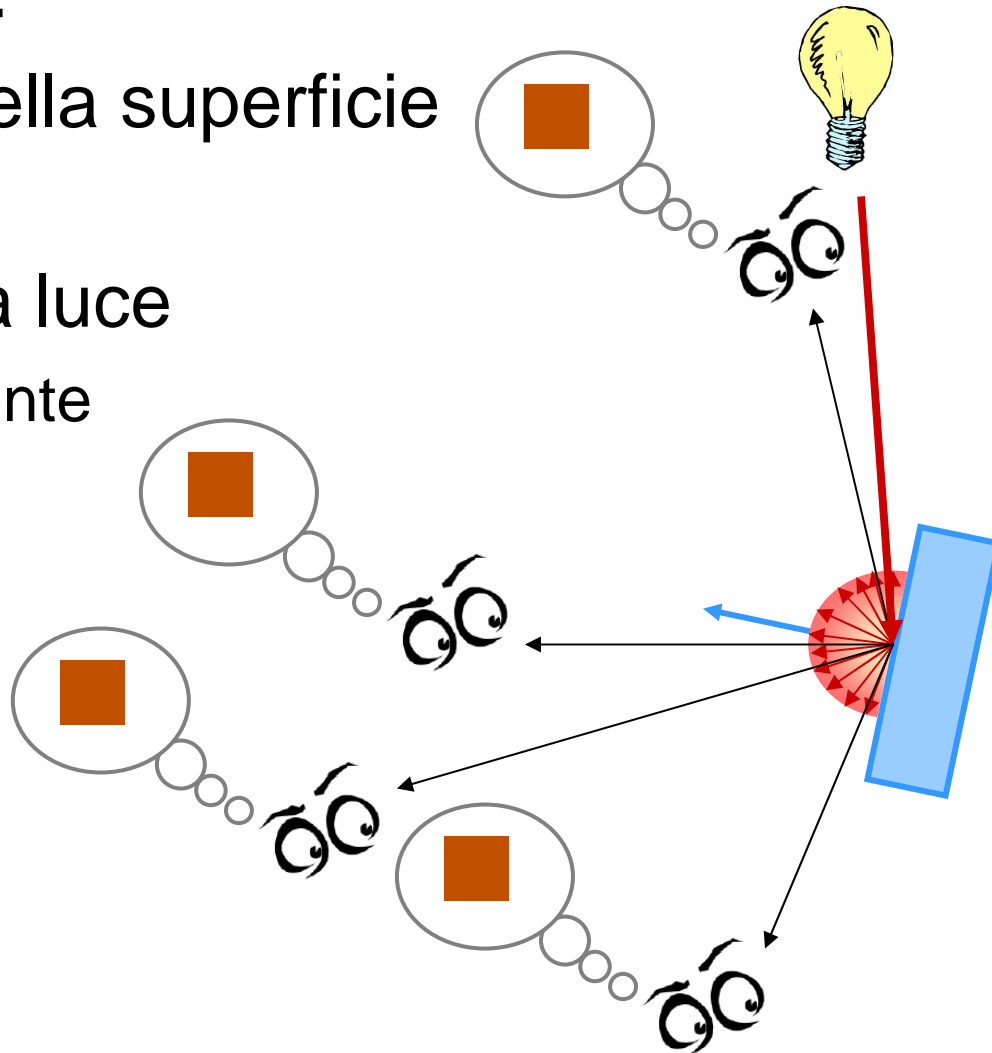
# Componente *riflessione diffusa*

- La luce che colpisce una superficie lambertiana si riflette in tutte le direzioni (nella semisfera)
  - nello stesso modo



# Componente *riflessione diffusa*

- Dipende solo da:
  - l'orientamento della superficie
    - (la "normale")
  - la direzione della luce
    - del raggio incidente



# Componente *riflessione diffusa*

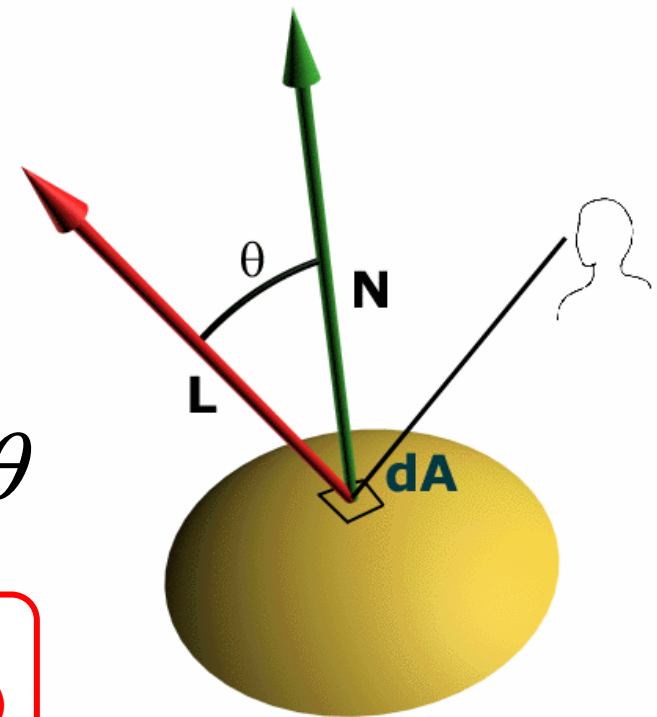
- Dipende solo da:
  - l'orientamento della superficie **N**
    - (cioè la sua "normale")
  - la direzione della luce **L**
    - (cioè del raggio incidente)

$$I_{diff} = I_{luce\ diff} \cdot k_{materiale\ diff} \cdot \cos \theta$$

R, G, B  
(di solito bianco: 1,1,1)

R, G, B  
(il "colore" dell'oggetto)

moltiplicazione componente per componente

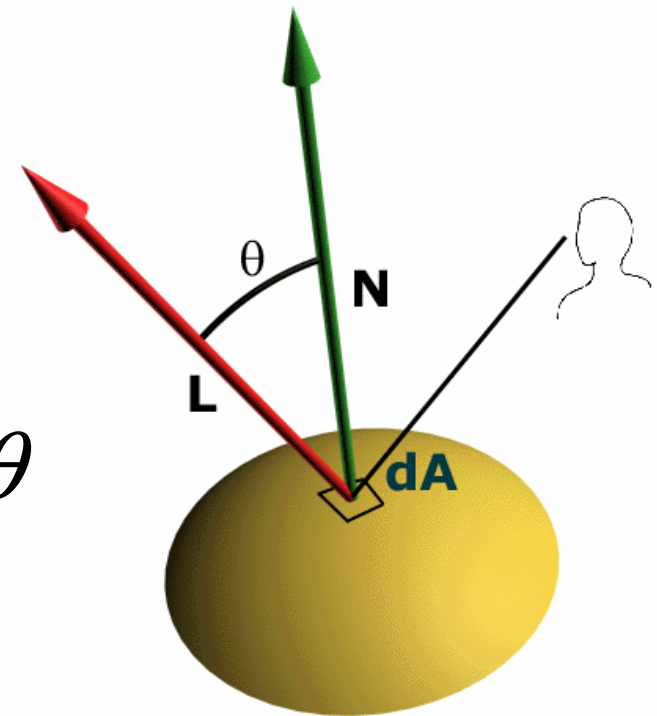


# Componente *riflessione diffusa*

- Dipende solo da:
  - l'orientamento della superficie **N**
    - (cioè la sua "normale")
  - la direzione della luce **L**
    - (cioè del raggio incidente)

$$I_{diff} = I_{luce\ diff} \cdot k_{materiale\ diff} \cdot \cos \theta$$

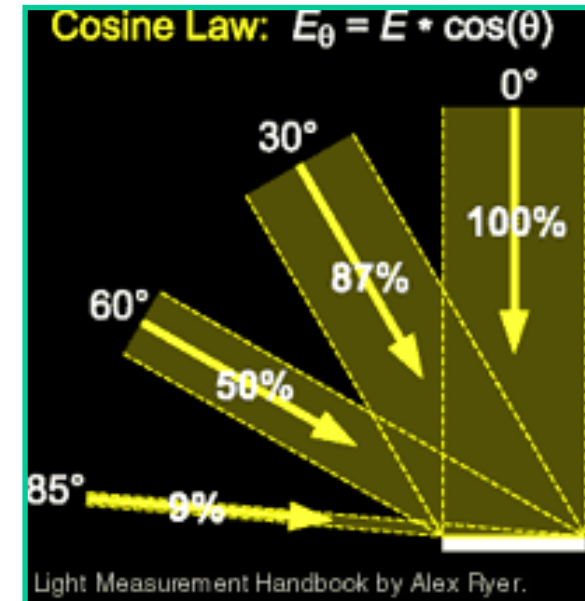
fa parte del  
"materiale"  
(caratteristica  
dell'oggetto)



# Componente *riflessione diffusa*

- Dipende solo da:
  - l'orientamento della superficie **N**
    - (cioè la sua "normale")
  - la direzione della luce **L**
    - (cioè del raggio incidente)

$$I_{diff} = I_{luce\ diff} \cdot k_{materiale\ diff} \cdot \cos \theta$$



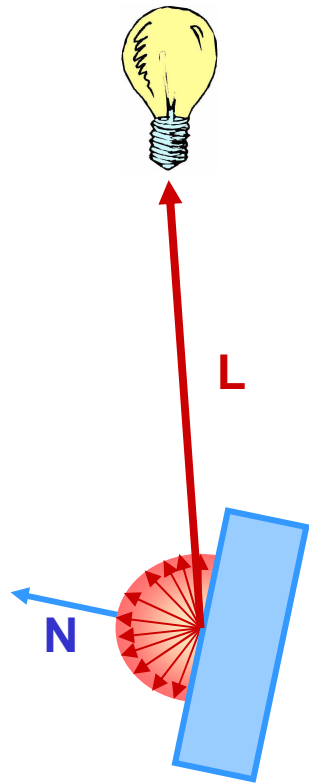
# Componente *riflessione diffusa*

- Dipende solo da:
  - l'orientamento della superficie **N**
    - (cioè la sua "normale")
  - la direzione della luce **L**
    - (cioè del raggio incidente)

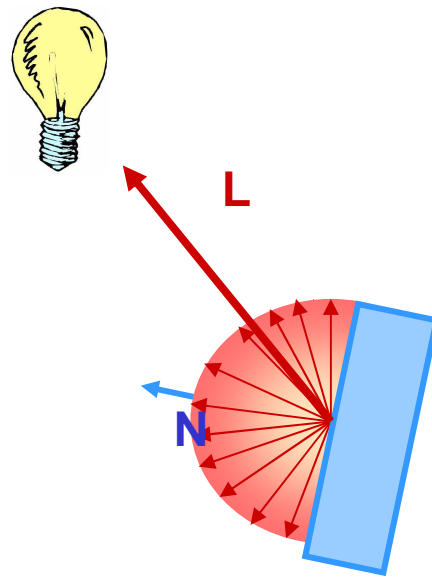
angolo compreso fra  $0^\circ$  e  $90^\circ$ ,  
senno fa 0,  
(oggetto in ombra di se stesso)

$$I_{diff} = I_{luce\ diff} \cdot k_{materiale\ diff} \cdot \cos \theta$$
$$= I_{luce\ diff} \cdot k_{materiale\ diff} \cdot (\hat{N} \cdot \hat{L})$$

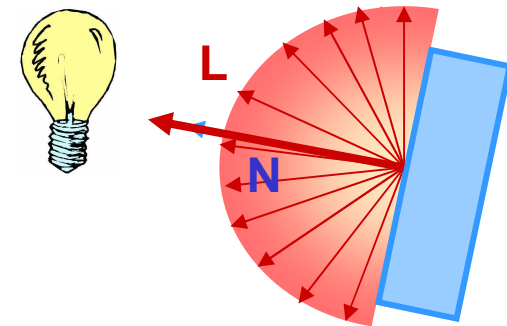
# Componente *riflessione diffusa*



componente  
diffusa  
piccola  
 $\theta = 70^\circ$

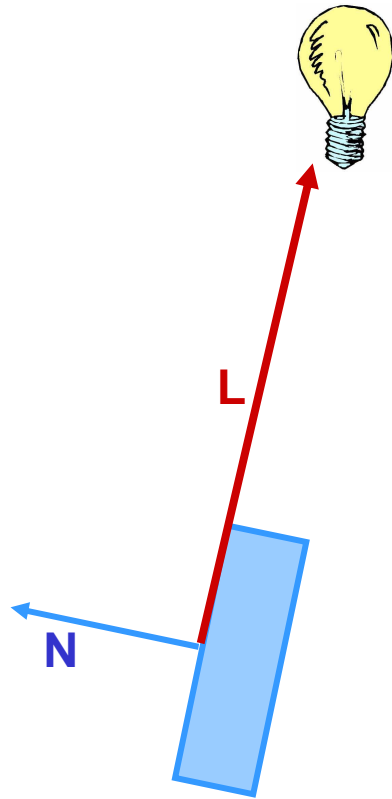


componente  
diffusa  
grande  
 $\theta = 35^\circ$

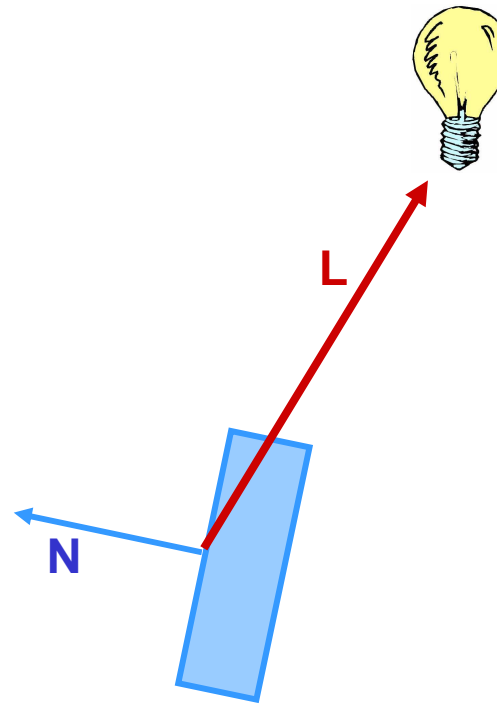


componente  
diffusa  
massima  
 $\theta = 0^\circ$

# Componente *riflessione diffusa*



componente  
diffusa  
ZERO  
 $\theta = 90^\circ$



componente  
diffusa  
ZERO  
 $\theta > 90^\circ$

(la superficie  
è nella propria stessa  
ombra)



# Componente *riflessione diffusa*

- Proprietà
  - modello fedele delle caratteristiche ottiche di alcuni materiali reali 😊
  - ma di pochi materiali 😞
  
  - modello fisicamente coerente 😊
    - per es, conserva l'energia
  - molto semplice da calcolare 😊

# I 4 fattori che consideriamo

luce finale

=

ambiente

+

riflessione diffusa

+

riflessione speculare

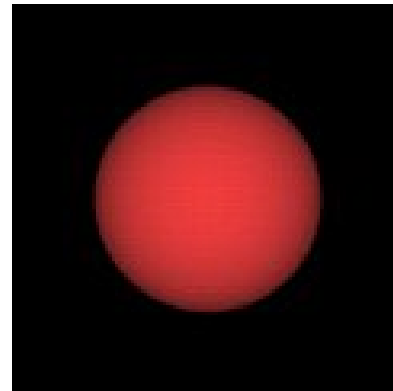
+

emissione

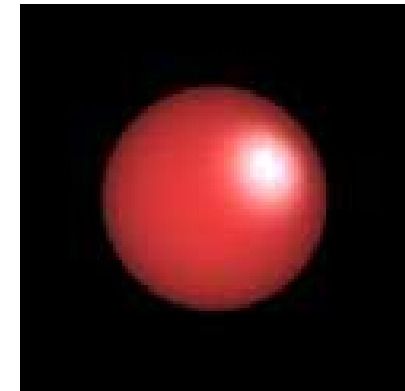


# Componente *riflessione speculare*

- "Specular" reflection
- Per materiali lucidi
  - con riflessi brillanti
  - ("highlights")



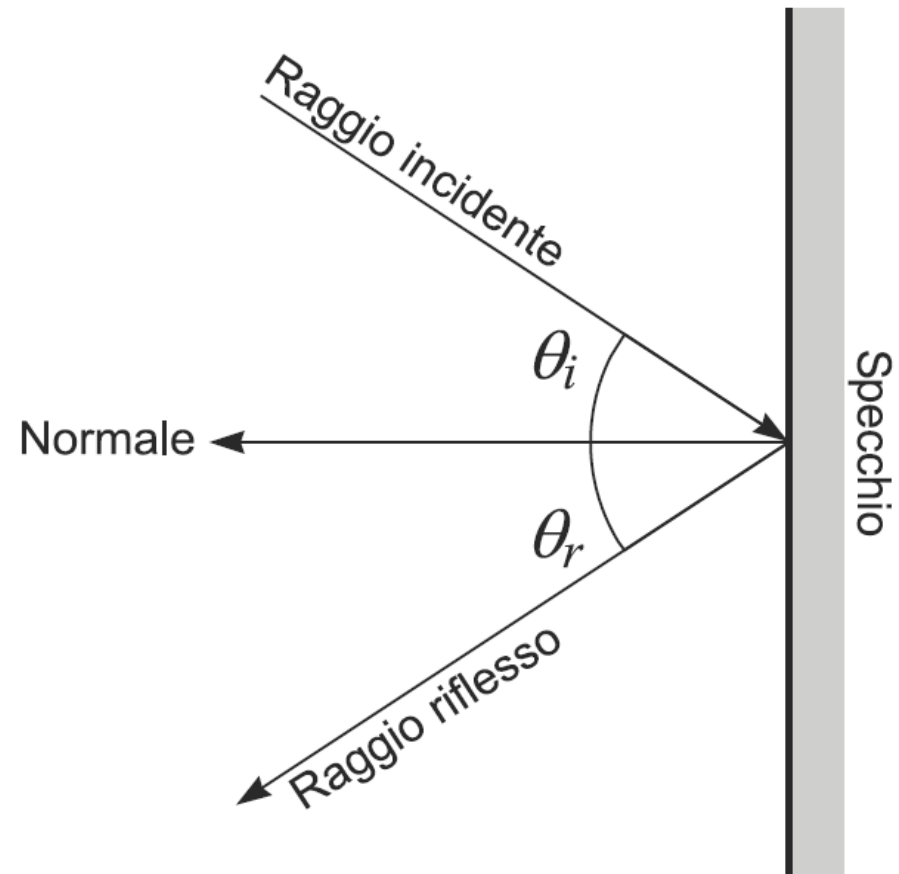
senza



con

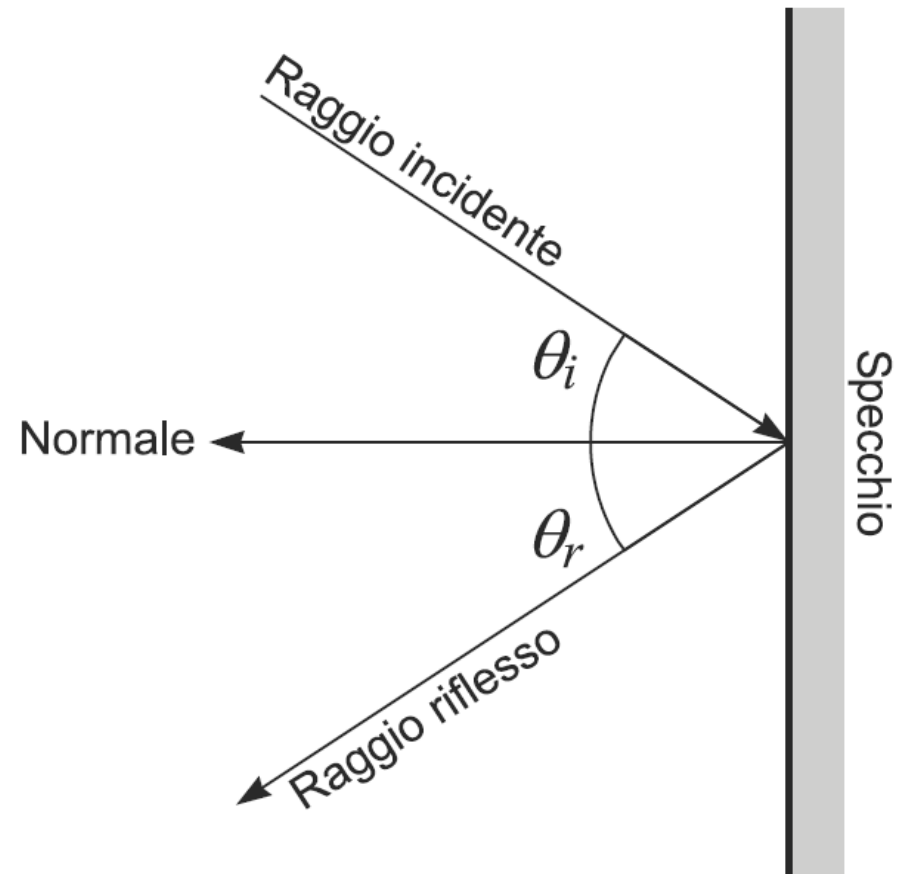
# Legge di Fresnel

- 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



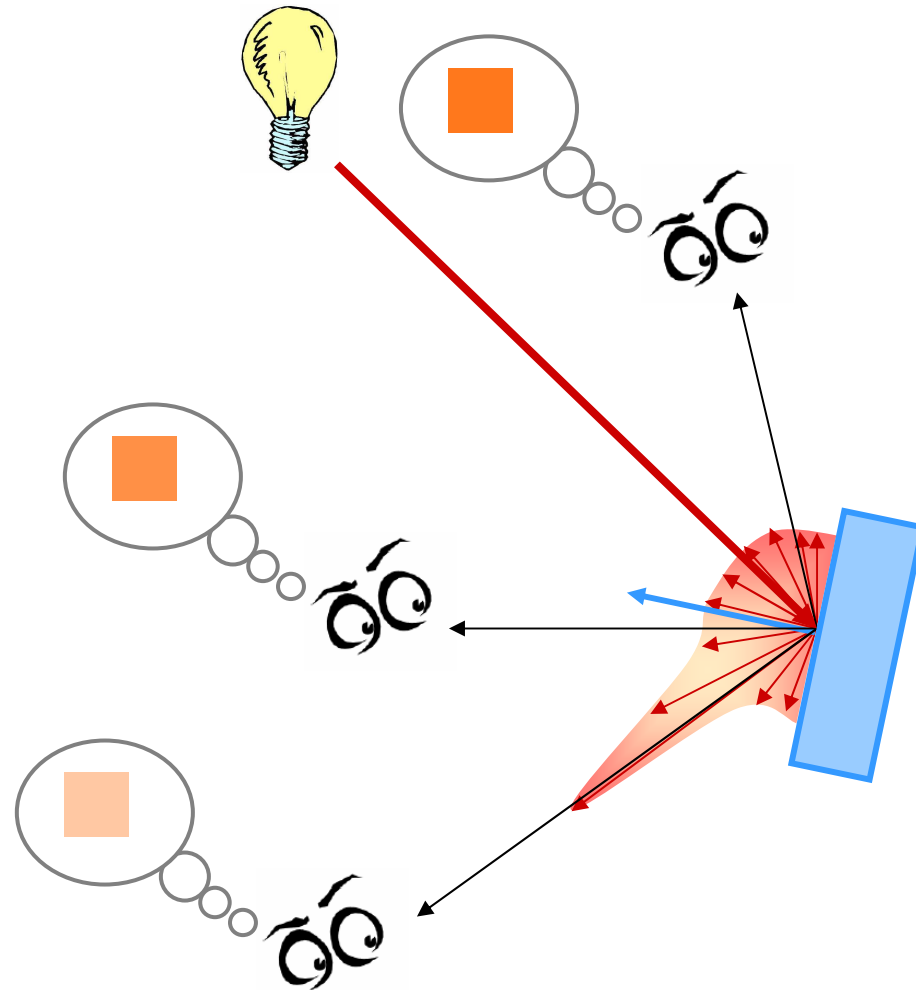
# Legge di Fresnel

- Se da aria a corpo solido non c'è rifrazione si ha solo riflessione
- L'angolo di incidenza è uguale all'angolo di riflessione
- Vale per materiali molto lisci e lucidi



# Componente *riflessione speculare*

- Idea base:  
la luce **non** viene riflessa  
da materiali **lucidi**  
in maniera eguale  
in tutte le direzioni



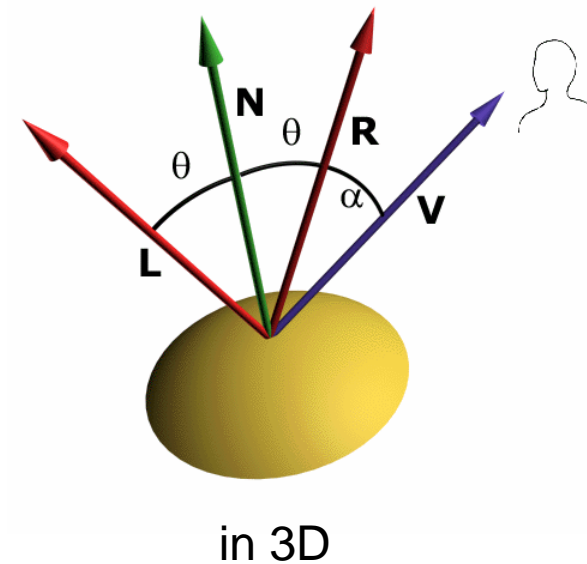
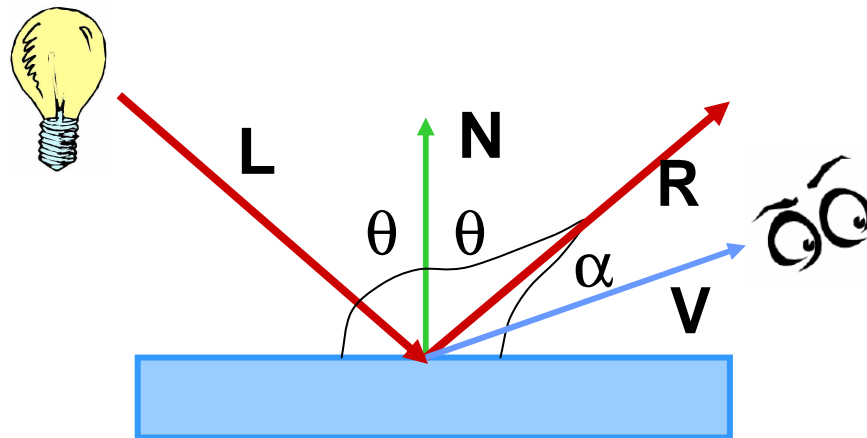
# Componente *riflessione speculare*

L: raggio incidente

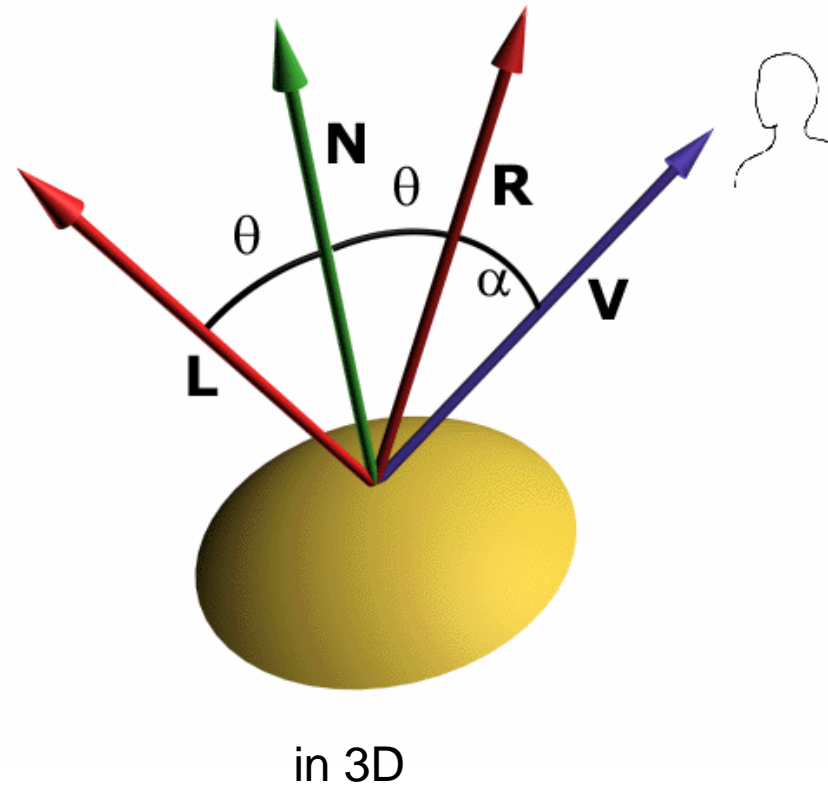
N: normale

R: raggio riflesso

V: dir. di vista



# Componente *riflessione speculare*

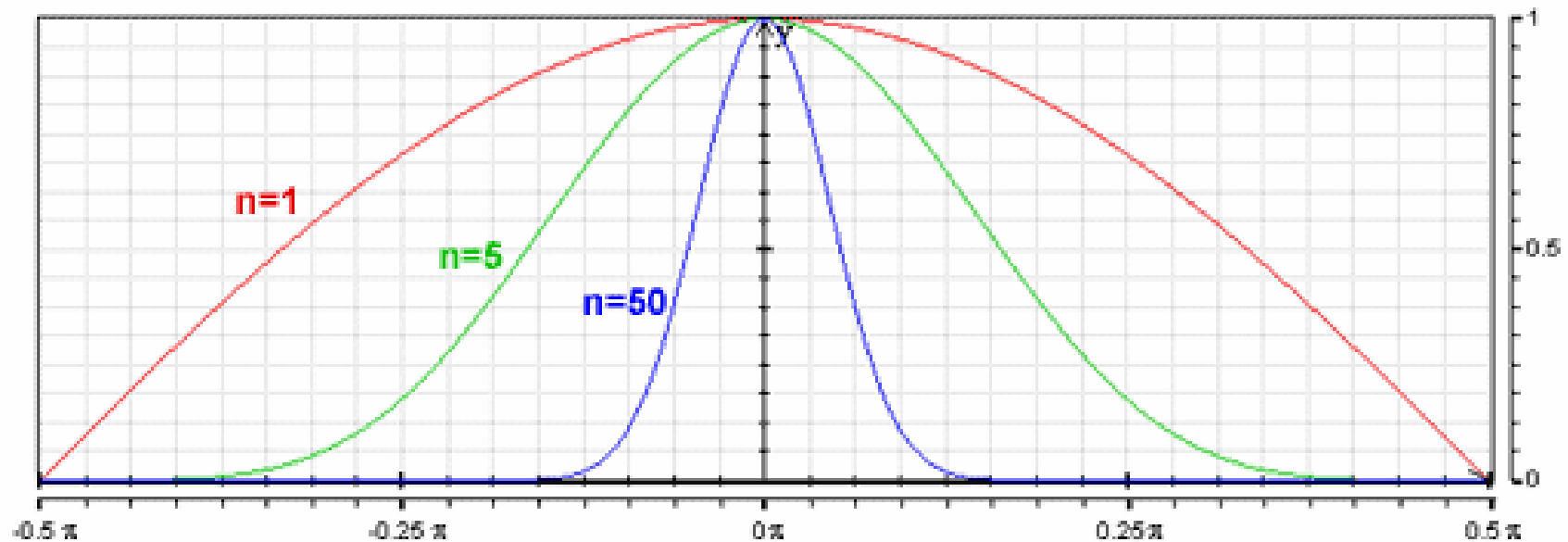


$$I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot \cos \alpha$$



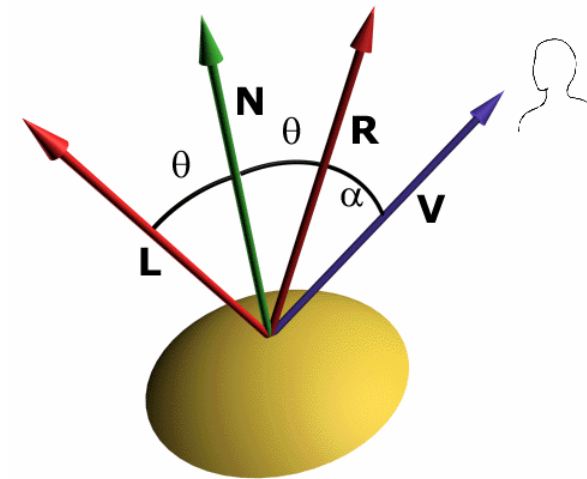
# Componente *riflessione speculare*

- Elevando il coseno ad una potenza, si ottengono riflessi piu' piccoli e brillanti



# Componente *riflessione speculare*

- Phong light model
  - by Bui-Tuong Phong, 1975



in 3D

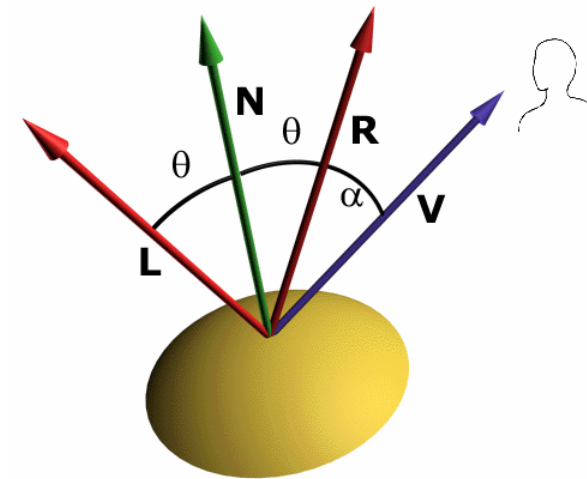
$$I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot \cos \alpha$$

$$I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot \cos^n \alpha$$

fanno parte del "materiale"  
(caratteristiche dell'oggetto)

# Componente *riflessione speculare*

- Phong light model
  - by Bui-Tuong Phong, 1975

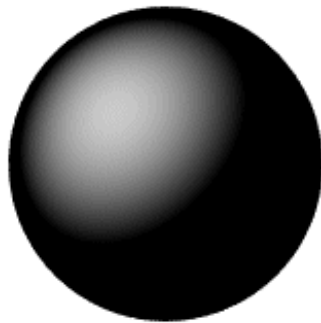


$$I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot \cos \alpha$$

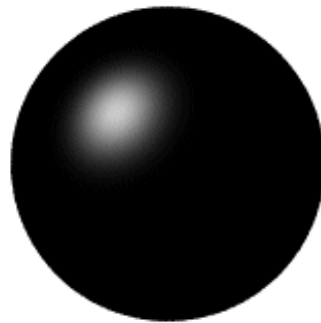
$$I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot \cos^n \alpha$$

$$= I_{luce\ spec} \cdot k_{materiale\ spec} \cdot (\hat{R} \cdot \hat{V})^n$$

# Componente *riflessione speculare*



$n = 1$



$n = 5$



$n = 10$



$n = 100$

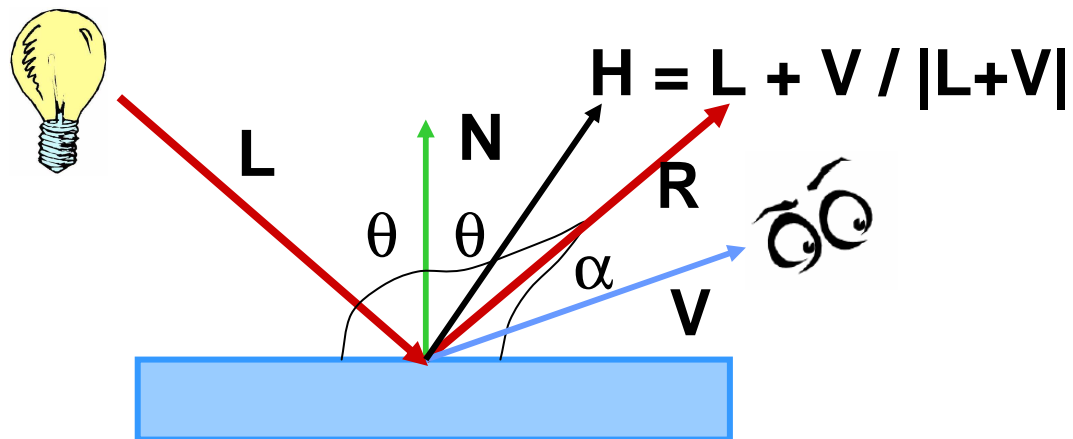
# Componente *riflessione speculare*

- Blinn-Phong light model:
  - semplificazione del Phong light model
  - risultati simili, formula diversa:

phong:  $I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot (\hat{R} \cdot \hat{V})^n$

blinn-phong:  $I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot (\hat{H} \cdot \hat{N})^n$

"half-way" vector



## Componente *riflessione speculare*

- Blinn-Phong light model:
  - semplificazione del Phong light model
  - risultati simili, formula diversa:

$$\text{phong: } I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot (\hat{R} \cdot \hat{V})^n$$

$$\text{blinn-phong: } I_{spec} = I_{luce\ spec} \cdot k_{materiale\ spec} \cdot (\hat{H} \cdot \hat{N})^n$$

# I 4 fattori che consideriamo

luce finale

=

ambiente

+

riflessione diffusa

+

riflessione speculare

+

emissione



# Equazione di lighting in totale

$$I_{tot} = I_{luce\ ambient} \cdot k_{materiale\ ambient} + I_{luce\ diffuse} \cdot k_{materiale\ diffuse} \cdot (L \cdot N) + I_{luce\ specular} \cdot k_{materiale\ specular} \cdot (H \cdot N)^n + k_{materiale\ emission}$$

proprietà del materiale

proprietà della luce



# Materiali...

Material	GL_AMBIENT	GL_DIFFUSE	GL_SPECULAR	GL_SHININESS
Emerald	0.0215 0.1745 0.0215 0.55	0.07568 0.61424 0.07568 0.55	0.633 0.727811 0.633 0.55	76.8
Jade	0.135 0.2225 0.1575 0.95	0.54 0.89 0.63 0.95	0.316228 0.316228 0.316228 0.95	12.8
Obsidian	0.05375 0.05 0.06625 0.82	0.18275 0.17 0.22525 0.82	0.332741 0.328634 0.346435 0.82	38.4
Pearl	0.25 0.20725 0.20725 0.922	1.0 0.829 0.829 0.922	0.296648 0.296648 0.296648 0.922	11.264
Ruby	0.1745 0.01175 0.01175 0.55	0.61424 0.04136 0.04136 0.55	0.727811 0.626959 0.626959 0.55	76.8
Turquoise	0.1 0.18725 0.1745 0.8	0.396 0.74151 0.69102 0.8	0.297254 0.30829 0.306678 0.8	12.8
Black Plastic	0.0 0.0 0.0 1.0	0.01 0.01 0.01 1.0	0.50 0.50 0.50 1.0	32
Black Rubber	0.02 0.02 0.02 1.0	0.01 0.01 0.01 1.0	0.4 0.4 0.4 1.0	10
Brass	0.329412 0.223529 0.027451 1.0	0.780392 0.568627 0.113725 1.0	0.992157 0.941176 0.807843 1.0	27.8974
Bronze	0.2125 0.1275 0.054 1.0	0.714 0.4284 0.18144 1.0	0.393548 0.271906 0.166721 1.0	25.6
Polished Bronze	0.25 0.148 0.06475 1.0	0.4 0.2368 0.1036 1.0	0.774597 0.468561 0.200621 1.0	76.8
Chrome	0.25 0.25 0.25 1.0	0.4 0.4 0.4 1.0	0.774597 0.774597 0.774597 1.0	76.8



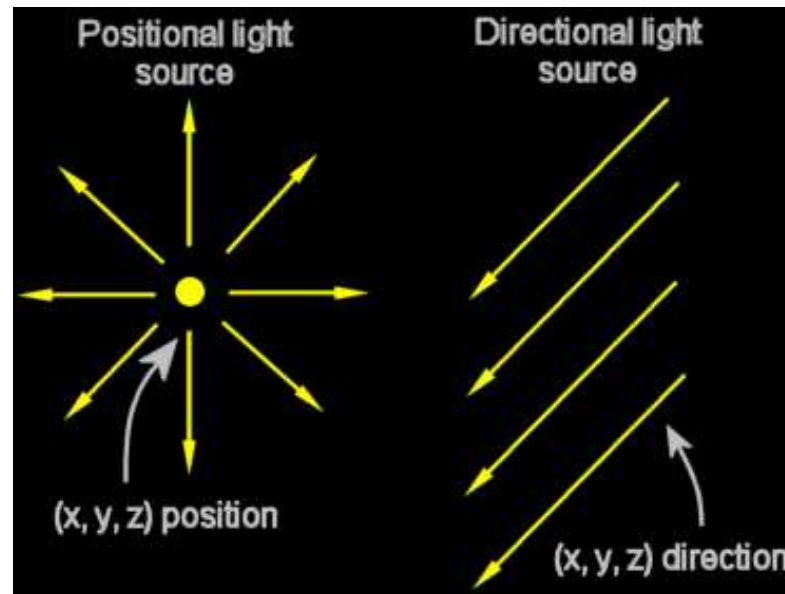
# Equazione di lighting: modellazione delle luci

$$I_{tot} = I_{luce\ ambient} \cdot k_{materiale\ ambient} +$$
$$I_{luce\ diffuse} \cdot k_{materiale\ diffuse} \cdot (\hat{L} \cdot \hat{N}) +$$
$$I_{luce\ specular} \cdot k_{materiale\ specular} \cdot (\hat{H} \cdot \hat{N})^n +$$
$$k_{materiale\ emission} \frac{(\hat{L} + \hat{V})}{|\hat{L} + \hat{V}|}$$

proprietà della luce

# Modellazione delle luci

- Come varia L?
  - costante nella scena: fonti di luci "direzionali"
    - buono per fonti di luce molto distanti, e.g. il sole
  - varia nella scena: fonti di luci "posizionali"
    - buono per fonti di luci vicine, e.g. lampadine



# Modellazione delle luci: luci posizionali

- Nelle luci posizionali, si può attenuare l'intensità in funzione della distanza
- In teoria (per la fisica) intensità = 1 / distanza<sup>2</sup>

$$f_{\text{attenuazione luce}} = \left( \frac{1}{c \cdot d_L^2} \right)$$

# Modellazione delle luci: luci posizionali

- In pratica, questo porta ad attenuazioni della luce troppo repentine
- Invece usiamo:

$$f_{\text{attenuazione luce}} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$$

# Equazione di lighting

$$I_{tot} = \left( \begin{array}{l} I_{luce\ ambient} \cdot k_{materiale\ ambient} + \\ I_{luce\ diffuse} \cdot k_{materiale\ diffuse} \cdot (L \cdot N) + \\ I_{luce\ specular} \cdot k_{materiale\ specular} \cdot (H \cdot N)^n \end{array} \right) \cdot f_{attenuazione\ luce} + k_{materiale\ emission}$$

$$f_{attenuazione\ luce} = \min \left( \frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right)$$

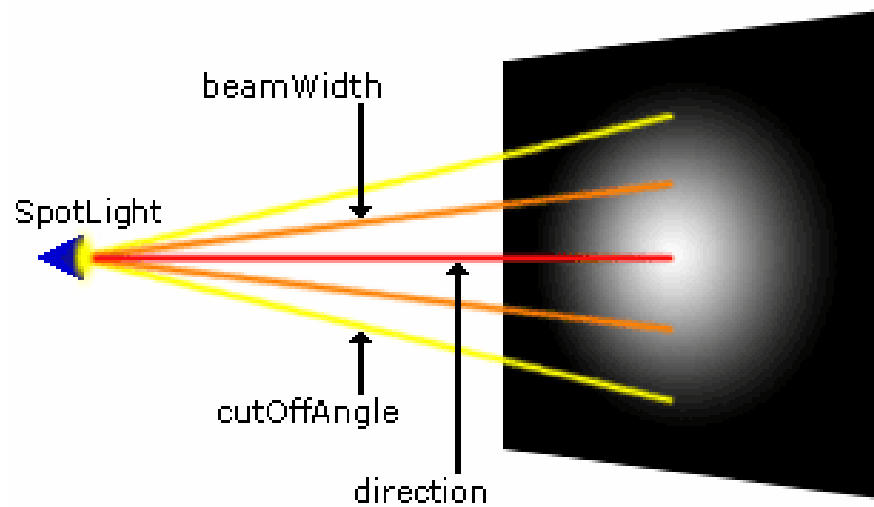
# Tipi di luci

- Tipi di luci:
  - posizionali
  - direzionali
  - spot-lights
    - (faretti)



# Spotlights

- Definite da tre parametri:





# Equazione di lighting

$$I_{tot} = \left( \begin{array}{l} I_{luce\ ambient} \cdot k_{materiale\ ambient} + \\ I_{luce\ diffuse} \cdot k_{materiale\ diffuse} \cdot (L \cdot N) + \\ I_{luce\ specular} \cdot k_{materiale\ specular} \cdot (H \cdot N)^n \end{array} \right) \cdot f_{attenuazione\ luce} \cdot f_{effetto\ spotlight} + k_{materiale\ emission}$$

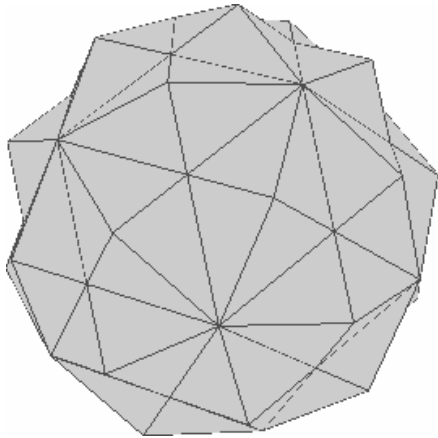
$$f_{attenuazione\ luce} = \min \left( \frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right)$$

$$f_{effetto\ spotlight} = f(L, spot_{direction}, spot_{cutoff\ Angle}, spot_{beam\ width})$$

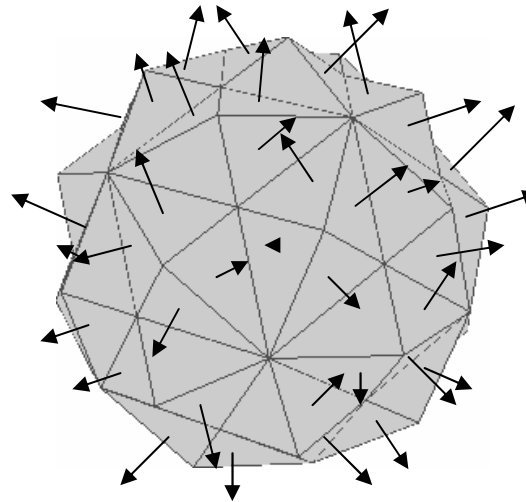
# Tecniche di shading

- Il modello di Phong descrive *come* deve essere calcolata l'interazione tra luce e materia
- Dobbiamo capire *dove* calcolare l'equazione d'illuminazione
- Sistema interattivo  $\Rightarrow$  generazione di un certo numero di frame per secondo  $\Rightarrow$  metodi approssimati

# Shading costante

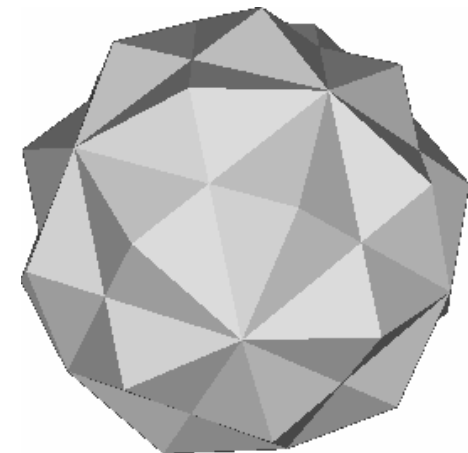


Dato l'oggetto per cui calcolare l'equazione di illuminazione  $I$  ...



...calcolare le normali in ogni faccia...

...e calcolo  $I$  una sola volta per faccia



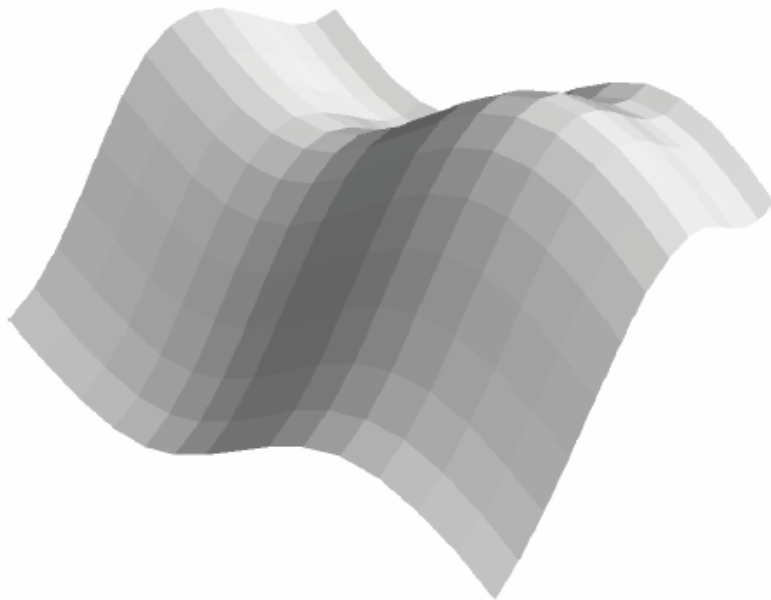
# Shading costante

- Se:
  - sorgenti luminose solo direzionali ( $N \cdot L = k$  per tutta la superficie)
  - osservatore a distanza infinita dalla scena (proiezione parallela)  $\Rightarrow N \cdot V = k$  e  $R \cdot V = k$  per tutta la superficie

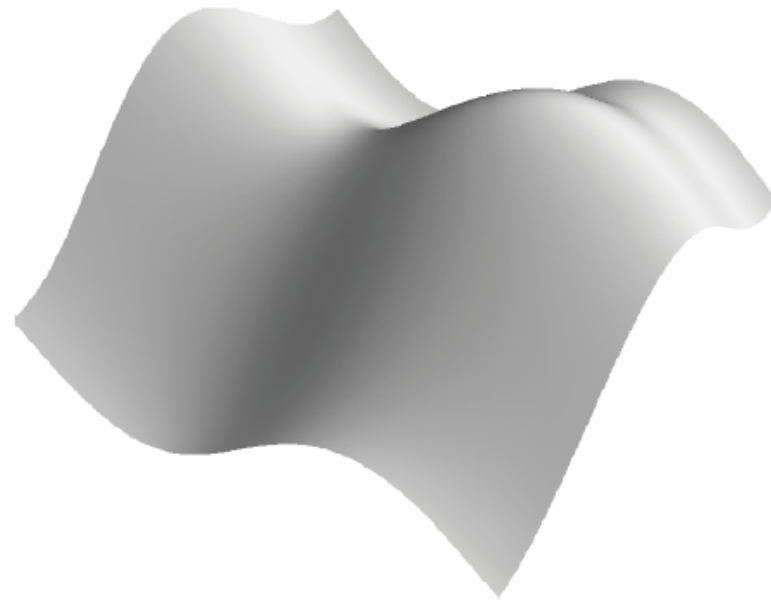
il metodo è la migliore approssimazione possibile

# Shading costante

- Problema: il modello discreto rappresenta in modo approssimato una superficie curva e continua



Com'è



Come dovrebbe essere

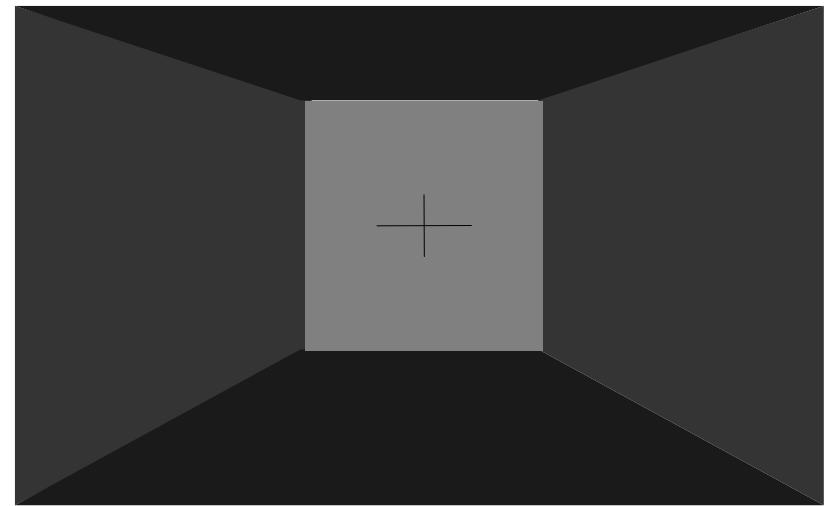
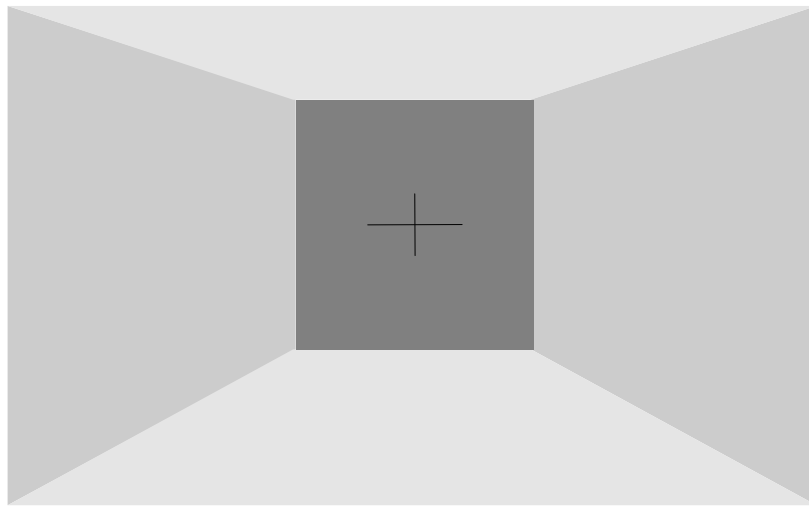
# Shading costante

- Soluzione: uso un numero elevato di facce
- Non funziona, si vedono comunque le discontinuità tra una faccia e la vicina



# Mach banding

- Alterazione della percezione visiva di una zona in cui la luminanza varia rapidamente
- Un oggetto messo vicino ad uno più chiaro risulta più scuro e messo vicino ad uno più scuro risulta più chiaro



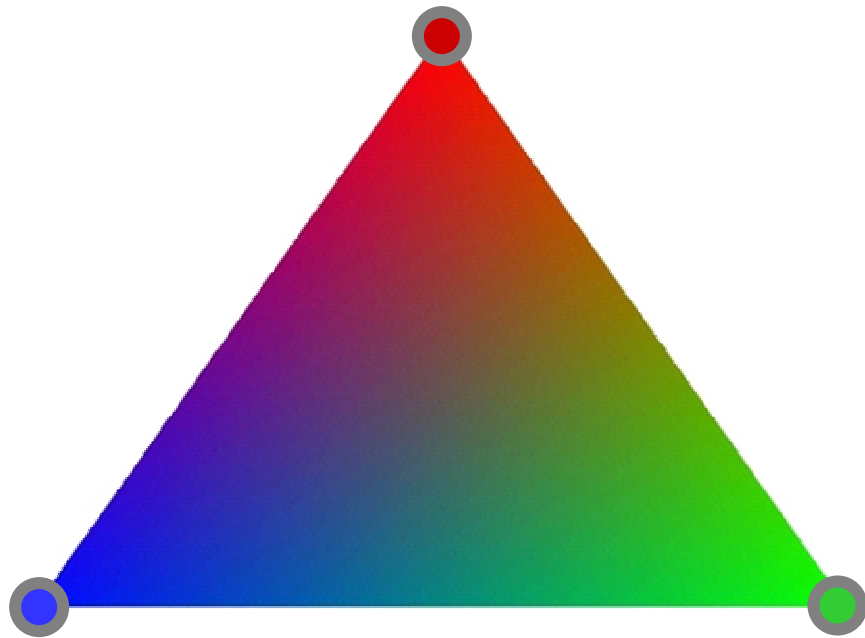
# Gouraud shading

- Proprietà fondamentale dello spazio colore RGB: linearità
- Il valore colore intermedio tra due colori dati nello spazio RGB si calcola per interpolazione lineare
- Interpolazione separata sulle tre componenti R, G, e B



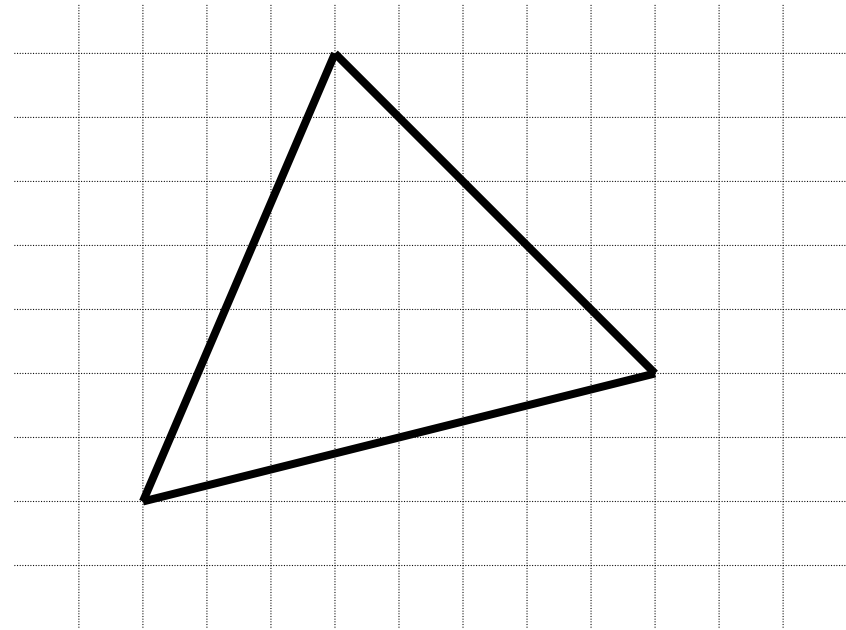
# Gouraud shading

- Calcolare l'equazione di illuminazione solo in alcuni punti nodali
- Interpolare linearmente tra questi valori



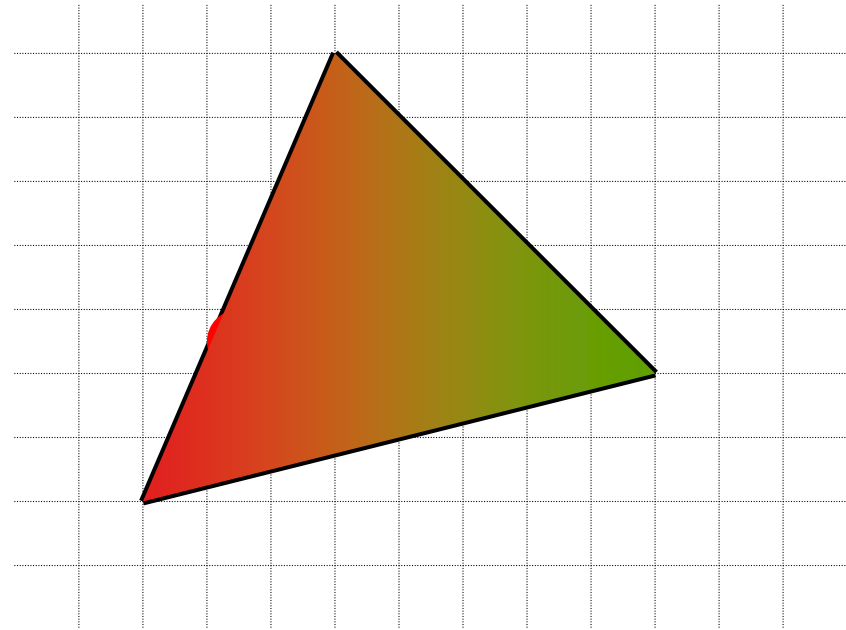
# Gouraud shading

- Aggiungere all'algoritmo di rasterizzazione l'operazione di interpolazione nello spazio colore comporta uno sforzo minimo



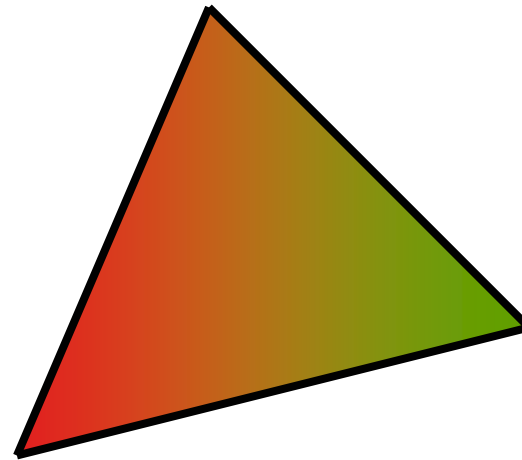
# Gouraud shading

- Per ogni span si calcola il valore di  $I$  all'estremo con un algoritmo incrementale, e, sempre incrementalmente, si calcolano i valori all'interno della span



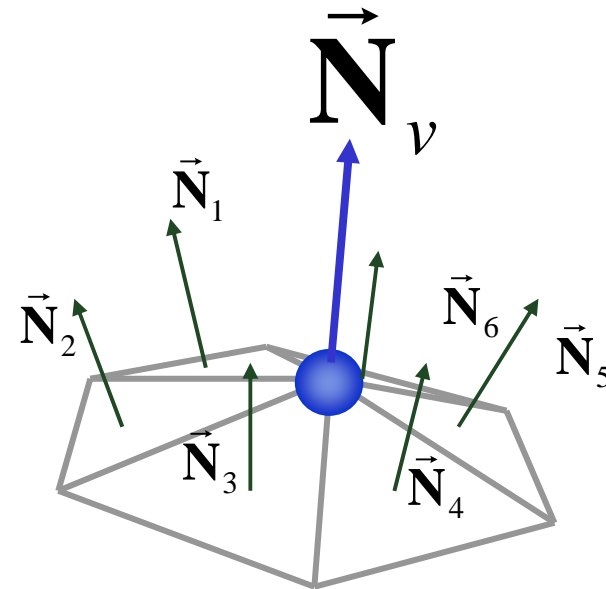
# Gouraud shading

- Il risultato così ottenuto approssima molto il modello di Phong per superfici generiche rispetto allo shading costante



# Gouraud shading

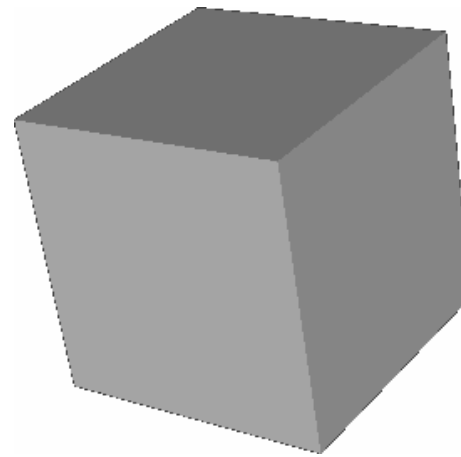
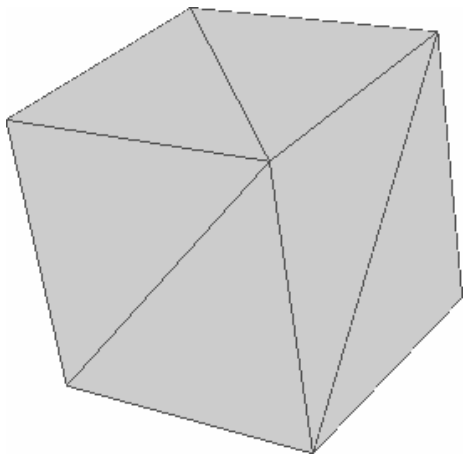
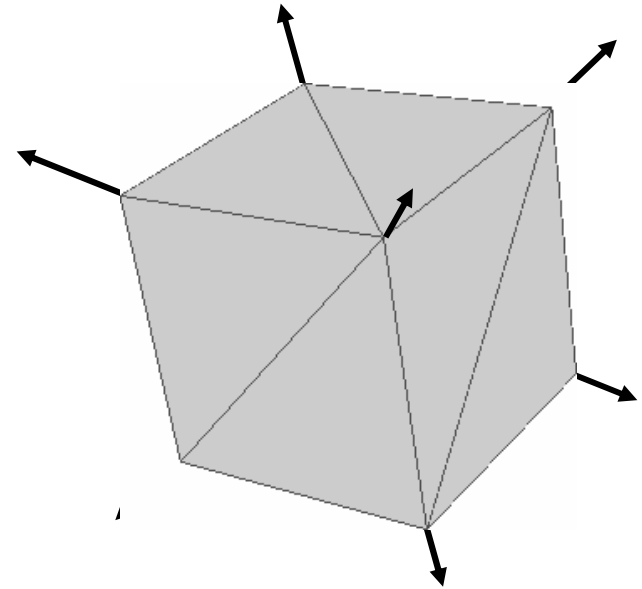
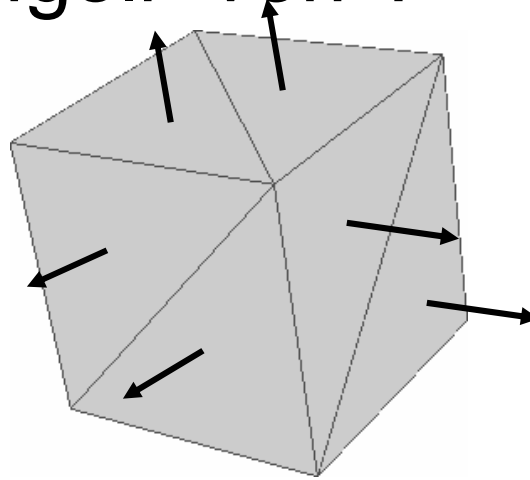
- Che normali utilizzo?
- La normale alla faccia è bene definita
- La normale al vertice la calcolo come media delle normali delle facce che insistono sul vertice



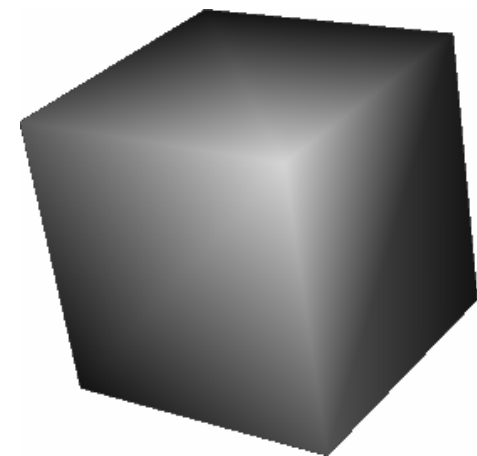
$$\vec{N}_v = \frac{\sum_i \vec{N}_i}{|\sum_i \vec{N}_i|}$$

# Gouraud shading

- Problema: gli spigoli “veri”?



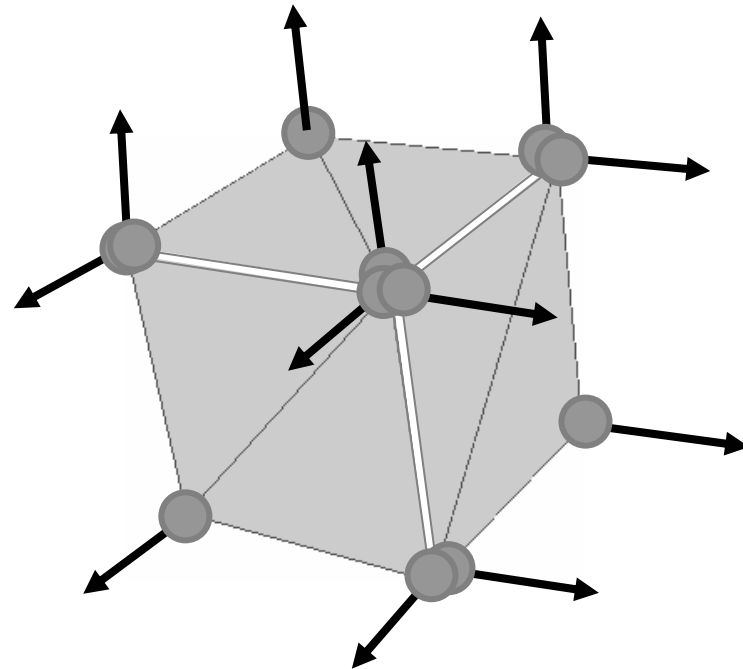
shading costante



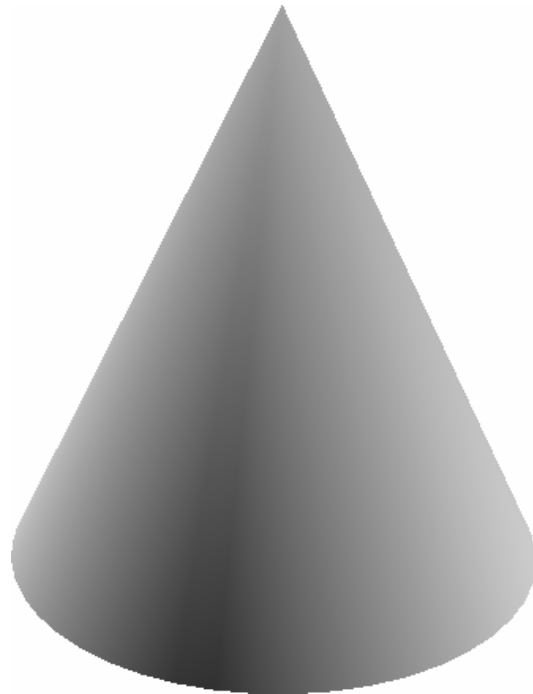
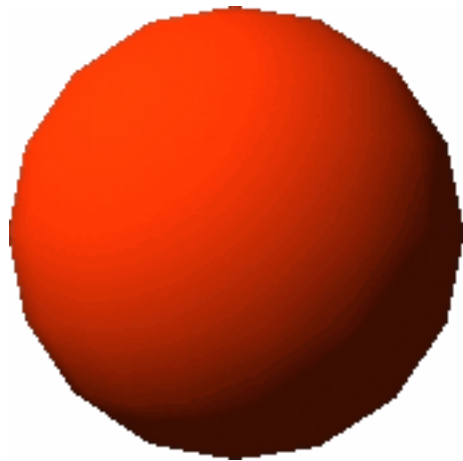
Gouraud shading

# Gouraud shading

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



# Paragone: costante e Gouraud



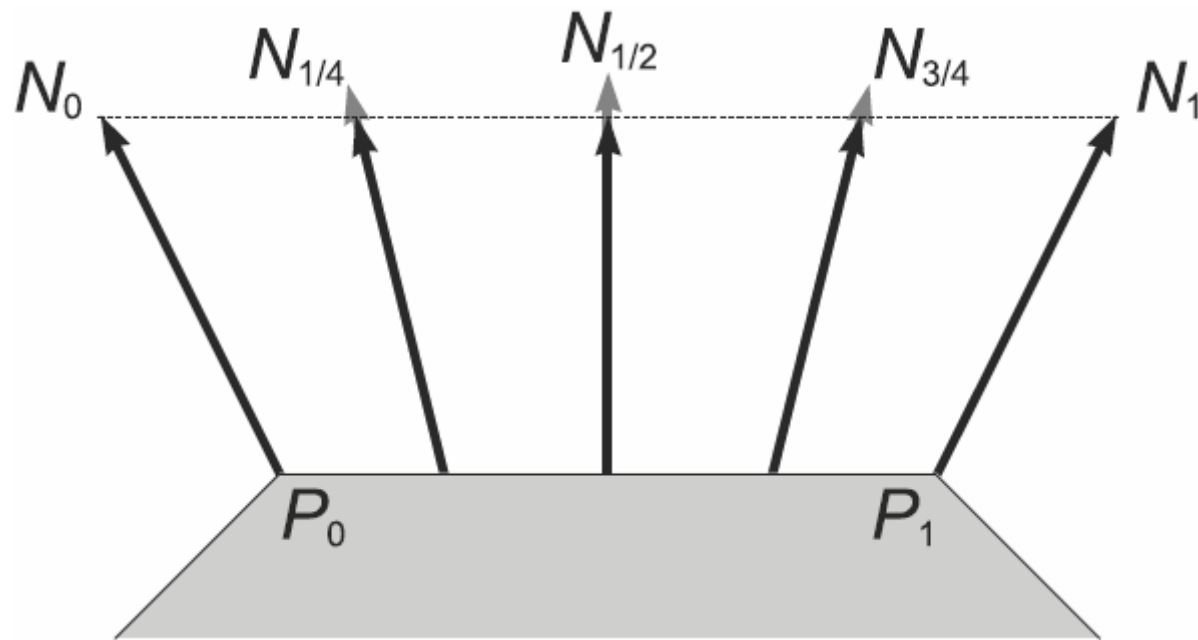


# Phong shading

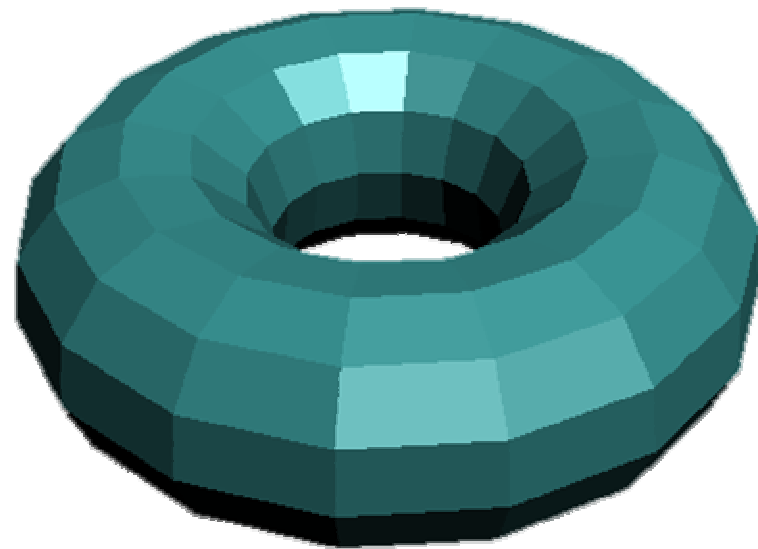
- Gouraud shading: ottimale rapporto qualità/prezzo
- Risultati non eccezionali per superfici dotate di un alto coefficiente di riflessione speculare
- Problema: per  $n$  alto lo *specular highlight* deve essere piccolo, invece si “propaga” per tutta la faccia (per interpolazione) se cade vicino a un vertice, si “perde” se è interno

# Phong shading

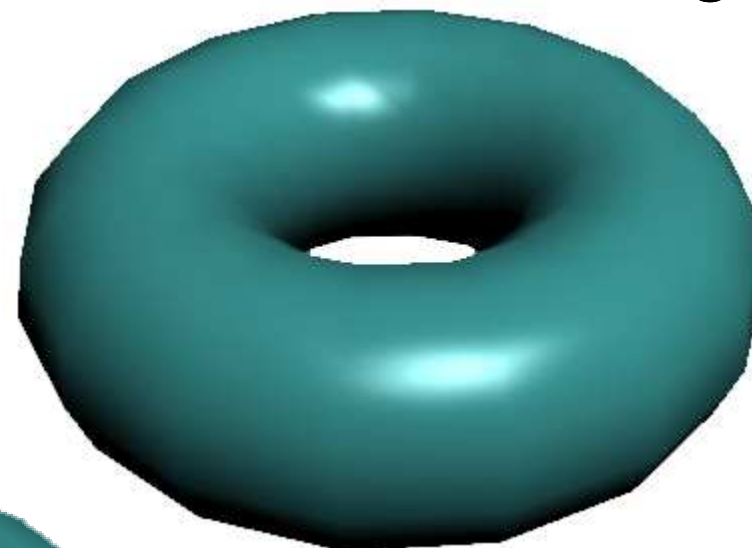
- Soluzione: si interpola nello spazio delle normali e si calcola l'equazione di illuminazione in ogni pixel



# Paragone: costante, Gouraud e Phong



**Costante**

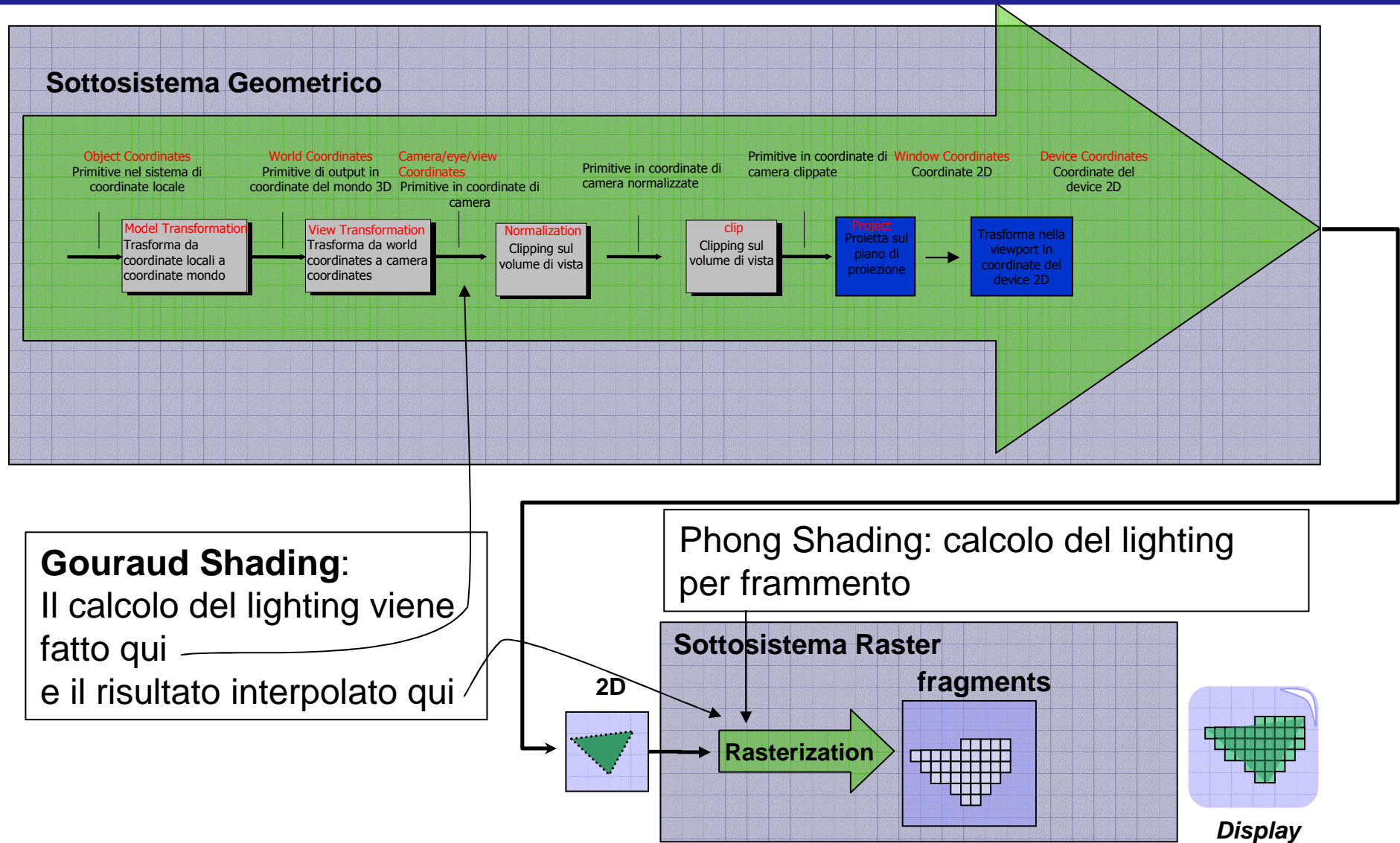


**Phong**



**Gouraud**

# Gouraud Shading, Phong Shading: memo



# Interpolazione

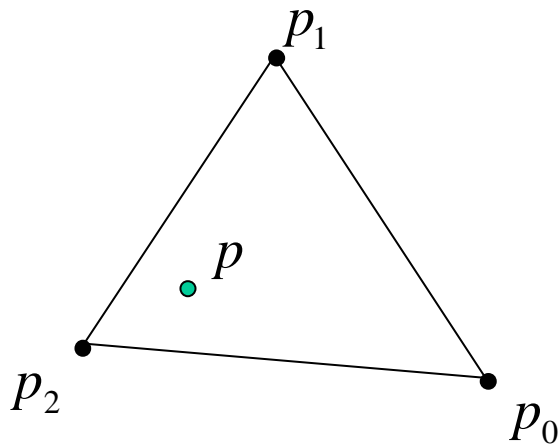
- Interpolazione all'interno del triangolo: coordinate baricentriche

Il punto  $p$  si può esprimere in coordinate **baricentriche**

$$p = \alpha p_0 + \beta p_1 + \gamma p_2$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha, \beta, \gamma \geq 0$$



È comodo anche scriverle così:

$$p = \alpha p_0 + \beta p_1 + (1 - \alpha - \beta) p_2 =$$

$$= p_2 + \alpha(p_0 - p_2) + \beta(p_1 - p_2)$$

$$\alpha + \beta \leq 1$$

$$\alpha, \beta \geq 0$$

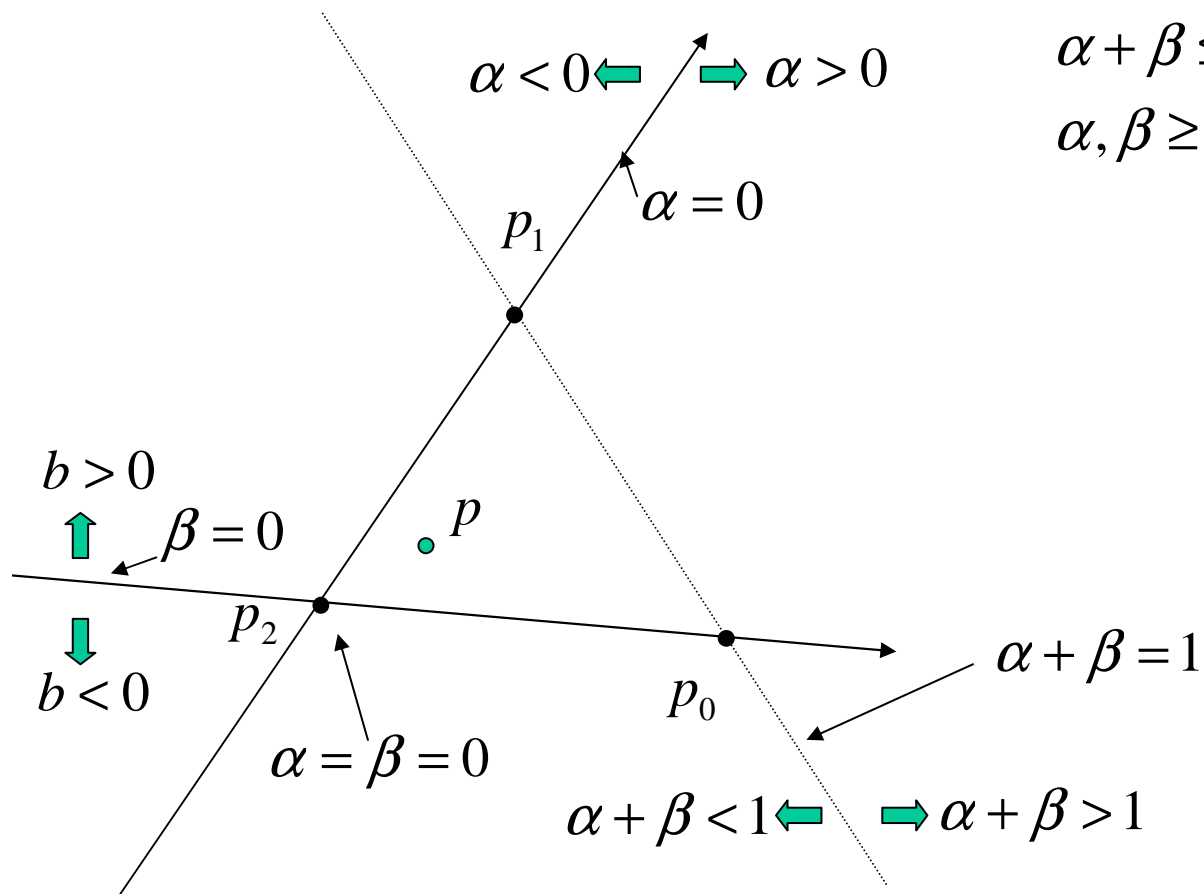
# Nota implementativa

- Interpolazione all'interno del triangolo: coordinate baricentriche

$$p = p_0 + \alpha(p_0 - p_2) + \beta(p_1 - p_2)$$

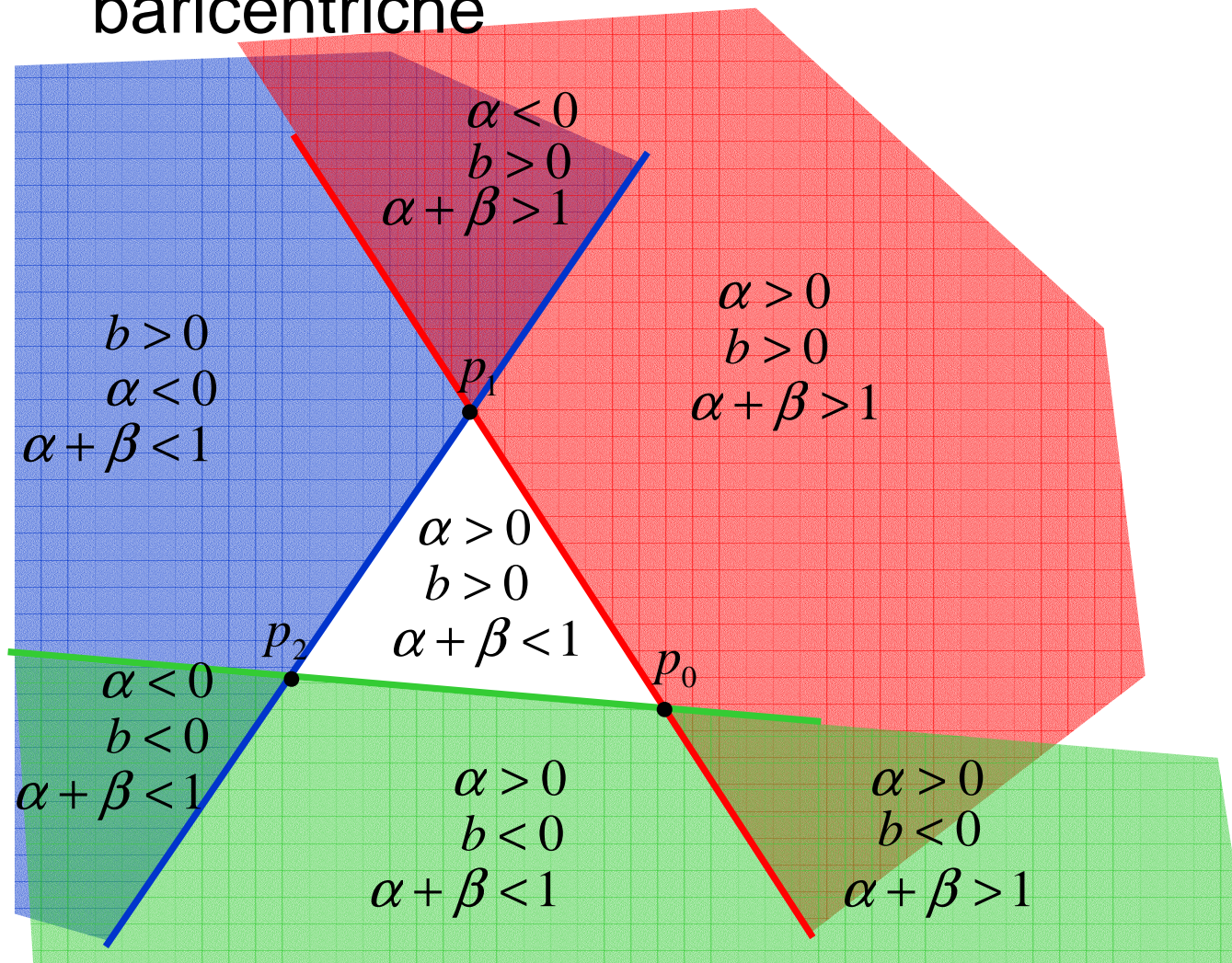
$$\alpha + \beta \leq 1$$

$$\alpha, \beta \geq 0$$



# Nota implementativa

- Interpolazione all'interno del triangolo: coordinate baricentriche



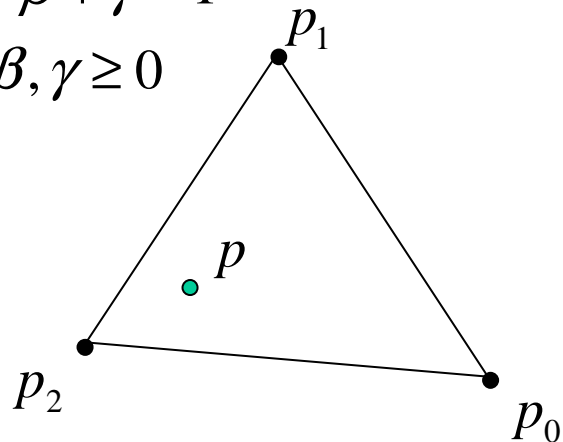
# Nota implementativa

- Come si determinano le coordinate baricentriche?

$$p = \alpha p_0 + \beta p_1 + \gamma p_2$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha, \beta, \gamma \geq 0$$



Per esempio risolvendo il sistema

$$\begin{bmatrix} 1 & 1 & 1 \\ p_{0_x} & p_{1_x} & p_{2_x} \\ p_{0_y} & p_{1_y} & p_{2_y} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 1 \\ p_x \\ p_y \end{bmatrix}$$

$$\alpha = \frac{2 \text{ area di } p, p_1, p_2}{(p_1 - p_0) \times (p_2 - p)} \quad \beta = \frac{2 \text{ area di } p, p_0, p_1}{(p_1 - p_0) \times (p_2 - p)} \quad \gamma = \frac{2 \text{ area di } p, p_1, p_2}{(p_1 - p_0) \times (p_2 - p)}$$

$2 \text{ area di } p_0, p_1, p_2$



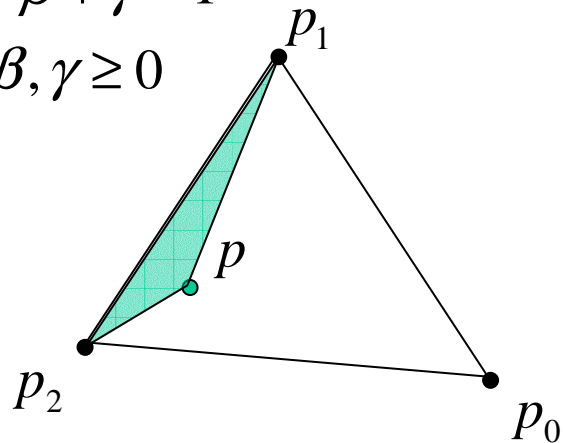
# Nota implementativa

- Come si determinano le coordinate baricentriche?

$$p = \alpha p_0 + \beta p_1 + \gamma p_2$$

$$\alpha + \beta + \gamma = 1$$

$$\alpha, \beta, \gamma \geq 0$$

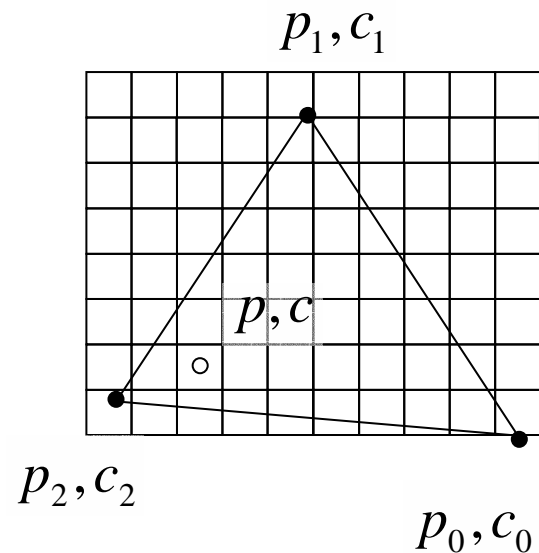


La coordinata baricentrica di un punto  $p$  relativa a un vertice è uguale al **rapporto** tra l'area del triangolo definito dai due **vertici opposti** e dal punto  $p$  e l'area del triangolo

$$\alpha = \frac{\text{Area}(p, p_1, p_2)}{\text{Area}(p_0, p_1, p_2)}$$

# Dove servono?

- Durante la rasterizzazione
  - ES: Gouraud shading



Dato (centro del pixel corrente nella rasterizzazione) determino le sue coordinate baricentriche

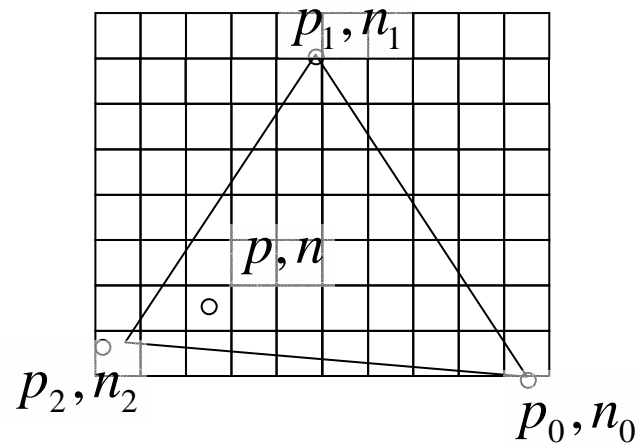
$$p = \alpha p_0 + \beta p_2 + \gamma p_2$$

...e le uso per interpolare il colore

$$c = \alpha c_0 + \beta c_2 + \gamma c_2$$

# Dove servono?

- Durante la rasterizzazione
  - ES: Phong shading



Dato (centro del pixel corrente nella rasterizzazione) determino le sue coordinate baricentriche

$$p = \alpha p_0 + \beta p_1 + \gamma p_2$$

...e le uso per interpolare la normale

$$n = \alpha n_0 + \beta n_1 + \gamma n_2$$