Modello di Illuminazione di Phong

Componenti del Modello di Illuminazione di Phong

Il modello di Phong è un modello di illuminazione locale, cioè che simula la luminosità di una superficie colpita direttamente ed esclusivamente da una sorgente luminosa; in pratica:

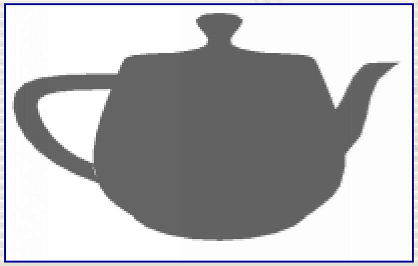
luce finale =

- (sorgente uniforme) illuminazione *ambiente*
- \+
- (sorgente puntiforme) *riflessione*
  - *diffusa*
  - \+
  - *speculare*

Componente *ambiente*

- Modella approssimativamente l'effetto di una luce uniforme che arriva da riflessioni secondarie

- Ogni superficie, anche quelle in ombra, sono raggiunte da luce proveniente da tutte le direzioni

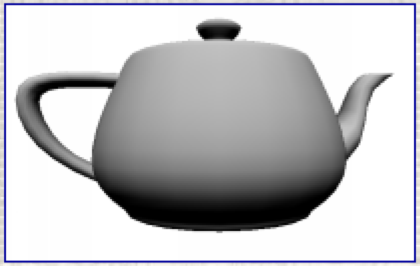


solo componente *ambiente*

$$I_{da} = k_a I_a$$

Componente *riflessione diffusa*

- E' tipica in superfici come per es.:
  - materiali molto opachi (il contrario di lucidi)
  - certi tipi di legno
  - gesso
  - ....

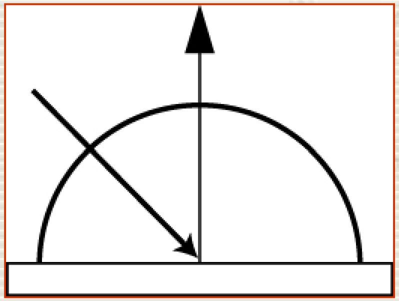


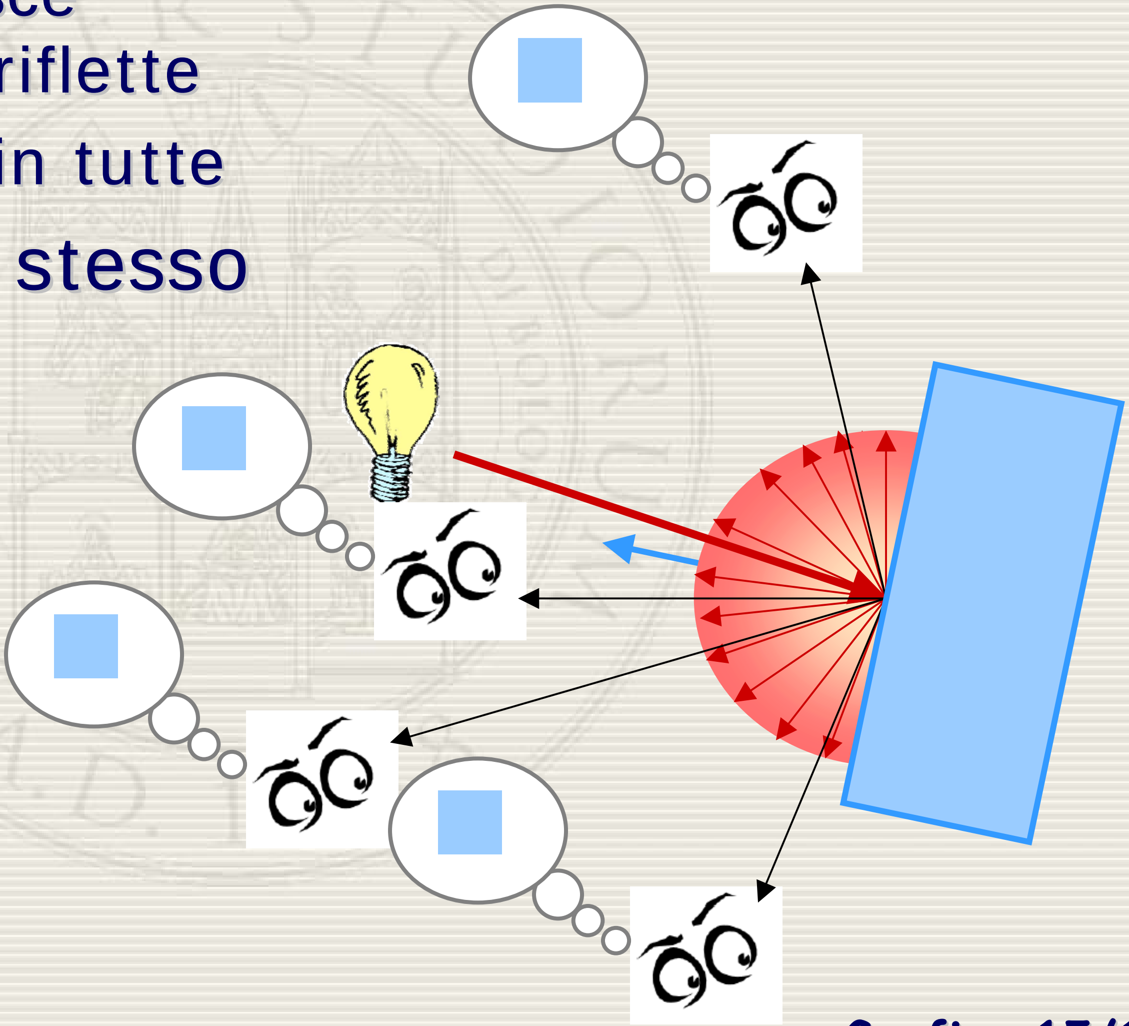
con componente *riflessione diffusa*

- Viene detta anche
  - Riflessione Lambertiana (da J.H.Lambert 1728-1777)

Componente *riflessione diffusa*

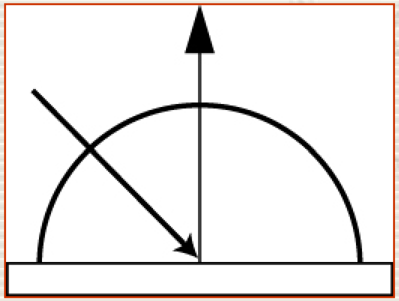
- La luce che colpisce una superficie si riflette (nel semispazio) in tutte le direzioni nello stesso modo

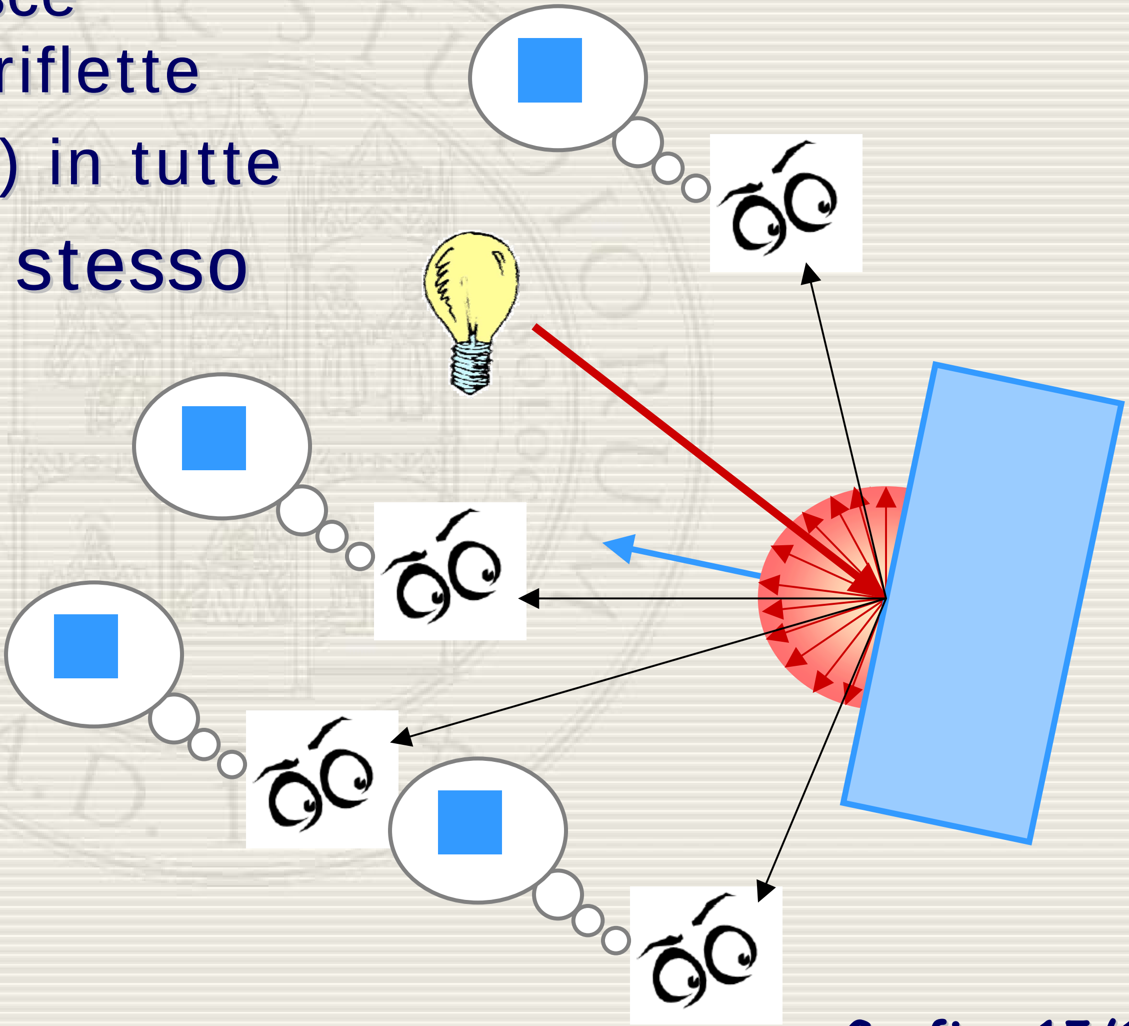




Componente *riflessione diffusa*

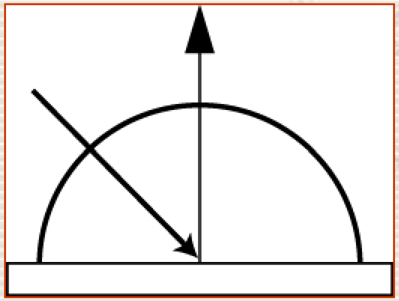
- La luce che colpisce una superficie si riflette (nella semispazio) in tutte le direzioni nello stesso modo





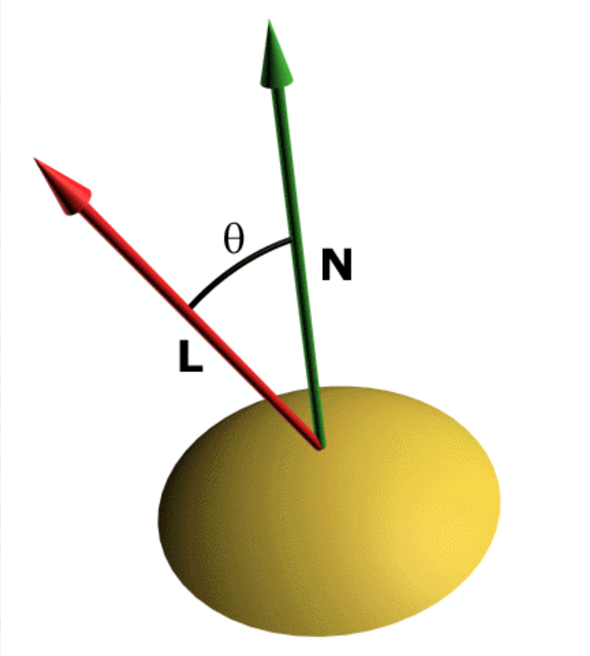
Componente *riflessione diffusa*

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



Componente *riflessione diffusa*

- Dipende da:
  - o **N** normale alla superficie (orientamento della superficie)
  - o **L** direzione della luce o del raggio incidente
  - o θ angolo compreso



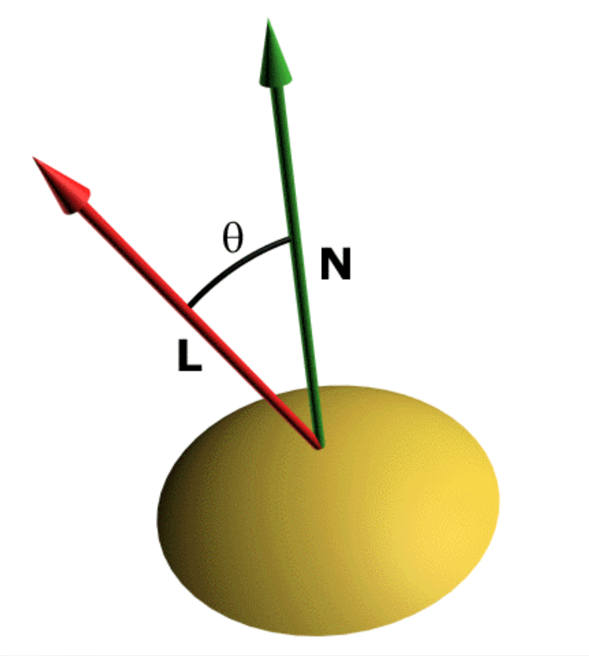
$$I_{dr} = k_d I_l \cos(\theta)$$

$K_R, K_G, K_B \in [0,1]$
(determina il colore dell'oggetto)

R, G, B
(luce bianca: 1,1,1)

Componente *riflessione diffusa*

- Dipende da:
  - **N** normale alla superficie (orientamento della superficie)
  - **L** direzione della luce o del raggio incidente
  - θ angolo compreso

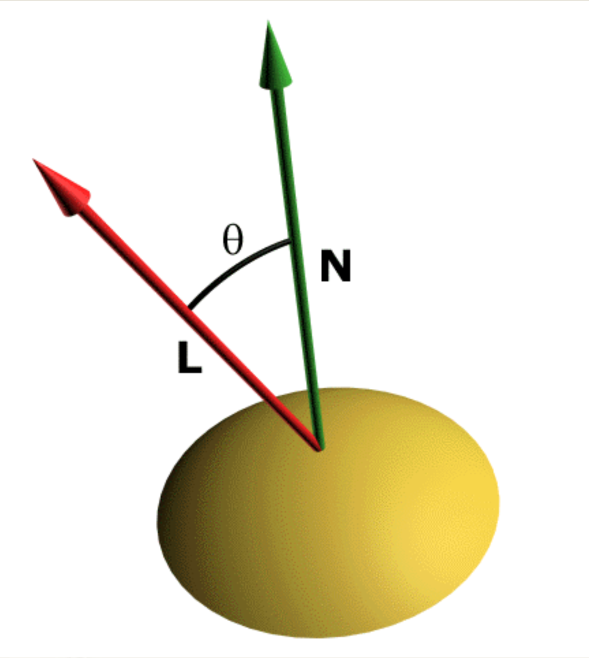


$$I_{dr} = k_d I_l \cos(\theta)$$

se l' angolo è compreso fra 0º e 90º,
altrimenti niente,
(oggetto in ombra di se stesso)

Componente *riflessione diffusa*

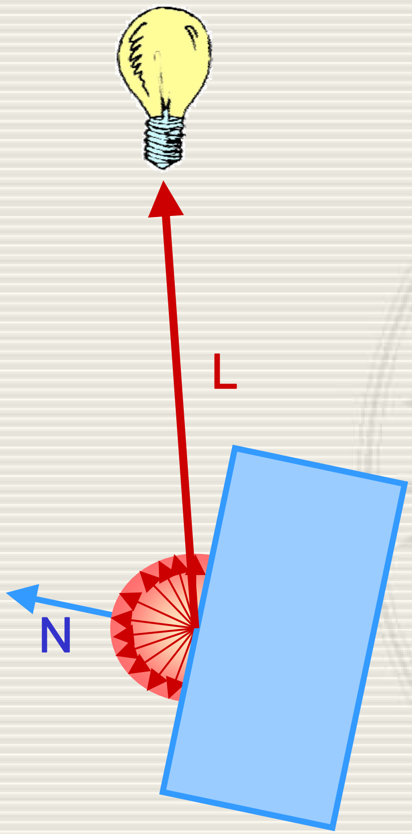
- Dipende da:
  - **N** normale alla superficie (orientamento della superficie)
  - **L** direzione della luce o del raggio incidente
  - θ angolo compreso



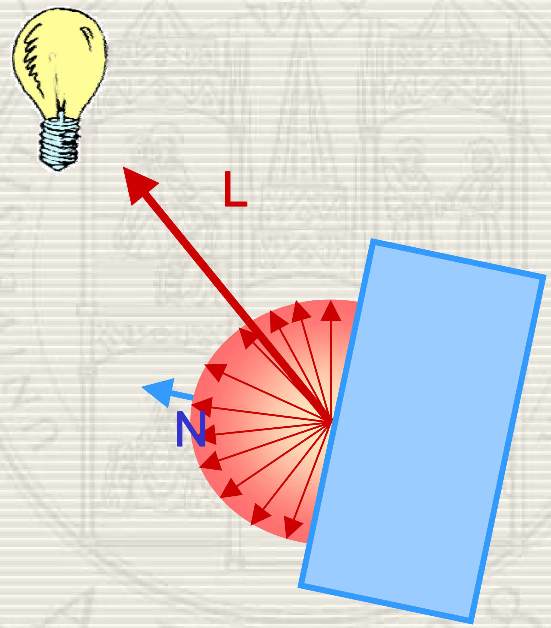
$$I_{dr} = k_d I_l \cos(\theta)$$

$$= k_d I_l (L \cdot N)$$

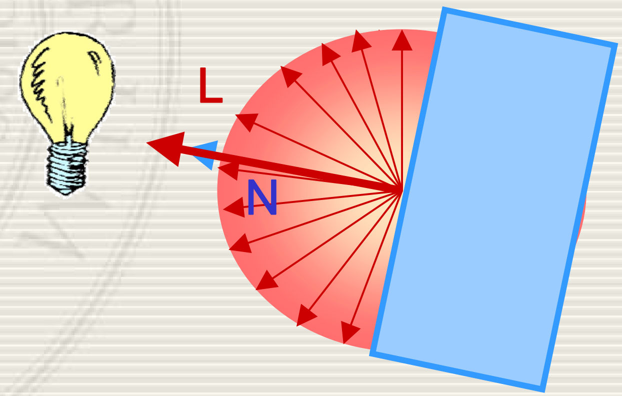
Componente *riflessione diffusa*



componente
diffusa
piccola
θ=70º

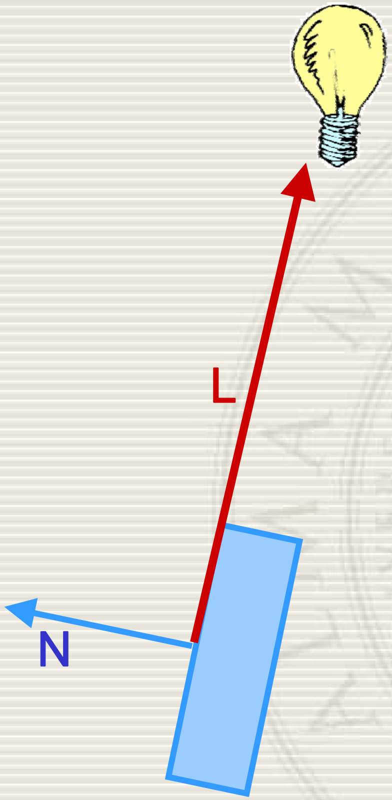


componente
diffusa
grande
θ=35º

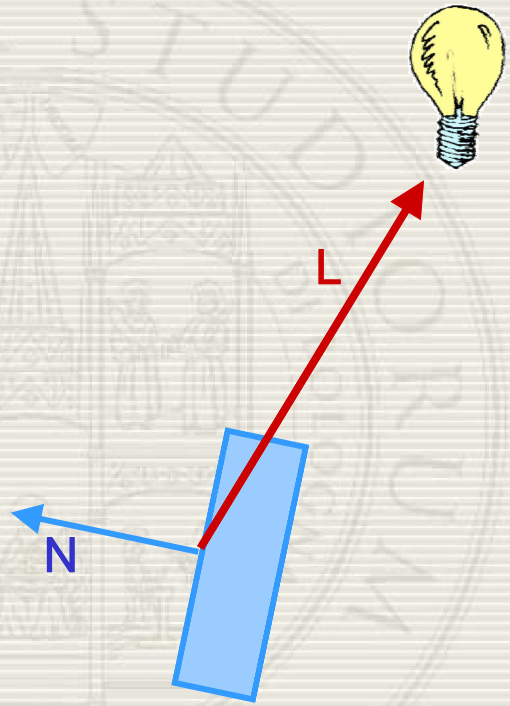


componente
diffusa
massima
θ=0º

Componente *riflessione diffusa*



componente
diffusa
ZERO
$\theta=90°$



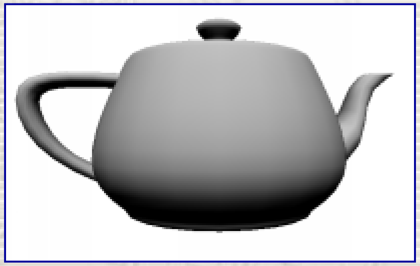
componente
diffusa
ZERO
$\theta>90°$

la superficie
non è esposta
alla luce

Componente *riflessione speculare*

- E' tipica in superfici come per es.:
  - ceramica
  - metallo
  - …
- Per materiali lucidi
  - con riflessi brillanti (highlights)

Componente *ambiente*

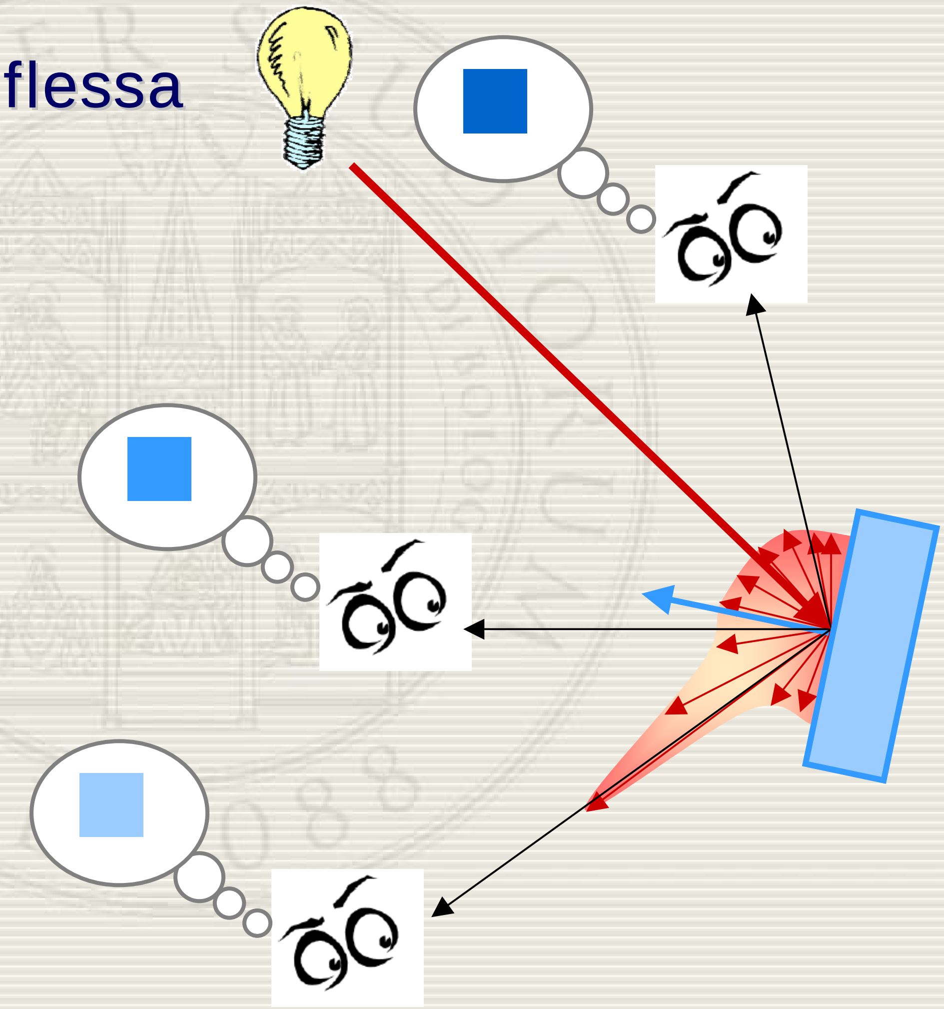


Componente *riflessione diffusa*

Componente *riflessione speculare*

Componente *riflessione speculare*

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



Componente *riflessione speculare*

L: raggio incidente

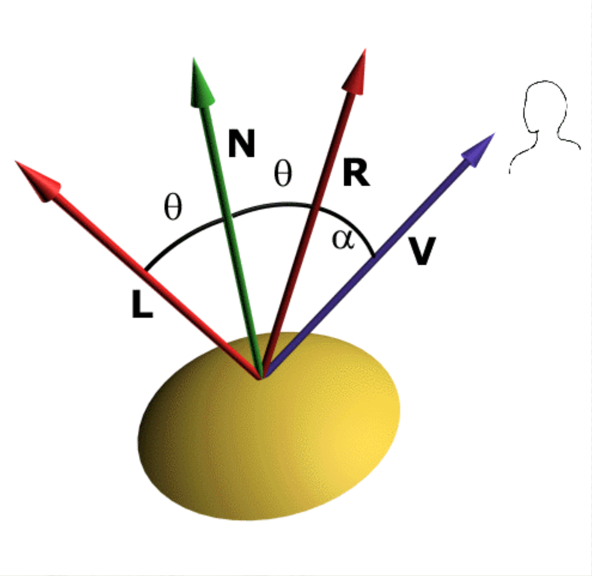
N: normale

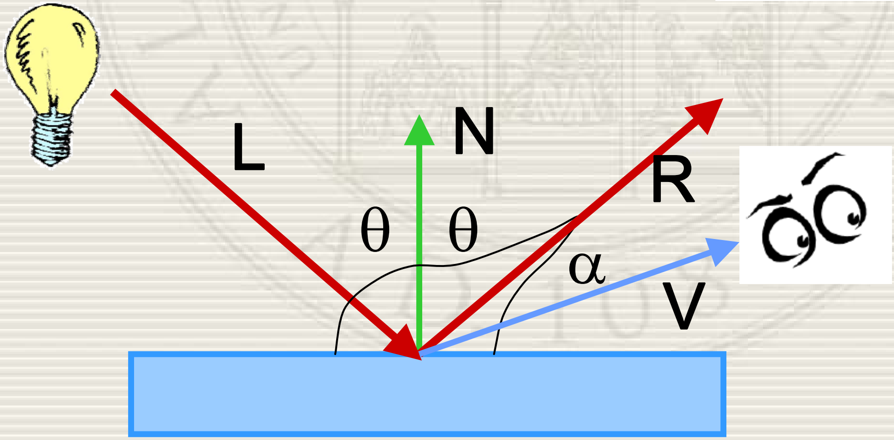
R: raggio riflesso

V: direzione di vista

θ: angolo fra L ed N

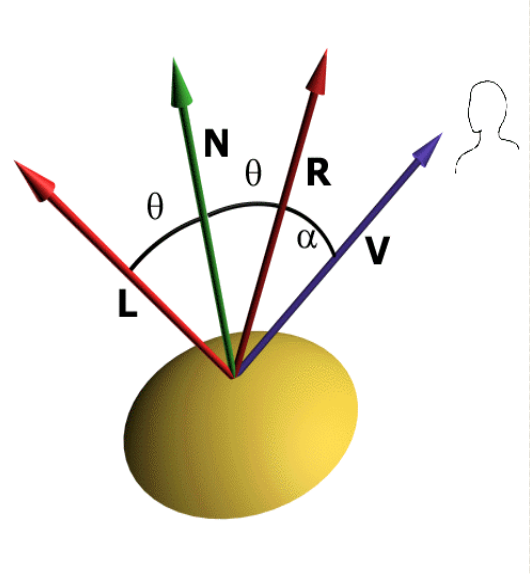
α: angolo fra R e V





Componente *riflessione speculare*

- Modello di Illuminazione di Phong
  (Bui-Tuong Phong, 1975)



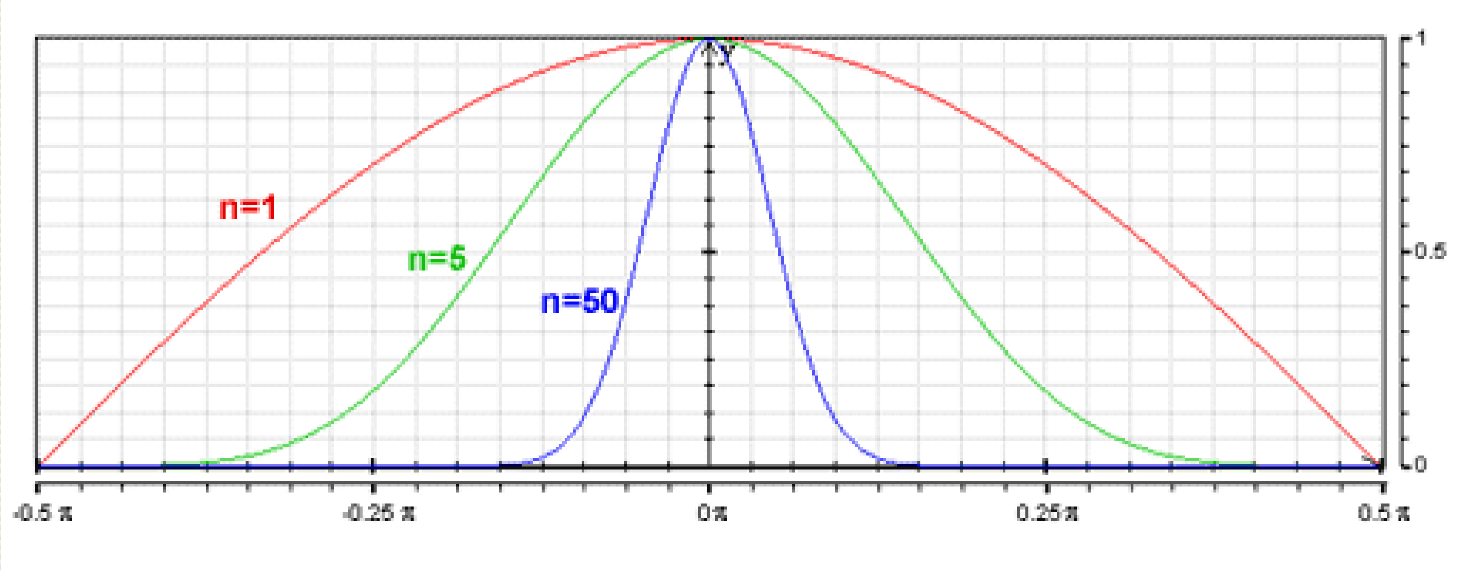
$$I_r = k_s \cdot I_l \cdot (\cos\alpha)^n$$

$$= k_s \cdot I_l \cdot (R \cdot V)^n$$

Scalari caratteristici del "materiale" dell'oggetto

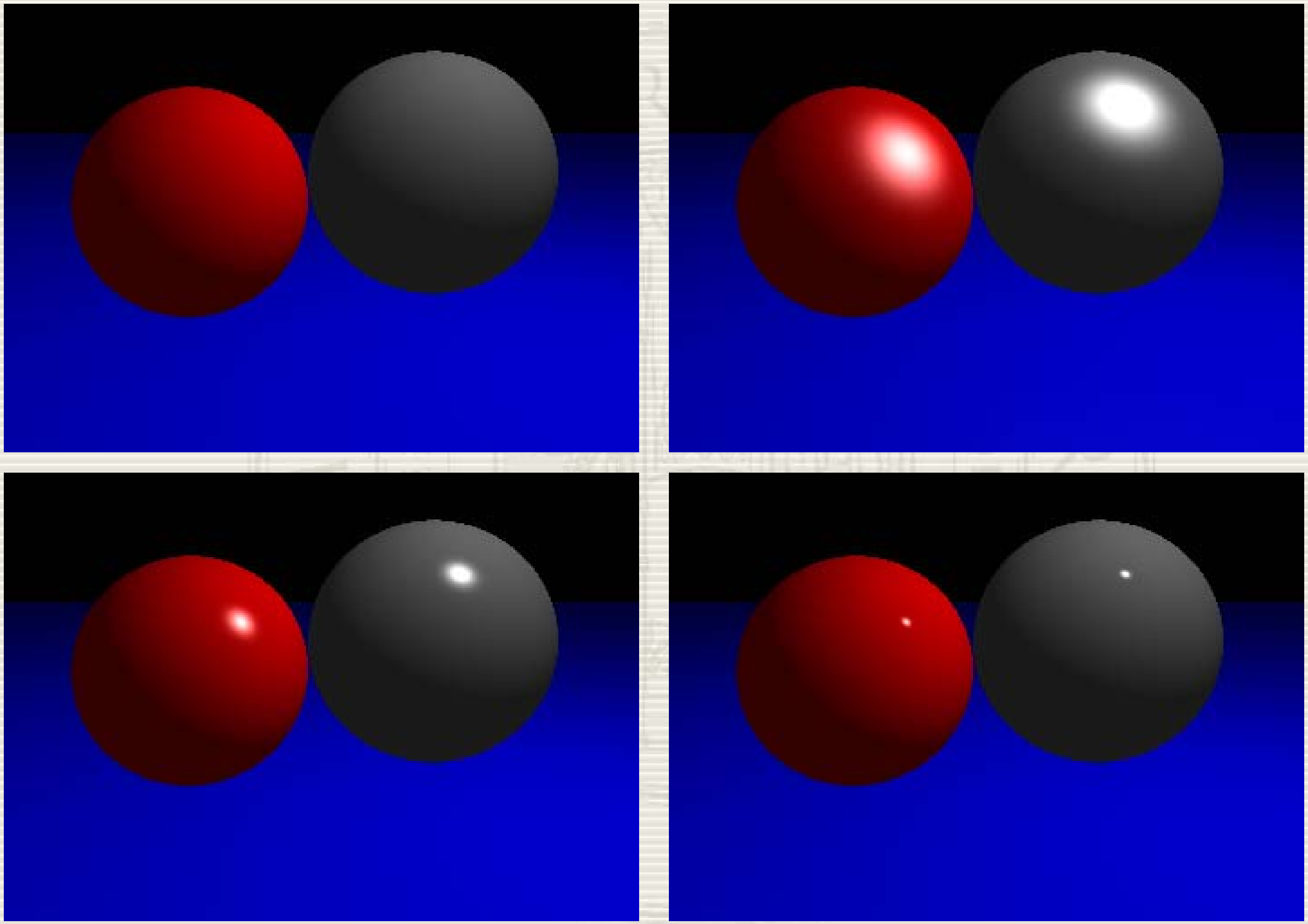
Componente *riflessione speculare*

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



$$y = (\cos\alpha)^n$$

Componente *riflessione speculare*



$n = 5$

$n = 10$

$n = 500$

Equazione di lighting totale

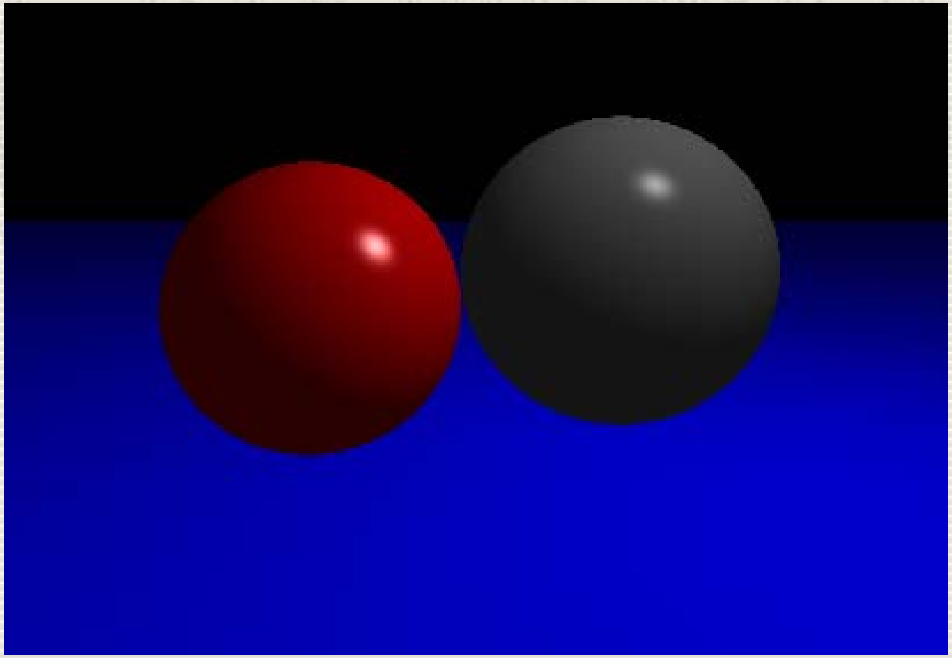
$$I = \boxed{k_a} \boxed{I_a} + \boxed{I_l} \left(\boxed{k_d} (L \cdot N) + \boxed{k_s} (R \cdot V)^{\boxed{n}} \right)$$

propretà della luce

propretà del materiale

Equazione di lighting totale

$$I = \boxed{k_a}\boxed{I_a} + \boxed{I_l}\left(\boxed{k_d}(L \cdot N) + \boxed{k_s}(R \cdot V)^{\boxed{n}}\right)$$

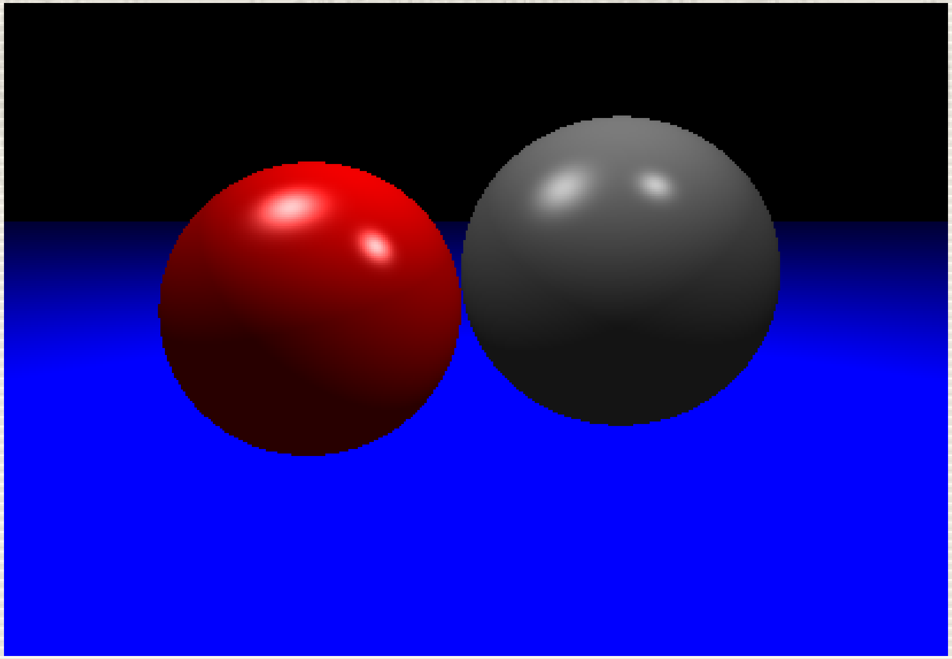


Equazione di lighting totale

$$I = \boxed{k_a}\boxed{I_a} + \boxed{I_l}\left(\boxed{k_d}(L \cdot N) + \boxed{k_s}(R \cdot V)^{\boxed{n}}\right)$$

$$I = k_a I_a + \sum_i I_{l_i}\left(k_d(L_i \cdot N) + k_s(R_i \cdot V)^n\right)$$

Più sorgenti luminose



Materiali...

Material	GL_AMBIENT	GL_DIFFUSE	GL_SPECULAR
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
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
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
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
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
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
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
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
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
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
Polished Bronze	0.25 0.148 0.06475 1.0	0.4 0.2368 0.1036 1.0	0.774597 0.458561 0.200621 1.0
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

k_a k_d k_s

Materiali in file .obj

```
mtllib pippo.mtl
...
usemtl red
f 2096 2097 2084
...
usemtl blue
f 2076 2075 2104
...
```

file.obj

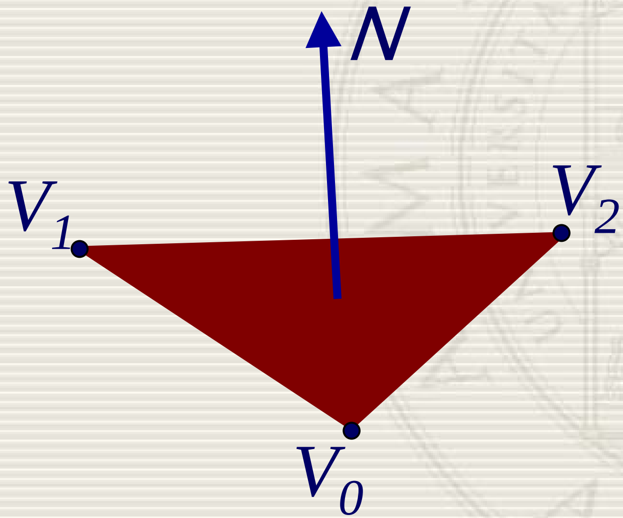
```
newmtl red
Ka 0.4000 0.4000 0.4000
Kd 0.3000 0.0000 0.0000
Ks 0.3000 0.3000 0.3000
illum 2
Ns 60.0000

newmtl blue
Ka 0.4000 0.4000 0.4000
Kd 0.0000 0.0000 0.3000
Ks 0.3000 0.3000 0.3000
illum 2
Ns 60.0000
```

pippo.mtl

Potenza n nel modello di Phong

Normale di un triangolo

- Cioè il suo orientamento nello spazio

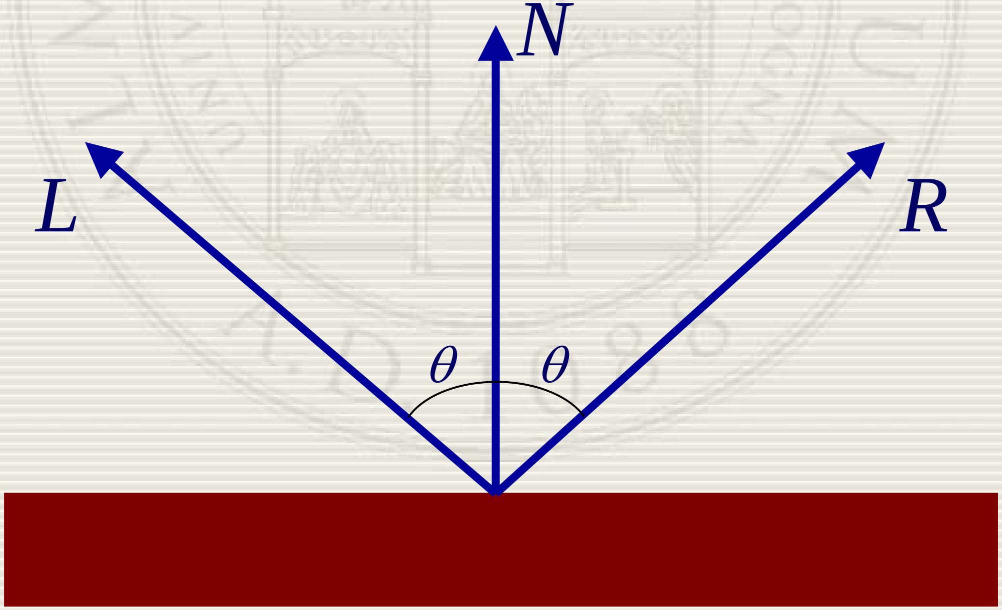


$$N = (V_2 - V_0) \times (V_1 - V_0)$$

$$\hat{N} = \frac{N}{\| N \|}$$

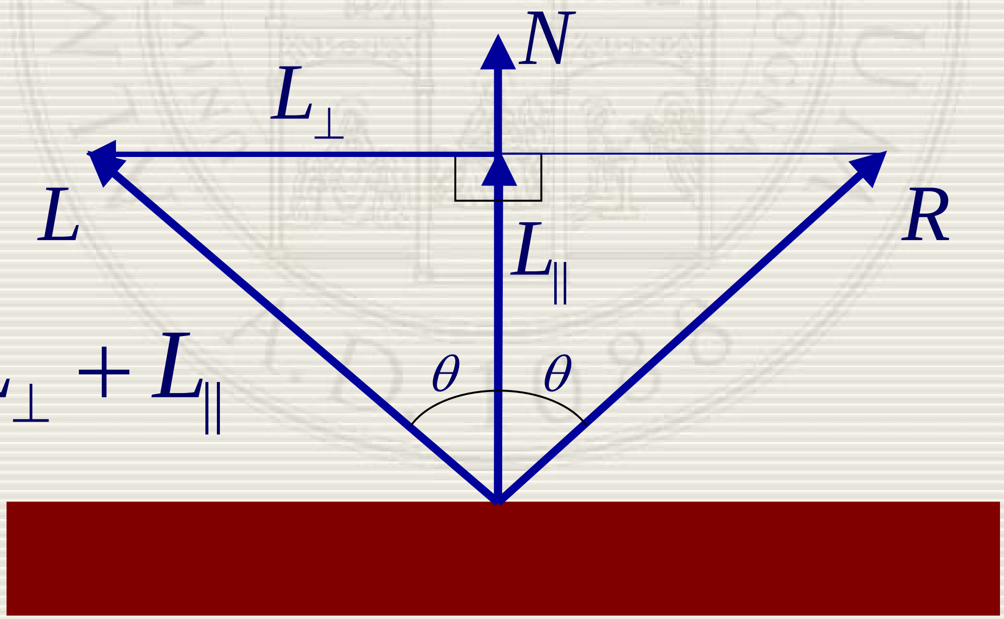
Ancora su *riflessione speculare*

- Come si calcola il vettore riflesso R ?



Ancora su *riflessione speculare*

- Come si calcola il vettore riflesso R ?



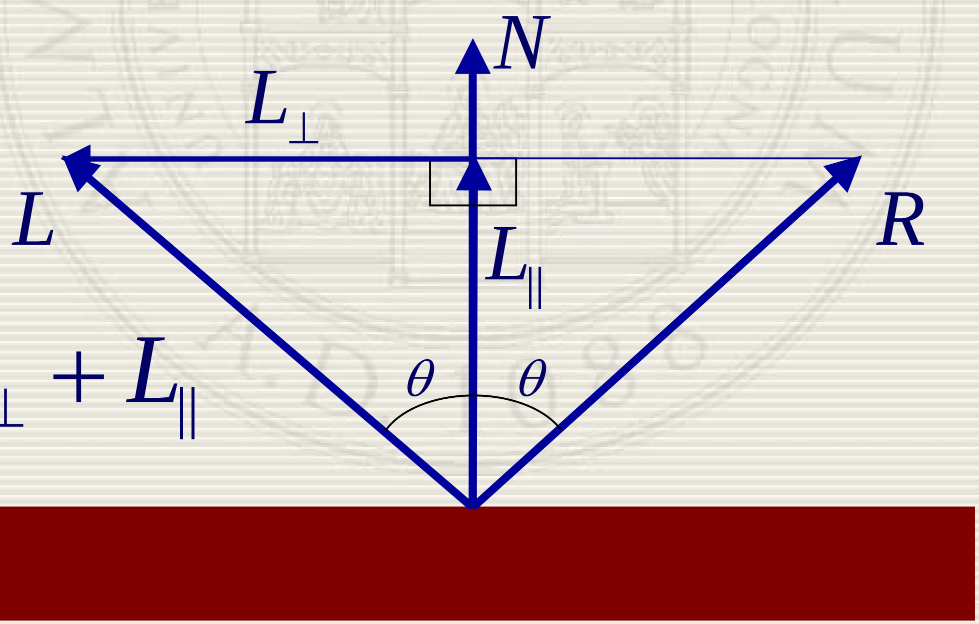
$$L = L_\perp + L_\parallel$$

Ancora su *riflessione speculare*

- Come si calcola il vettore riflesso R ?

$$L_{\parallel} = N\cos(\theta) = N(L \cdot N)$$

$$L_{\perp} = L - L_{\parallel}$$

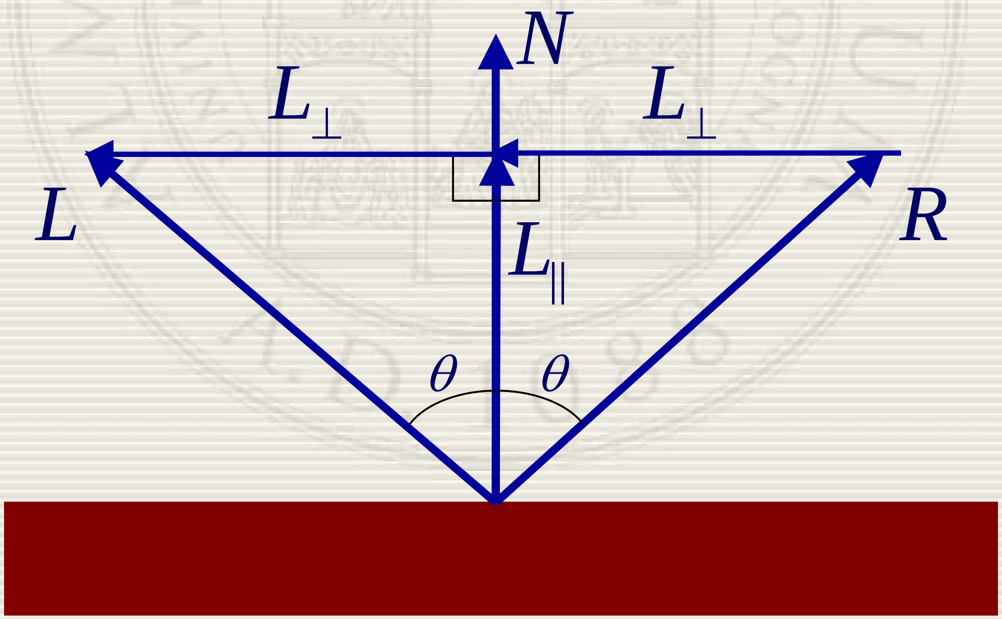


$$L = L_{\perp} + L_{\parallel}$$

Ancora su *riflessione speculare*

- Come si calcola il vettore riflesso R ?

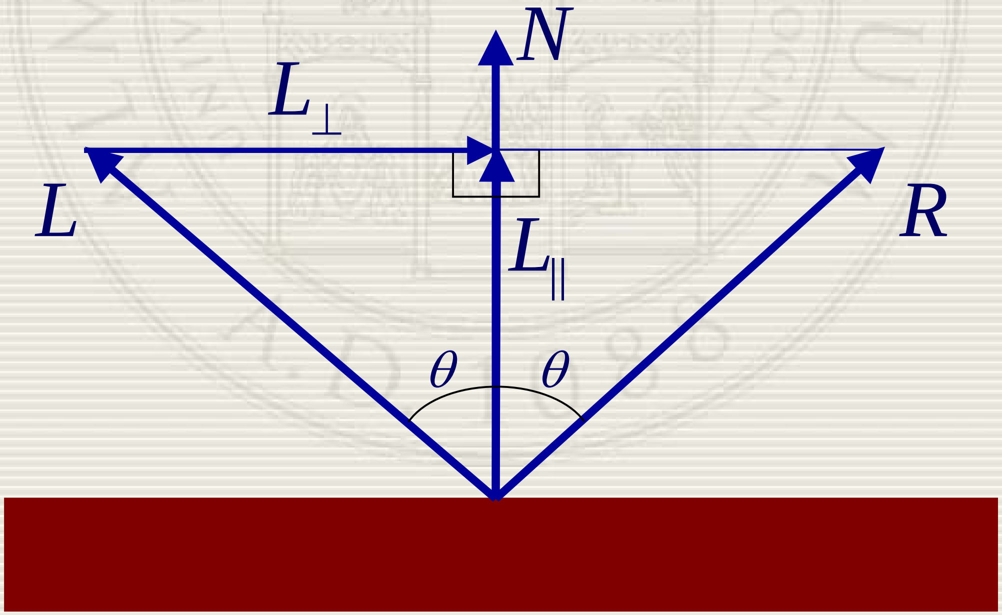
$$R = L_{\parallel} - L_{\perp}$$



Ancora su *riflessione speculare*

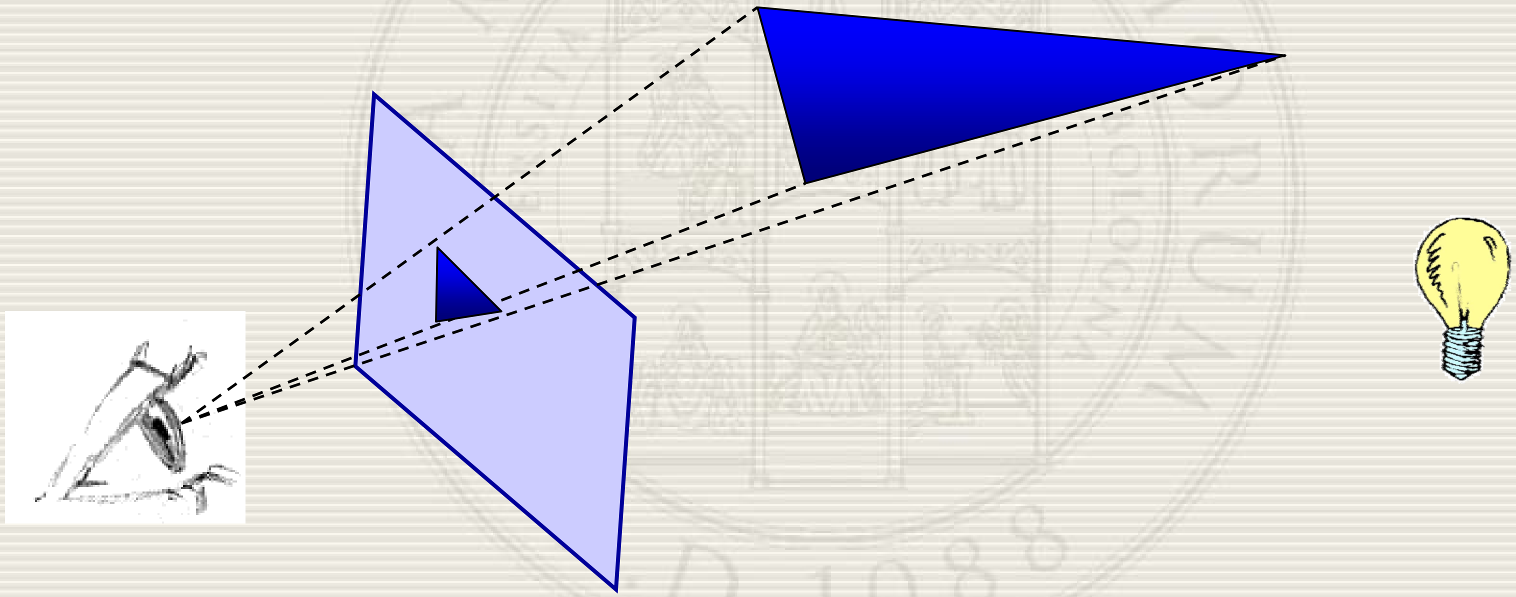
- Come si calcola il vettore riflesso R ?

$$R = L_{\parallel} - L_{\perp}$$
$$= 2(L \cdot N)N - L$$



Problema

- Come si colora un triangolo illuminato? mediante rasterizzazione con una particolare Tecnica di Shading

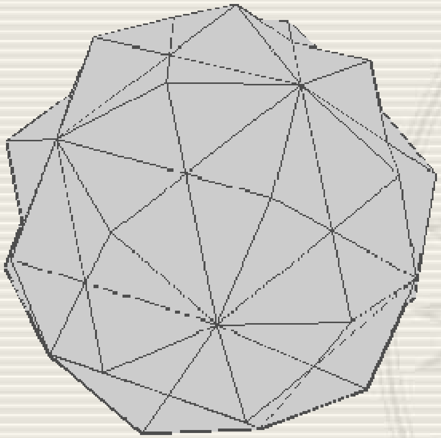


Algoritmi di Shading

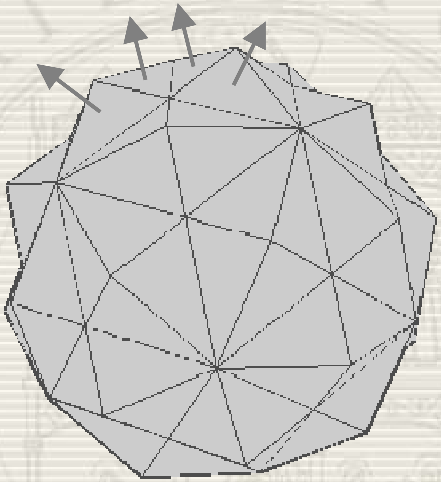
- Vediamo le seguenti semplici Tecniche di Shading:
  - ❑ Flat Shading
  - ❑ Gouraud Shading
  - ❑ Phong Shading

Flat Shading:

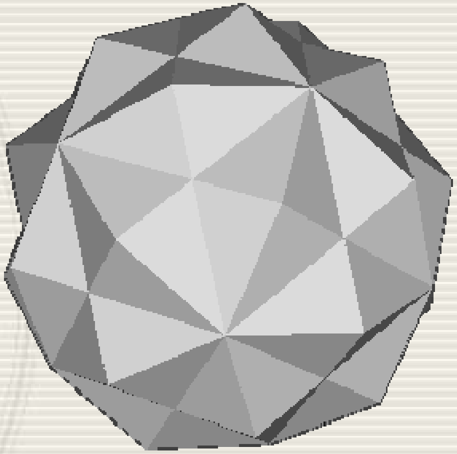
Illuminazione faccia per faccia



1.
geometria di partenza



2.
per ogni faccia,
calcolo della normale



3.
applico modello di illuminazione al centro di ogni faccia e calcolo un colore per faccia

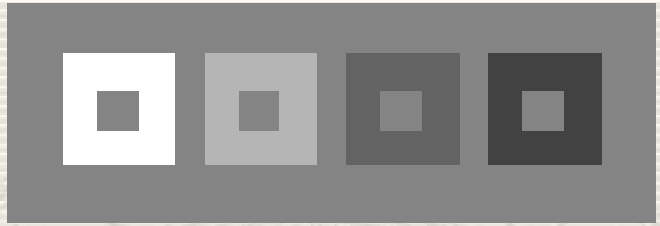
Flat Shading: problema

Le superfici curve vengono approssimate con triangoli (tassellazione), poi si applica il Flat Shading ad ogni triangolo

Risultato:

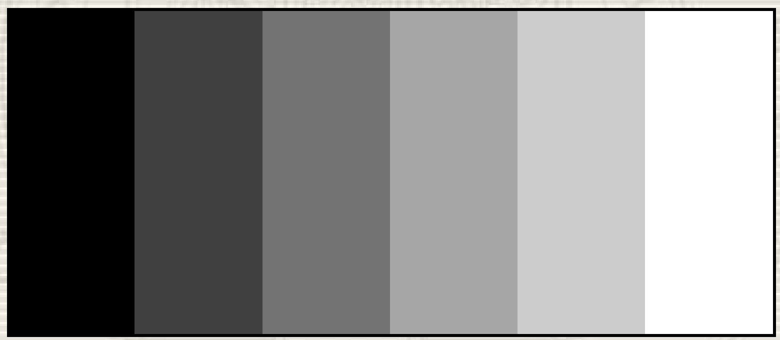
spigoli evidenti: un brutto risultato!

Flat Shading: problema



A peggiorare le cose c’è l'effetto ottico:

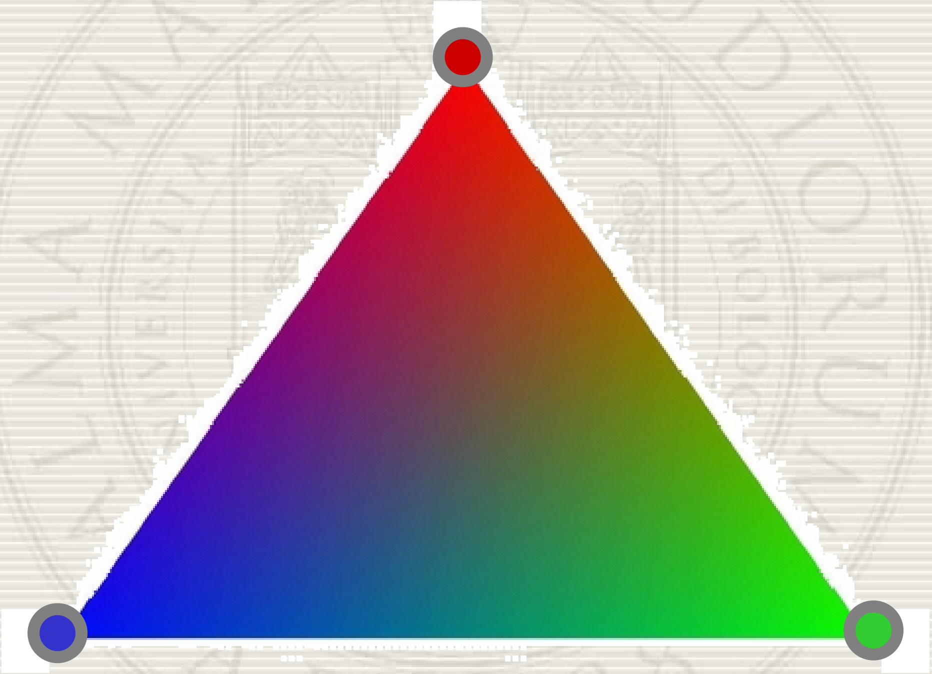
Mach-band



Il contrasto fra zone di colore uniforme difficilmente sfugge al nostro occhio (neanche se le zone sono molte, e la differenza fra loro è relativamente piccola).
Il cervello umano aumenta il contrasto fra zone di colore uniforme.

Idea già vista

Utilizziamo l'interpolazione colore dentro la faccia



Gouraud Shading (Henri Gouraud, 1971)

Utilizzare l'interpolazione del colore dentro la faccia

1- Si applica il modello di illuminazione ai 3 vertici di ogni triangolo; si determina un colore per ciascun vertice
2- Si interpolano i 3 colori nel triangolo

Per applicare il modello di illuminazione serve la normale!
Normale non definita per una faccia, ma per un vertice!

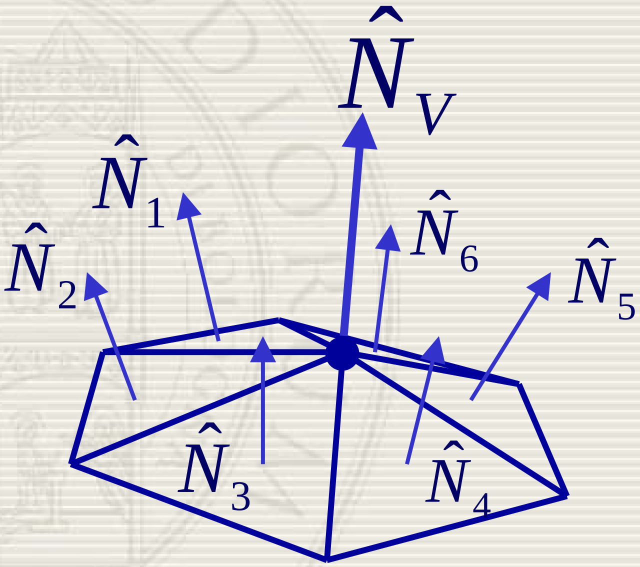
Normali per vertice

Normale per un vertice condiviso da *n* triangoli:

$$N = \hat{N}_1 + \hat{N}_2 + \ldots + \hat{N}_n$$

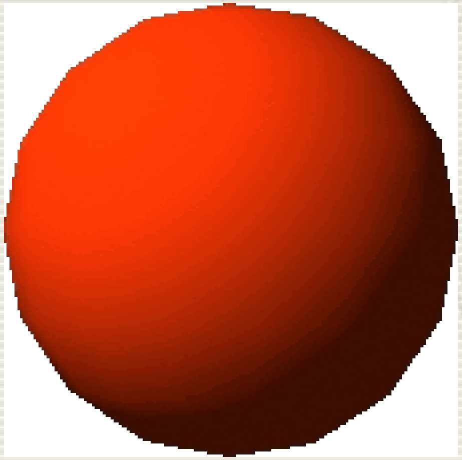
Quindi si normalizza

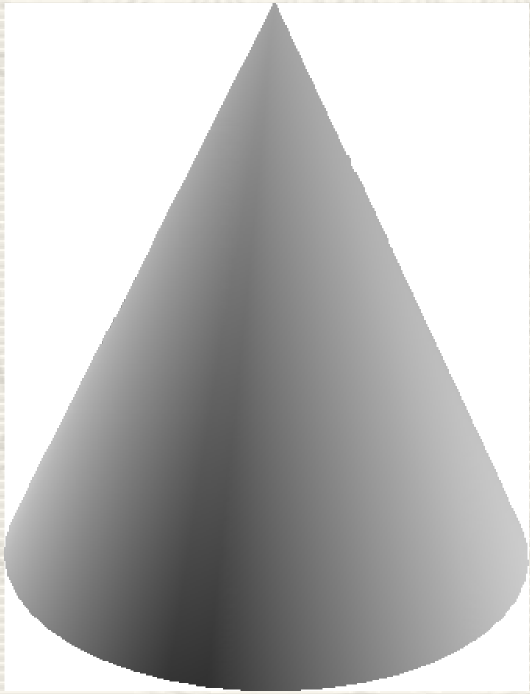
$$\hat{N}_V = \frac{N}{\| N \|}$$

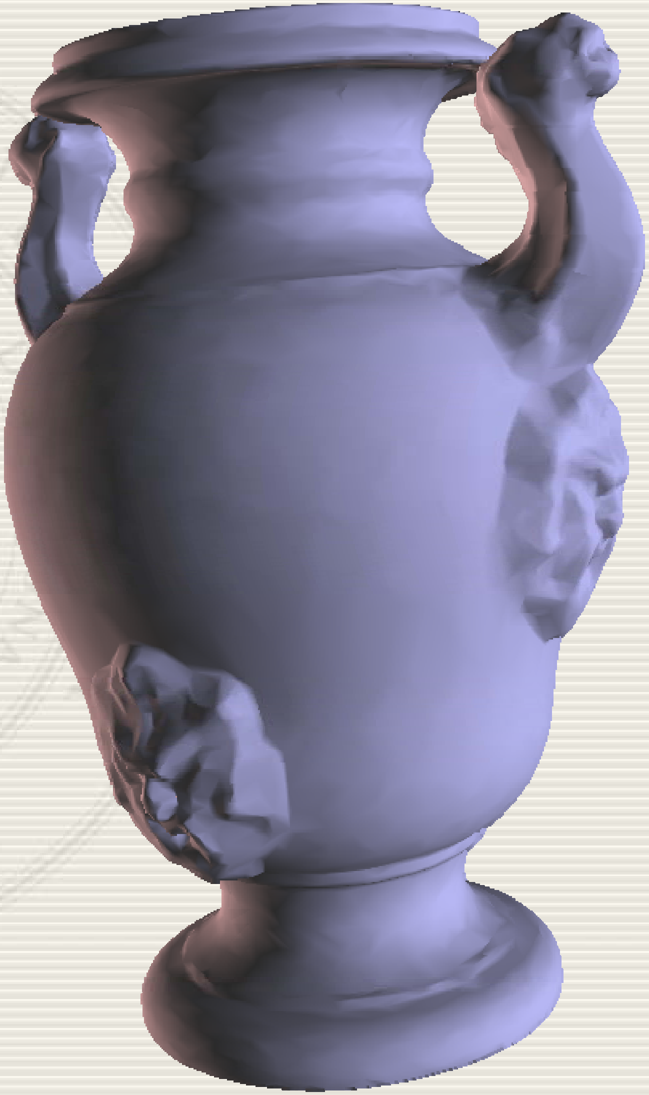


Gouraud Shading

Risultati:

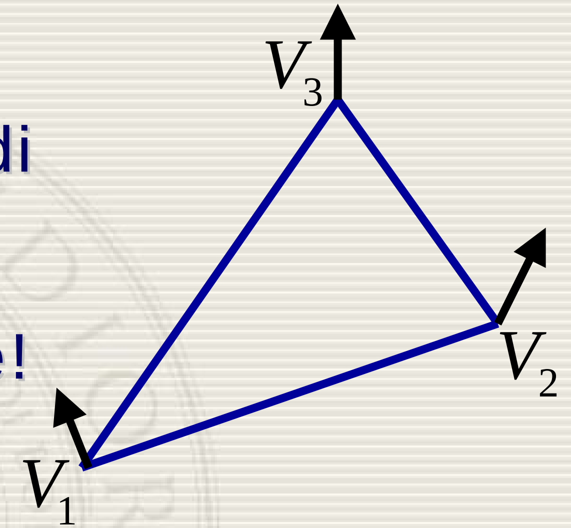






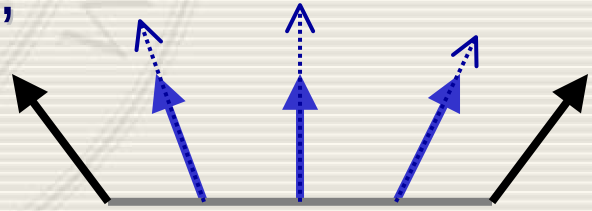
Si può fare meglio?

- Invece di interpolare il colore *dopo* l'applicazione del modello di illuminazione nei 3 vertici, si interpola la normale nei 3 vertici *prima* di applicare l'illuminazione!



- Attenzione:
  interpolando due vettori normali, non si ottiene un vettore normale:

  si deve rinormalizzare dopo l'interpolazione



Phong Shading (Bui-Tuong Phong, 1973)

1- Si interpolano le normali nei 3 vertici per ottenere differenti normali nella faccia

2- Si rinormalizzano

3- Si applica il modello di illuminazione ad ogni punto

Attenzione a non confondere il Phong Shading (uno shading) con il Phong Lighting Model (un modello di illuminazione)

Gouraud vs Phong Shading

- Goraud Shading - illuminazione per vertice;
  molto veloce:
  si applica l'illuminazione una volta per vertice!

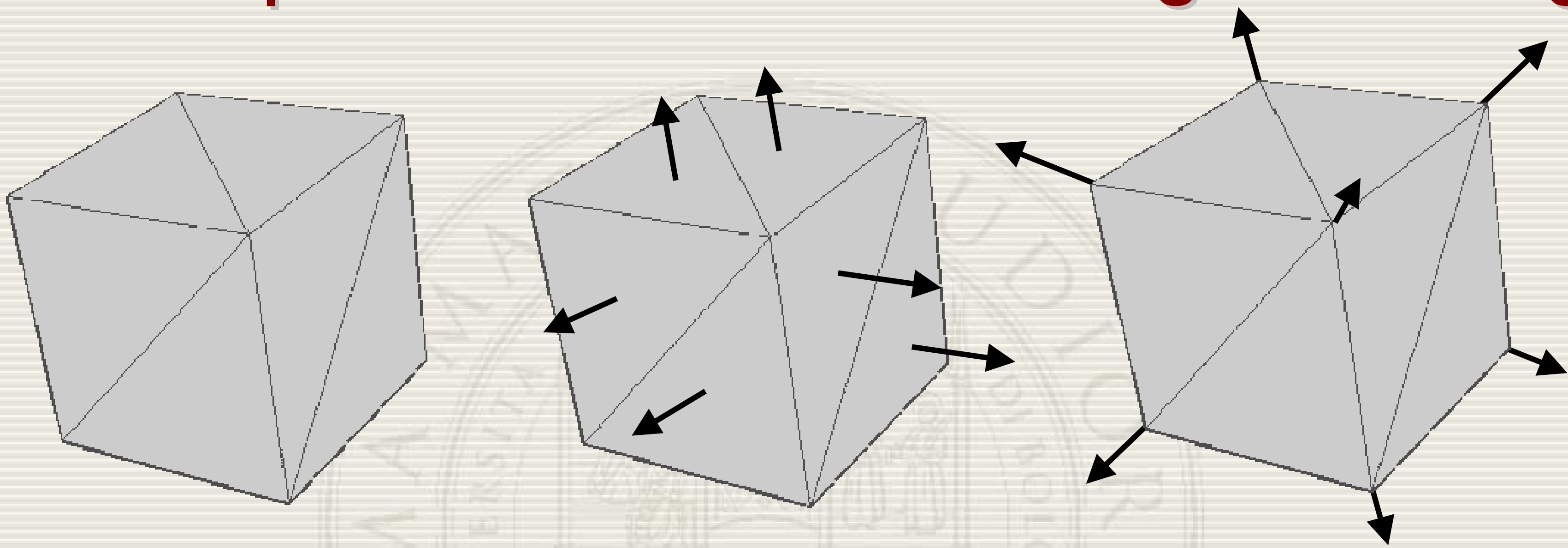
- Phong Shading - illuminazione per "frammento";
  risultati migliori, ma molto costoso
  ottimo per riflessi luminosi (esponente speculare alto)

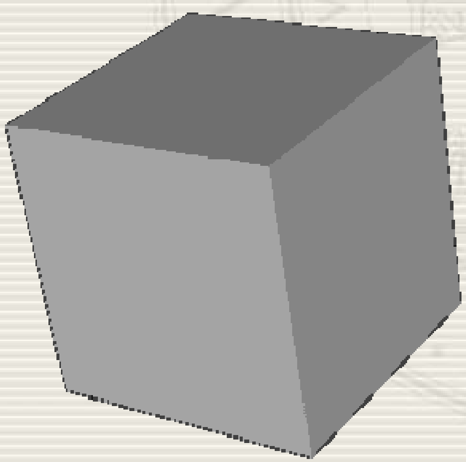
Flat Shading

Goraund Shading

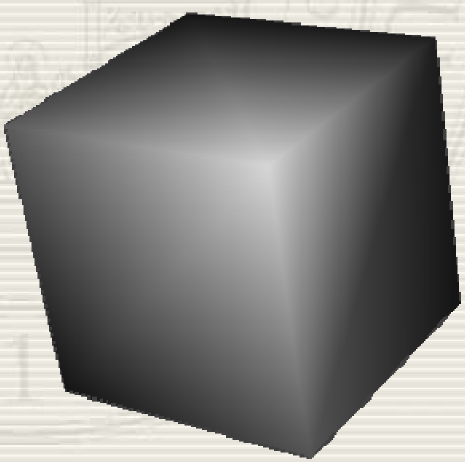
Phong Shading

Note per Gouraud e Phong Shading





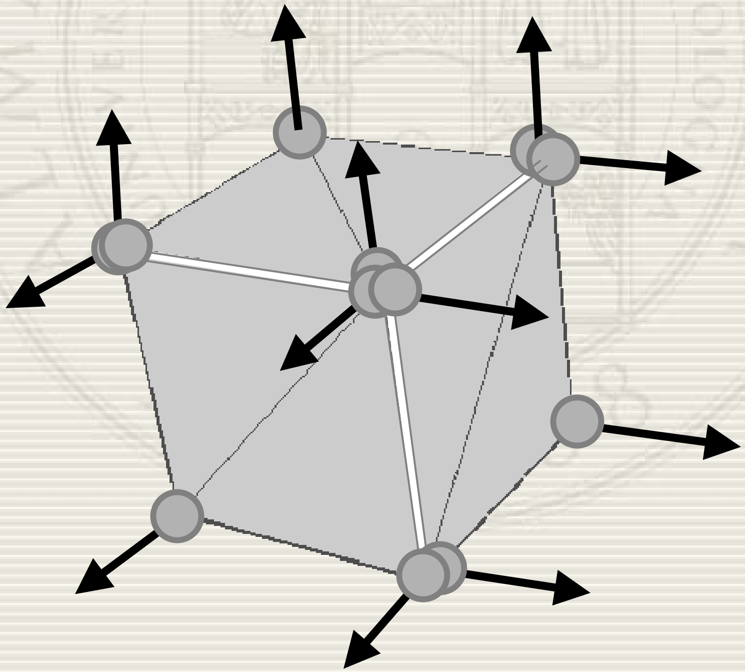
Flat Shading



Goraud Shading
(Phong Shading in questso caso è simile)

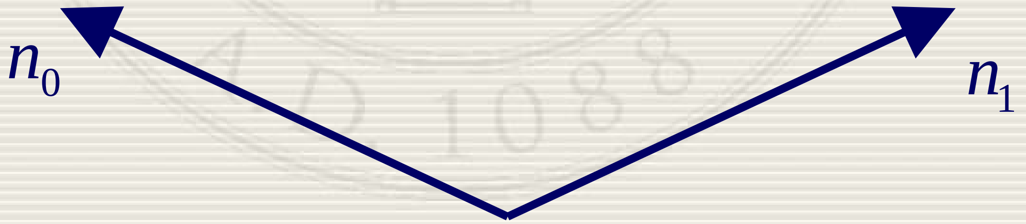
Note per Gouraud e Phong Shading

- Goraud e Phong servono per superfici lisce
  - eliminano gli spigoli artefatti
  - ma eliminano anche gli spigoli veri!
- in questo ultimo caso duplicare i vertici



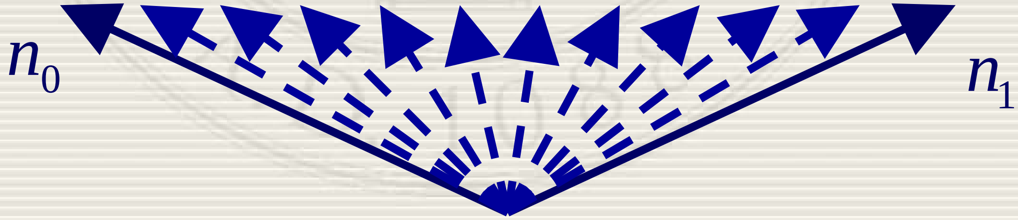
Nota su interpolazione delle normali

- Si procede come per il colore
- Si deve rinormalizzare
- Non produce normali equispaziate!



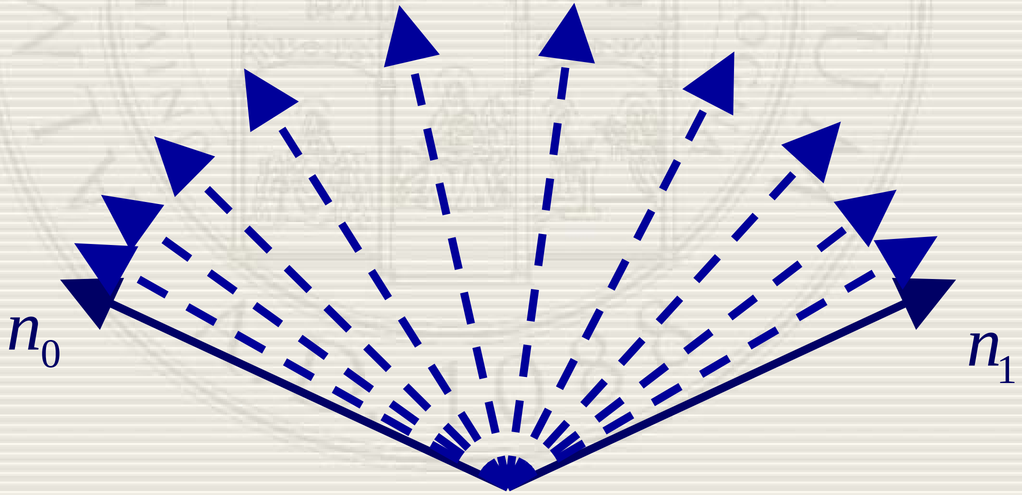
Interpolazione Normali

- Si procede come per il colore
- Si deve rinormalizzare
- Non produce normali equispaziate!

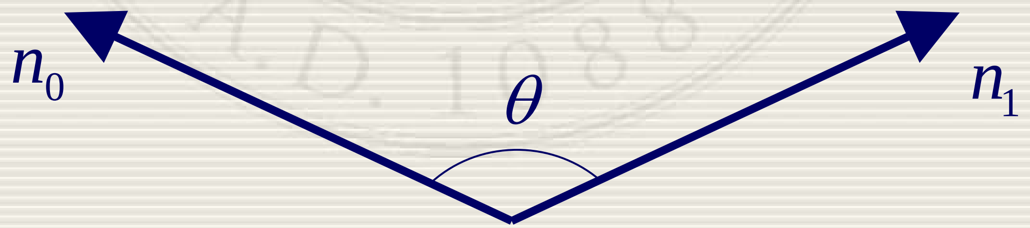


Interpolazione Normali

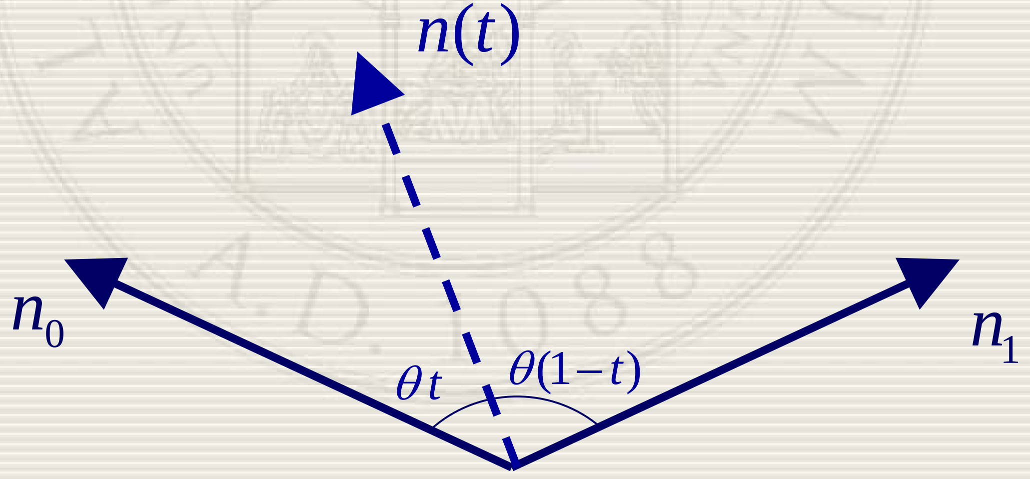
- Si procede come per il colore
- Si deve rinormalizzare
- Non produce normali equispaziate!



SLERP
(Spherical Linear Interpolation)

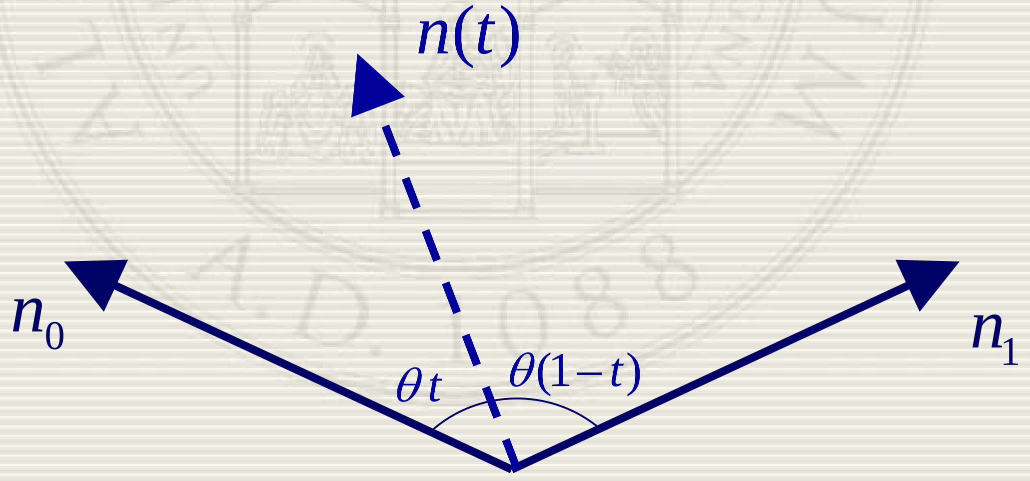


SLERP
(Spherical Linear Interpolation)



SLERP
(Spherical Linear Interpolation)

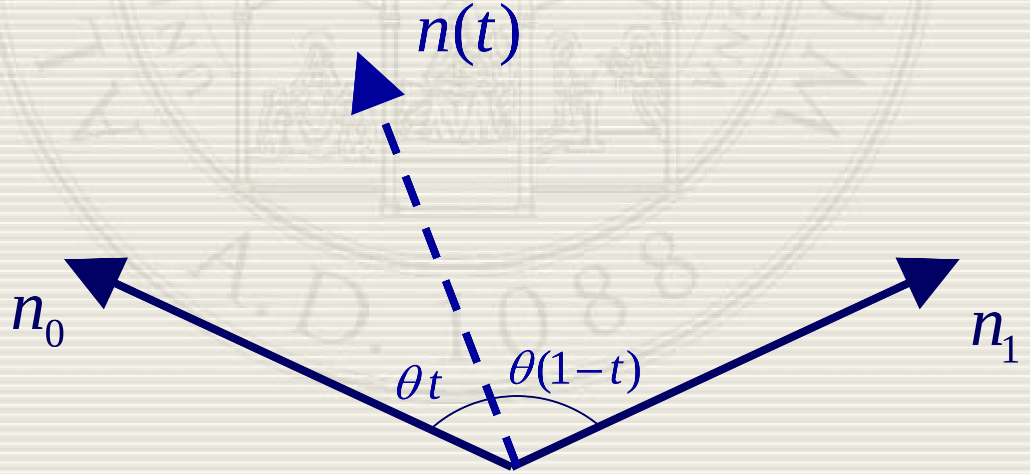
$$|n_0| = |n_1| = |n(t)|$$



SLERP
(Spherical Linear Interpolation)

$$|\, n_0 \,| = |\, n_1 \,| = |\, n(t) \,|$$

$$n(t) = \alpha\, n_0 + \beta\, n_1$$



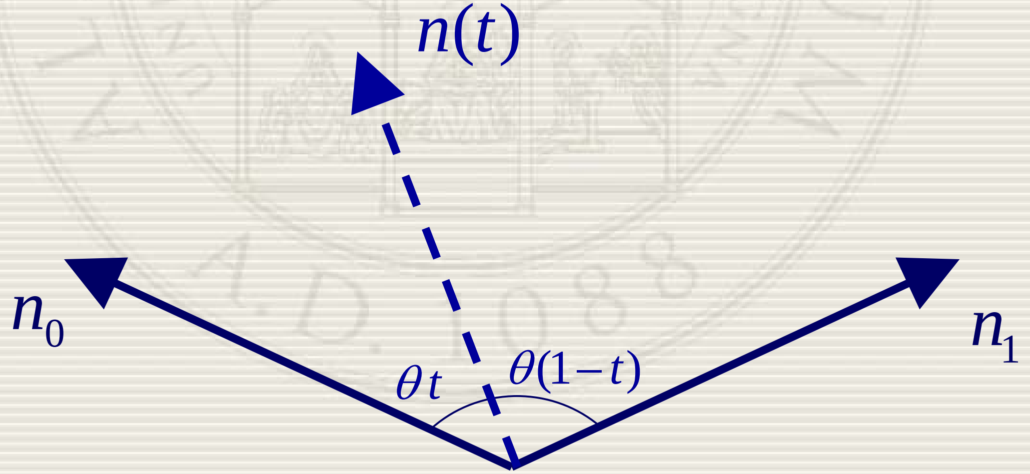
SLERP
(Spherical Linear Interpolation)

$$|n_0| = |n_1| = |n(t)|$$

$$n(t) = \alpha\, n_0 + \beta\, n_1$$

$$n_0 \times n(t) = n_0 \times (\alpha\, n_0 + \beta\, n_1)$$

$$n_1 \times n(t) = n_1 \times (\alpha\, n_0 + \beta\, n_1)$$



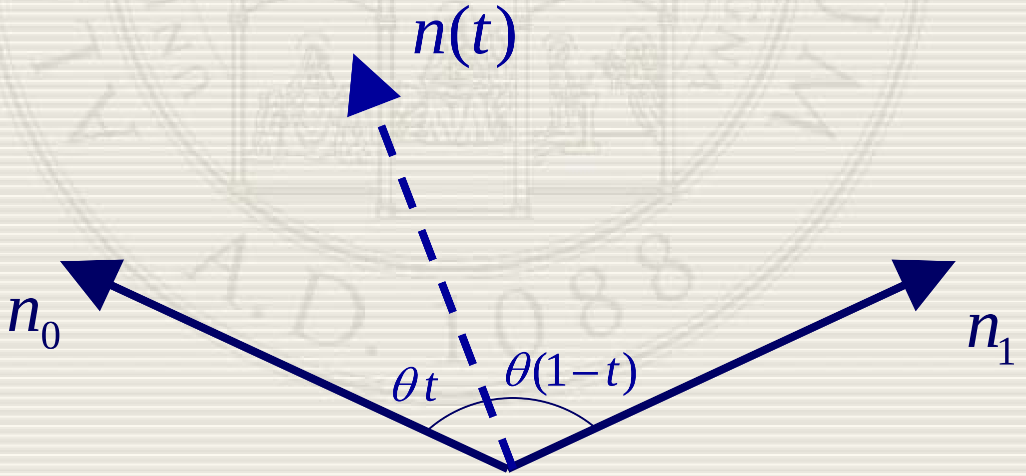
SLERP
(Spherical Linear Interpolation)

$$|n_0| = |n_1| = |n(t)|$$

$$n(t) = \alpha\, n_0 + \beta\, n_1$$

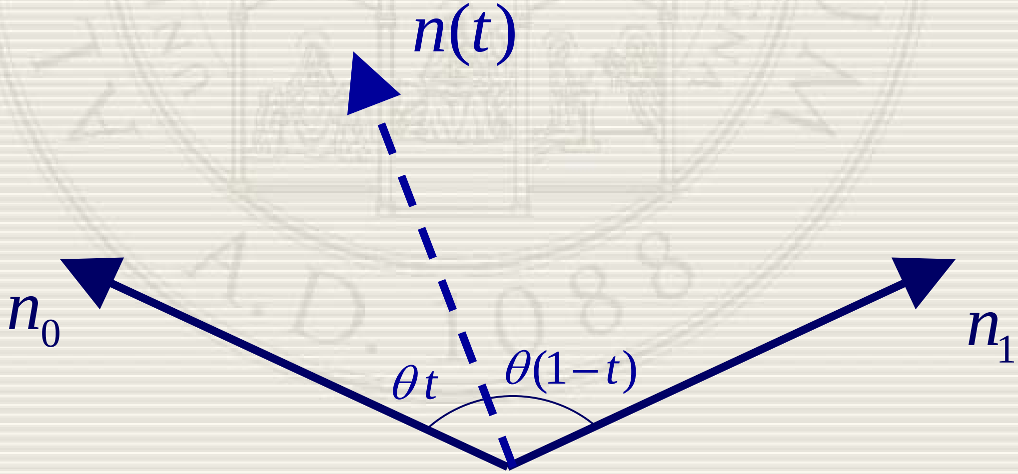
$$n_0 \times n(t) = \beta(n_0 \times n_1)$$

$$n_1 \times n(t) = \alpha(n_1 \times n_0)$$



SLERP
(Spherical Linear Interpolation)

$$|n_0| = |n_1| = |n(t)|$$
$$n(t) = \alpha\, n_0 + \beta\, n_1$$
$$|n_0 \times n(t)| = \beta\, |n_0 \times n_1|$$
$$|n_1 \times n(t)| = \alpha\, |n_1 \times n_0|$$



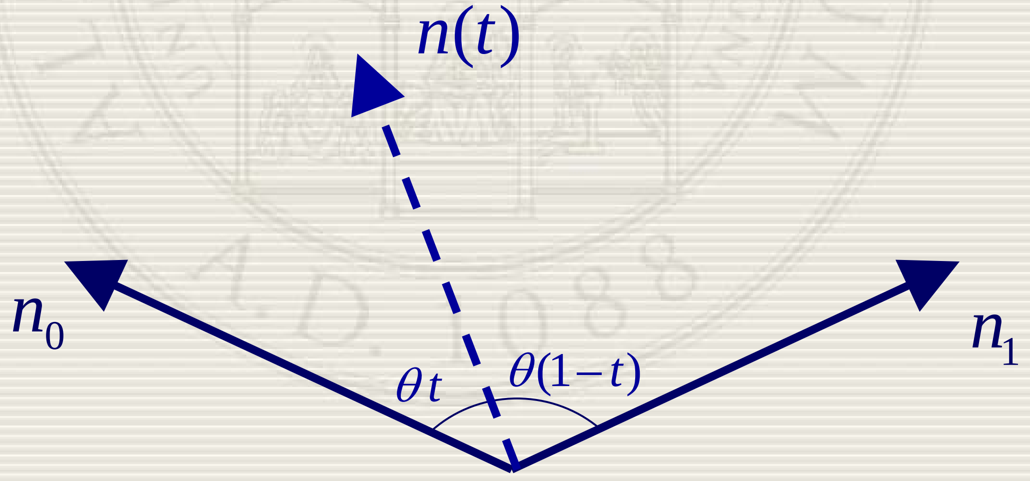
SLERP (Spherical Linear Interpolation)

$$|n_0| = |n_1| = |n(t)|$$

$$n(t) = \alpha\, n_0 + \beta\, n_1$$

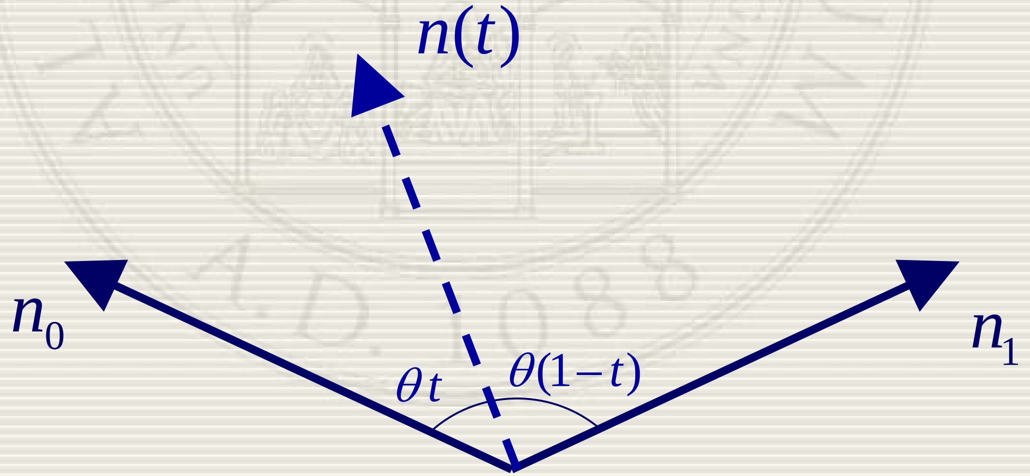
$$|n_0|\,|n(t)|\sin(\theta t) = \beta\, |n_0|\,|n_1|\sin(\theta)$$

$$|n_1|\,|n(t)|\sin(\theta(1-t)) = \alpha\, |n_1|\,|n_0|\sin(\theta)$$



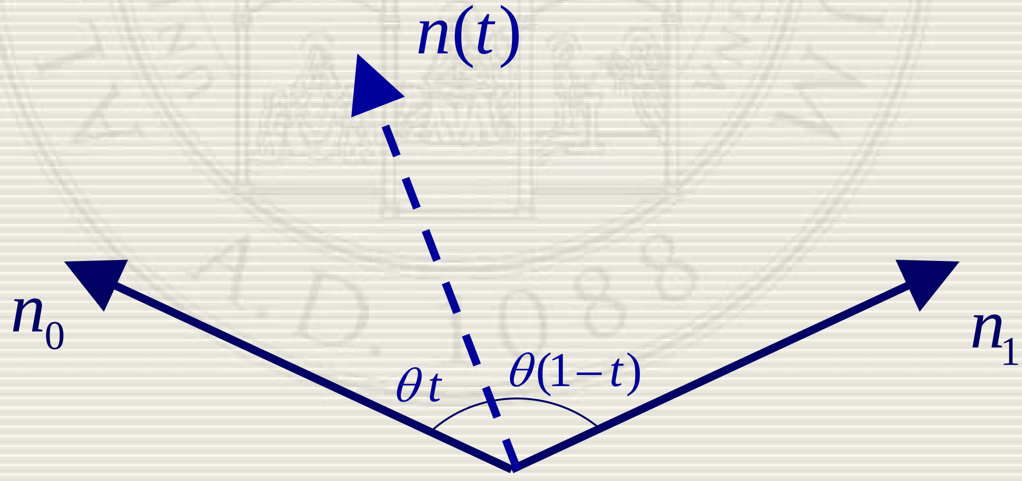
SLERP
(Spherical Linear Interpolation)

$$|n_0| = |n_1| = |n(t)|$$
$$n(t) = \alpha\, n_0 + \beta\, n_1$$
$$\sin(\theta\, t) = \beta \sin(\theta)$$
$$\sin(\theta(1-t)) = \alpha \sin(\theta)$$



SLERP
(Spherical Linear Interpolation)

$$n(t) = \frac{\sin(\theta(1-t))n_0 + \sin(\theta t)n_1}{\sin(\theta)}$$

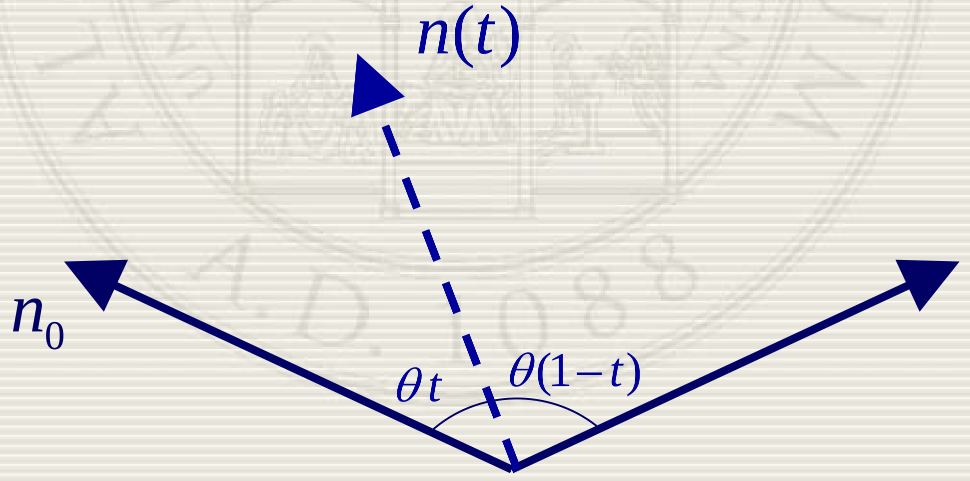


SLERP
(Spherical Linear Interpolation)

$$n(t) = \frac{\sin(\theta(1-t))n_0 + \sin(\theta t)n_1}{\sin(\theta)}$$

$$n(0) = n_0$$

$$n(1) = n_1$$



SLERP
(Spherical Linear Interpolation)

$$n(t) = \frac{\sin(\theta(1-t))n_0 + \sin(\theta t)n_1}{\sin(\theta)}$$