Materiale Lambertiano (o puramente diffusivo)

- ✓ E' definito un materiale in cui la BRDF è costante

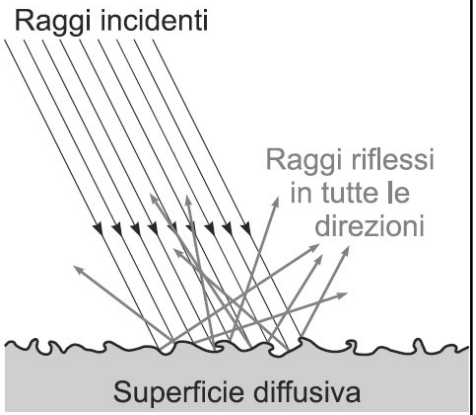
$$f_r(\vec{\omega}_i, \vec{\omega}_r) = const$$

- ✓ E' anche materiale (puramente) diffusivo
  - ⇒ Una superficie di questo materiale riflette la luce da cui è raggiunta *diffondendola uniformemente*
- ✓ La costante è detta l'**albedo** (del materiale): il rapporto fra la luce incidente e la luce riflessa
  - ⇒ da 0, materiale perfettamente nero, assorbe tutta la luce
  - ⇒ a 1, materiale perfettamente bianco, diffonde tutta la luce

15

Materiali «lambertiano» (o diffusivi) reali

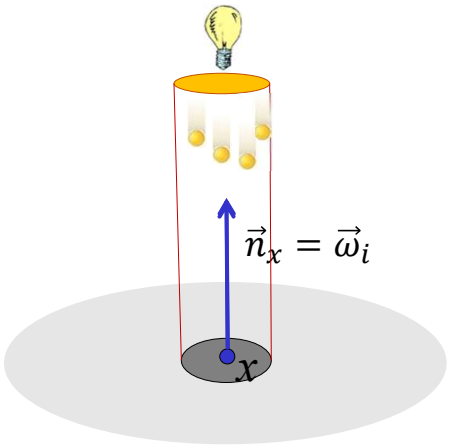
- ✓ Si verificano quando…
  - ⇒ a livello microscopico, la superficie presenta micro-sfaccettature disposte in modo molto irregolare, caotico
- ✓ Nota:
  - ⇒ le BRDF dei materiali riflettono (anche) le configurazioni microscopiche (e altro, come le proprietà elettromagnetiche)



16

Sottoproblema

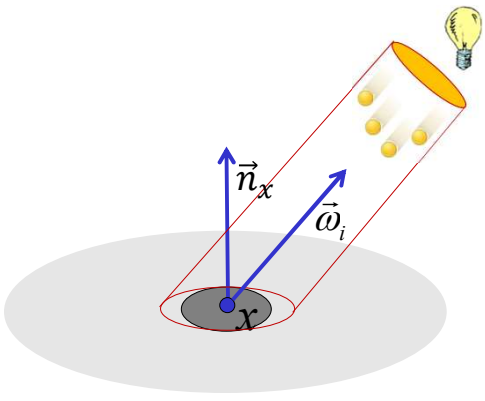
✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale $\vec{n}_x$?



17

Sottoproblema

✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale $\vec{n}_x$?



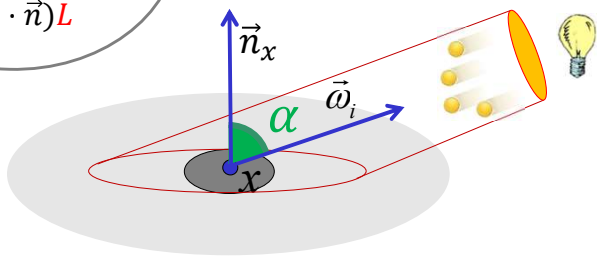
19

Sottoproblema

✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale $\vec{n}_x$?

risposta:
("legge del coseno")
$\cos(\alpha)L$
$=$
$(\vec{\omega}_i \cdot \vec{n})L$

Johann
Heinrich
Lambert
1728 - 1777

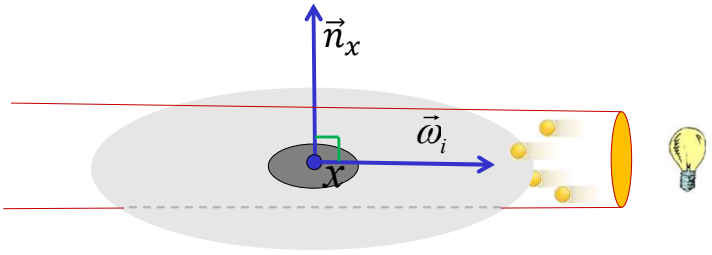


23

Sottoproblema

✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale $\vec{n}_x$?

Caso luce perfettamente radente:
$(\vec{\omega}_i \cdot \vec{n}) = 0$
infatti nessun fotone colpisce l'intorno

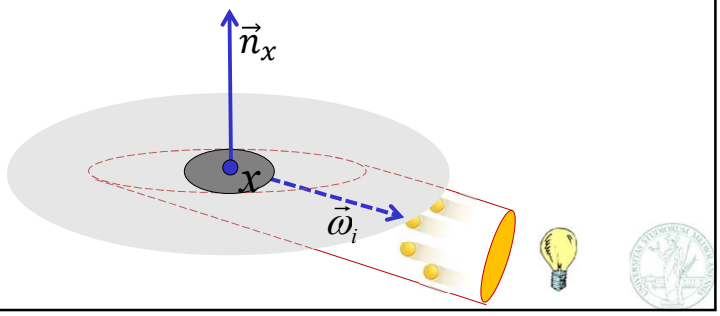


24

Sottoproblema

✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale $\vec{n}_x$?

Caso luce da dietro:
$(\vec{\omega}_i \cdot \vec{n}) < 0$
allora la risposta è 0
(non certo un numero negativo)
perché l'intorno si trova nella sua stessa ombra)



25

Equazione di lighting per materiali lambertiani (o puramente diffusivi)

✓ L'equazione di lighting risultante è

$$P = (\vec{n}_x \cdot \vec{\omega}_i)\ D\ L$$

- P: Luminosità finale del pixel
- $(\vec{n}_x \cdot \vec{\omega}_i)$: Prodotto dot, ma zero se negativo
- $\vec{n}_x$: mormale della sup
- $\vec{\omega}_i$: direzione della luce (vettore unitario che va verso la luce)
- L: Inteisità della luce
- D: albedo del materiale (quanto è chiaro o scuro il suo aspetto)

26

Interpretazione intuitiva del lighting nel modello di materiale Lambertiano

- ✓ Anche se il modello di materiale diffusivo (o di Lambert) è basato sulla fisica reale, possiamo anche darne una interpretazione geometrica intuitiva
- ✓ Il fattore $(\vec{n}_x \cdot \vec{\omega}_i)$, il coseno dell'angolo fra i due vettori, è anche una misura della similarità fra la direzione $\vec{n}_x$ normale alla superficie e la direzione $\vec{\omega}_i$ «verso la luce»
- ✓ Quindi, la legge del coseno di Lambert quindi dice: «tanto più la superficie è orientata verso la luce (cioè, tanto più la sua normale è simile alla direzione di luce), maggiormente chiara ci apparirà.»

27

Lighting lambertiano… a colori

- ✓ L'equazione di lighting vista (che è basata sul fenomeni fisici ben compresi) sarebbe utilizzabile per immagini in bianco e nero
- ✓ Per passare ad immagini a colori, un modo semplice è applichiare la stessa equazione separatamente sui i canali rosso, verde, e blu
  - ⇒ Questa è una approssimazione brutale, ma molto utilizzata

"albedo", (per così dire) ma solo per quell che riguarda la luce rossa

canale rosso del pixel risultante

$$P_R = (\vec{n}_x \cdot \vec{\omega}_i)\ D_R\ L_R$$

$$P_G = (\vec{n}_x \cdot \vec{\omega}_i)\ D_G\ L_G$$

$$P_B = (\vec{n}_x \cdot \vec{\omega}_i)\ D_B\ L_B$$

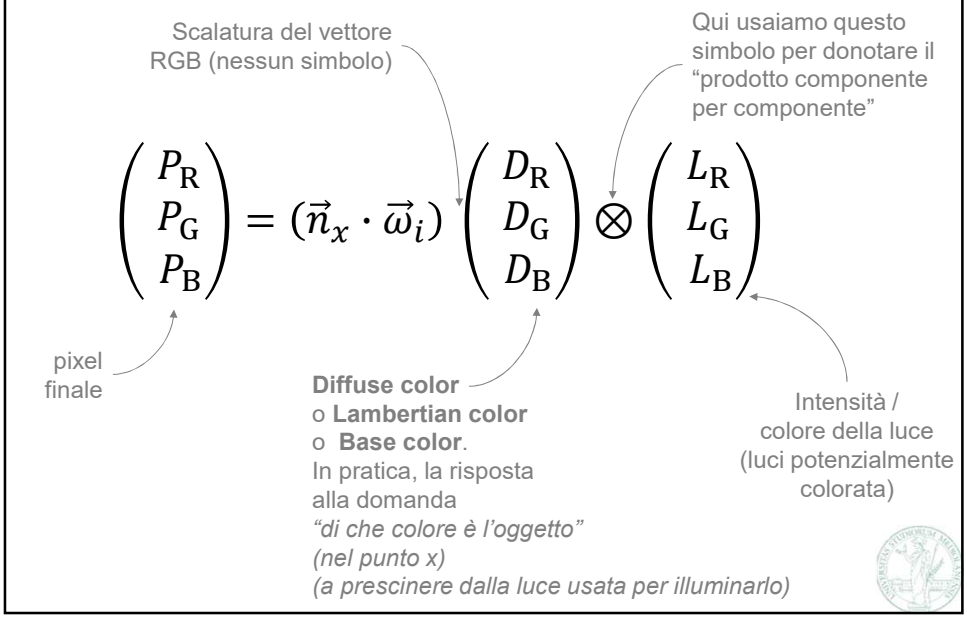
Intensità della luce sul canale rosso (quanta luce rossa emette)

Idem, per il verde

Idem, per il blue

28

Lighting… a colori: una riscrittura



29

Modelli (o equazioni) di lighting

- ✓ L'equazione di lighting vista è molto semplice
  - ⇒ consiste esclusivamente di una componente di riflessione diffusiva
  - ⇒ però è physically based – il fenomeno ottico modellato è accuratamente riprodotto (eccetto che per la tricromia RGB)
- ✓ E' un esempio di modello di illuminazione locale
- ✓ Il questo modello:
  - ⇒ il «materiale» è descritto unicamente dal base color (detto anche diffuse color -- o albedo se in bianco e nero)
  - ⇒ L'unico aspetto geometrico che è rilevante è la normale del punto illuminato. In particolare, la direzione di osservazione non conta.
  - ⇒ Per questo, il modello è un modello «view independent».
  - ⇒ Il chiaroscuro non dipende dalla direzione di osservatore: non abbiamo riflessi (infatti materiali come gesso ne sono privi)
- ✓ Vedremo in seguito altri modelli, in grado di modellare riflessi

30

Lighting con più sorgenti di luce

- ✓ Abbiamo supposto di avere una sola sorgente di luce
  - ⇒ di direzione $\vec{\omega}$ e intensità (L_R, L_G, L_B)
- ✓ Aumentare il numero di luci è molto semplice: basta calcolare e sommare i contributi di ciascuna luce
- ✓ Esempio con due luci

$$\begin{pmatrix} P_R \\ P_G \\ P_B \end{pmatrix} = (\vec{n}_x \cdot \vec{\omega}_1) \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_{1,R} \\ L_{1,G} \\ L_{1,B} \end{pmatrix} + (\vec{n}_x \cdot \vec{\omega}_2) \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_{2,R} \\ L_{2,G} \\ L_{2,B} \end{pmatrix}$$

pixel finale

Direzione luce 1

Intensità / colore della luce 1

Direzione luce 2

Intensità / colore della luce 2

31

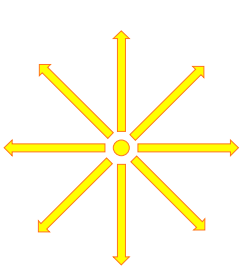
Lighting con più sorgenti di luce: osservazioni

- ✓ L'aggiuta di ogni luce comporta un aggravio di computazione del lighting
  - ⇒ Dobbiamo computare un altro termine, su ogni pixel della scena (o ogni elemento da calcorare)
- ✓ In assenza di qualsiasi fonte di luce, il lighting degli oggetti ha zero addendi, e il colore risultante è (0,0,0)
  - ⇒ Com'è sensato che sia: al buio, tutti gli oggetti sono neri!
- ✓ All'aggiungere fonti di luce, il lighting si satura verso il bianco, dato che il pixel finale al massimo è (1,1,1)
  - ⇒ Quindi se vogliamo aggiungere molte luci, ciascuna di loro deve essere più fievole

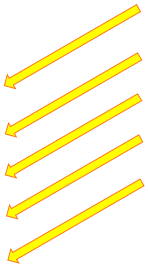
32

Modellazione delle sorgenti di luce

- ✓ Possiamo modellare due tipi di sorgenti di luce:
  - ⇒ "direzionali" : modella fonti di luce molto distanti, per es, il sole
  - ⇒ "posizionali" : modella fonti di luci vicine, per es, lampadine



LUCE POSIZIONALE



LUCE DIREZIONALE

33

Luci posizionali e direzionali

- ✓ Una luce direzionale ha una direzione di luce incidente $\vec{\omega}$ costante su tutta la scena
  - ⇒ Il vettore unitario $\vec{\omega}$ definisce la geometria della luce
- ✓ Una luce posizionale è invece descritta da una posizione nello spazio P_L
  - ⇒ la direzione della luce è diversa per ogni punto da illuminare
  - ⇒ un punto in posizione Q da riceverà la luce dalla direzione:

$$\vec{\omega} = \frac{(P_L - Q)}{\|P_L - Q\|}$$

- ✓ In ogni caso, sia il vettore $\vec{\omega}$ che il punto Q sono tipicamente espressi in spazio mondo

34

Luci posizionali e direzionali

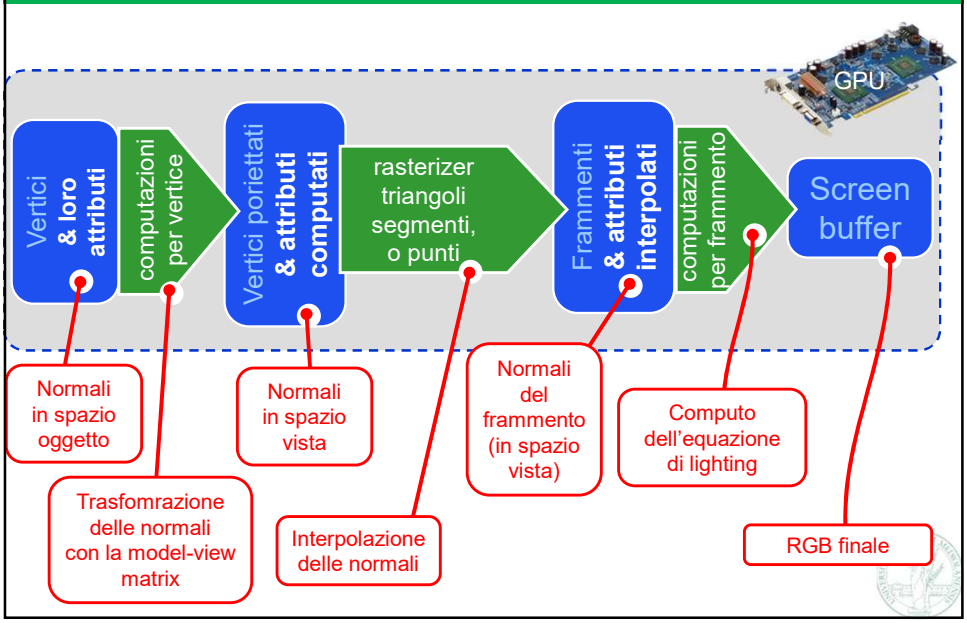
- ✓ Una luce direzionale ha un'intesità/colore (L_R, L_G, L_B) costante su tutta la scena
- ✓ In una luce posizionale, l'intensità/colore può essere attenuato, per ogni punto illuminato Q, da un fattore di affievolimento che varia in funzione della sua distanza dalla luce $d = \|P_L - Q\|$
  - ⇒ La formula fisica è

$$f_{affievolimento} = \frac{1}{\|P_L - Q\|^2}$$

  - ⇒ a volte si usa un esponete minore di 2 per rendere artificialmente meno decisa l'attenuazione della luce con la distanza

35

Calcolo del lighting nel pipeline di rendering



36

Calcolo del lighting nel pipeline di rendering

- ✓ il computo del lighting tipicamente viene svolto nella fase per frammento
  - ⇒ Tecnica nota come “per-pixel lighting”, calcolo dell’illuminazione per ogni pixel
  - ⇒ (ma esistono anche varianti e ottimizzazioni)
  - ⇒ Dunque, il lighting avviene dopo la trasformazione dei vertici
  - ⇒ L’equazione di lighting è calcolata in un fragment-shader
  - ⇒ Usando three.js, stiamo usando un fragment-shader fornito dalla libreria per questo scopo
- ✓ I punti e vettori usati nell’equazione di lighting devono, ovviamente, essere tutti espressi nello stesso spazio
  - ⇒ direzione luce, direzione vista, normale, posizione luce…
  - ⇒ spesso si sceglie lo spazio vista oppure lo spazio mondo: a questo scopo, le normali della mesh (originalmente espresse, ovviamente, in spazio oggetto) devono essere trasformate in spazio vista o mondo, nella fase di trasformazione per vertice, usando la matrice model-view o di model
  - ⇒ (nota: NON in spazio clip: la matrice di proiezione prospettica non può essere usata per trasformare vettori, ma solo punti)
  - ⇒ Le direzioni/posizioni della luce vanno espresse nello stesso spazio

37

Sperimentiamo il lighting di materiali Lambertiani in three.js (note 1/2)

- ✓ 1: costruiamo il materiale e usiamolo per la mesh

```
var mioMat = new THREE.MeshLambertMaterial();
var miaMesh = new THREE.Mesh( buffers, mioMat );
```

- ✓ 2: settiamo il colore base (diffuse color) del materiale ad esempio, ad un blu chiaro

```
mioMat.color.set(0xFF8855);
```

- ✓ 3: creiamo le luci e aggiungiamole alla scena ad esempio, una luce direzionale

```
var luce0 = new THREE.DirectionalLight();
miaScena.add(luce0);
```

- ✓ 4: settiamo la direzione (chiamata, per confondere, “position”) e l’intensità-colore della luce (chiamato color), ad esempio, una luce bianca (il default)

```
luce0.color.set(0xFFFFFF);
luce0.position.set(0,3,-2); // viene normalizzato
```

38

Sperimentiamo il lighting di materiali Lambertiani in three.js (note 2/2)

- ✓ 5: possiamo aggiungere una seconda, ad esempio, questa una luce posizionale

```
var luce1 = new THREE.PositionalLight();
miaScena.add(luce1);
```

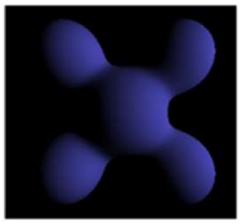
- ✓ 6: settiamone la posizione (per es, x=2,y=1.5,z=5) e l'intensità-colore della luce (chiamato color), ad esempio, una luce rossastra

```
luce1.color.set(0xDD6666);
luce1.position.set(2,1.5,5);
```

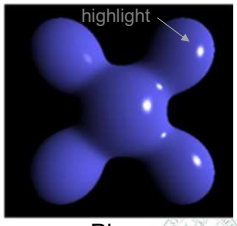
39

Limiti del modello di Lambert

- ✓ Un materiale Lambertiano riproduce solo l'aspetto materiali opachi (dull) privi di riflessi luminosi
  - ⇒ ma, almeno, lo fa in modo fisicamente corretto
- ✓ Vediamo il modello di materiale di Phong, capace di riprodurre anche l'aspetto di superfici lucide,
  - ⇒ come quelle bagnate, levigate, incerate, etc,
  - ⇒ aggiungendo riflessi lucidi (specular reflections, o highlights)
- ✓ Il modello di Phong,
  - ⇒ non è ispirato da considerazioni fisiche
  - ⇒ non è confermato da misurazion radiometriche.
  - ⇒ è basato su una costruzione geometrica molto approssimata ma efficace



Lambert

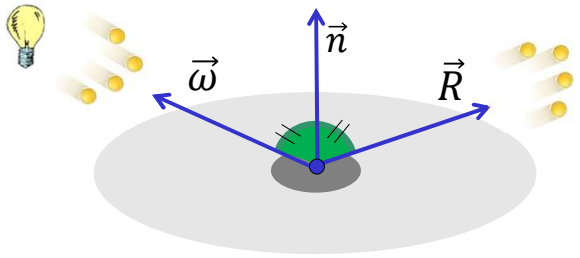


Phong

40

Modello di illuminazione di Phong: la spiegazione spaziale intuitiva

- ✓ In presenza di un materiale molto levigato / liscio / lucido, i fotoni rimbalzeranno sulla superficie (riflessione) in modo simile a quello di una pallina da ping-pong su un tavolo



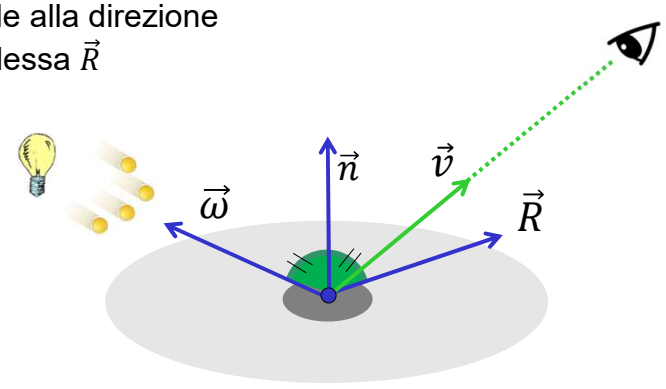
Nota:
I tre vettori mostrati sono planari

- ✓ Intuizione: la BRDF di un materiale lucido manderà un numero maggiore dei fotoni provenienti dalla direzione $\vec{\omega}$ nella direzione riflessa $\vec{R}$, che non nelle altre direzioni
  - ⇒ a differenza di quella di un materiale diffusivo / lambertiano, che li rimbalza in qualsiasi direzione con uguale probabilità

41

Modello di illuminazione di Phong: componente speculare: spiegazione intuitiva

- ✓ Il modello di Phong somma alla componente di riflesso **diffusiva** (quella vista) una componente di riflesso **speculare**, che sarà tanto maggiore tanto più la direzione di vista $\vec{v}$ sarà simile alla direzione di luce riflessa $\vec{R}$

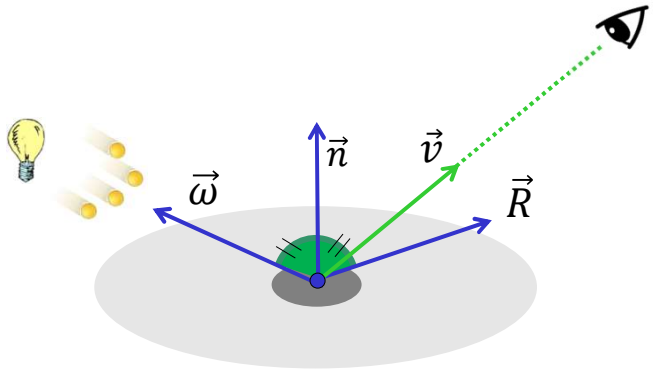


42

Modello di illuminazione di Phong: componente speculare: calcolo preliminare

calcolo di $\vec{R}$ a partire da $\vec{\omega}$ e $\vec{n}$ (vedi lez algebra punti e vettori)

$$\vec{R} = -\vec{\omega} + 2\,(\vec{\omega} \cdot \vec{n}\,)\,\vec{n}$$



43

Modello di illuminazione di Phong: componente speculare: calcolo

Prodotto dot, ma 0 se negative. Misura la similarità fra i due vettori

Intesità della luce (RGB)

$$\left(\vec{v} \cdot \vec{R}\right)^{S} \begin{pmatrix} H_{\mathrm{R}} \\ H_{\mathrm{G}} \\ H_{\mathrm{B}} \end{pmatrix} \otimes \begin{pmatrix} L_{\mathrm{R}} \\ L_{\mathrm{G}} \\ L_{\mathrm{B}} \end{pmatrix}$$

intesità e colore RGB degli highlights, scelto a piacere per un dato materiale

Specular exponent (o shininess)

44

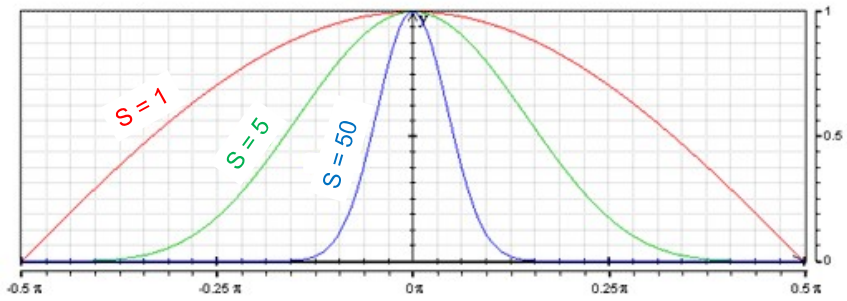
Modello di illuminazione di Phong: shininess (o specular exponent)

- ✓ La formula $(\vec{v} \cdot \vec{R})$ (fra 1 e 0) produrebbe riflessi troppo grandi, perché decresce troppo lentamente al discostarsi di $\vec{v}$ da $\vec{R}$
- ✓ Per ovviare, basta elevare ad una potenza $S > 1$
  - ⇒ Elevando un numero minore di 1 ad un a S > 1 , lo si avvicina allo 0
- ✓ Il fattore S è detto specular exponent, o shininess, o glossiness.
- ✓ Tanto maggiore vale S, tanto più simile dovranno essere $\vec{v}$ e $\vec{R}$ per avere un riflesso visibile, cioè tanto più piccoli e concentrati saranno gli highlights.
- ✓ Nuovamente, questo passaggio non ha nessun preciso significato fisico, ed è solo un espediente si usa per ottenere l'effetto desiderato

45

Componente *riflessione speculare*

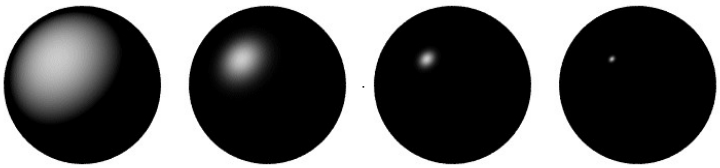
- ✓ Elevando il coefficiente ad una potenza S >1, si ottengono highlights più piccoli e concentrati



angolo fra $\vec{v}$ e $\vec{R}$

46

Effetto dello specular exponent" nella *riflessione speculare*



$S = 2$ $S = 5$ $S = 10$ $S = 100$

47

Modello di illuminazione di Phong: riassunto degli input del calcolo

Input:

- ✓ la direzione di luce $\vec{\omega}$
- ✓ la direzione di vista $\vec{v}$

è un modello *view-dependent* !

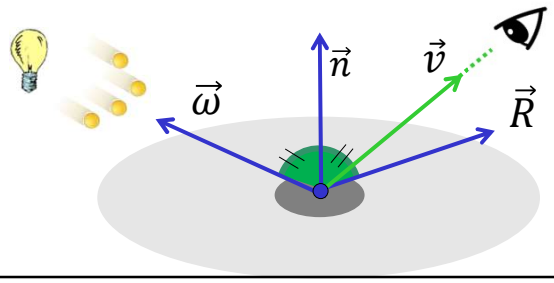
- ✓ Intensità-colore della luce $\begin{pmatrix} L_R \\ L_G \\ L_B \end{pmatrix}$
- ✓ Base-color o diffuse-color $\begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix}$
- ✓ Intensità-colore degli highlights $\begin{pmatrix} H_R \\ H_G \\ H_B \end{pmatrix}$
- ✓ Esponente di specularità, o shininess, o glossiness S

Nuove caratteristiche dei materiali che concorrono a differenziare un materiale da un altro

48

Calcolo della componente speculare con Blinn-Phong 1/2

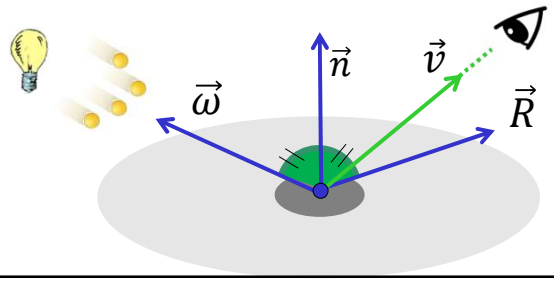
- ✓ Blinn-Phong è una variante (quasi equivalente) per calcolare la similarità fra $\vec{v}$ e $\vec{R}$ senza dover calcolare esplicitamente $\vec{R}$
- ✓ Il concetto: $\vec{v}$ coincide con $\vec{R}$ **sse** la normale $\vec{n}$ è esattamente a metà strada dei due
- ✓ Invece che calcolare $\vec{R}$ e poi chiedersi quanto sia simile a $\vec{v}$, la formula Blinn-Phong calcola la direzione m



49

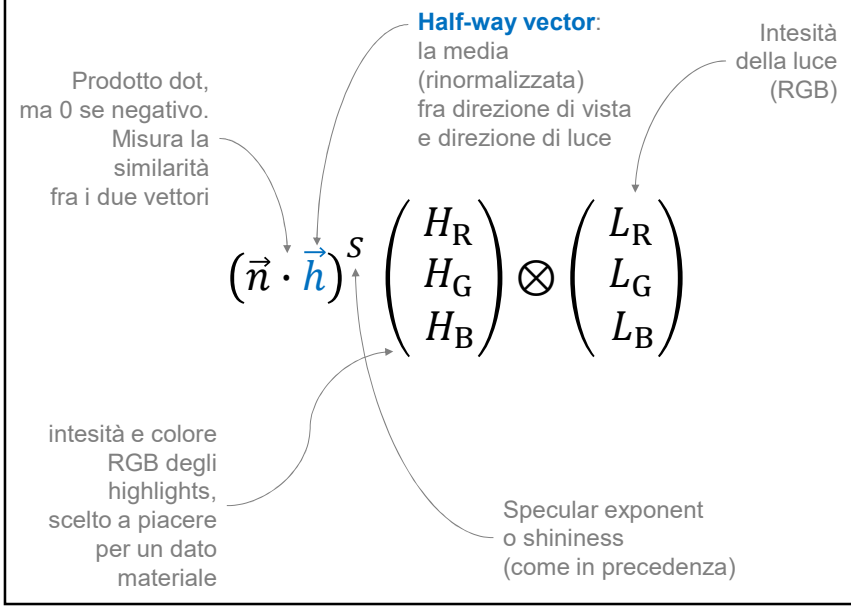
Calcolo della componente speculare con Blinn-Phong 1/2

- ✓ Invece che calcolare $\vec{R}$ (a partire da $\vec{\omega}$ e $\vec{n}$) e poi chiedersi quanto sia simile a $\vec{v}$, la formula Blinn-Phong calcola la direzione media fra $\vec{\omega}$ e $\vec{v}$, detta half-way vector, e si chiede quanto sia simile ad $\vec{n}$
- ✓ Questo produce un risultato molto simile ma meno oneroso calcolare



50

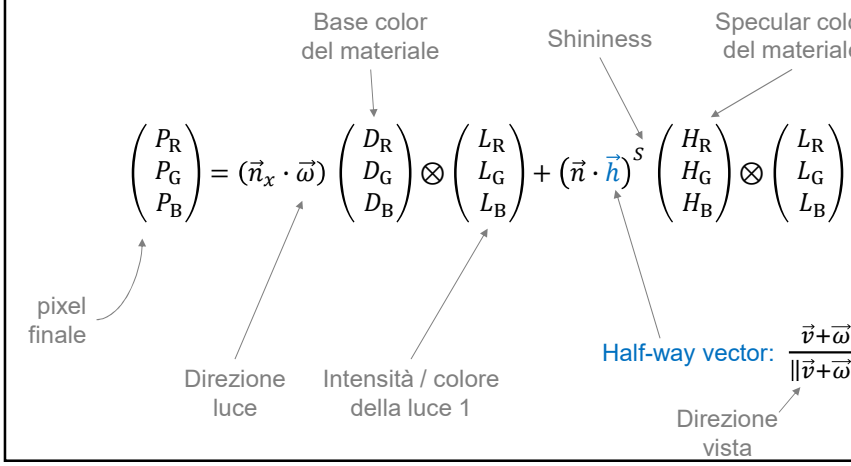
Modello di illuminazione di Blinn-Phong: componente speculare: calcolo



51

Modello di illuminaizione di Phong: in totale

- ✓ Il modello di illuminazione di Phong aggiunge, (per ogni luce) la componente diffusiva (di Lambert) alla componente Speculare (qui, calcolata con Blinn-Phong)



52

Sperimentiamo il lighting di materiali Phong con three.js (note)

- ✓ 1: costruiamo il materiale di tipo Phong

```
var mioMat = new THREE.MeshPhongMaterial();
var miaMesh = new THREE.Mesh( buffers, mioMat );
```

- ✓ 2: come prima, il colore base (diffuse color) del materiale ad esempio, ad un blu chiaro

```
mioMat.color.set(0xFF8855);
```

- ✓ 3: il materiale di Phong però prevede anche un colore speculare (colore degli highlights)

```
mioMat.specular.set(0x222222);
```

e un coefficiente speculare, qui chiamato shininess

```
mioMat.shininess = 100;
```

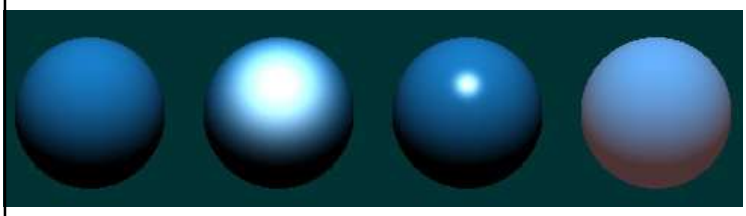
53

Componente ambiente

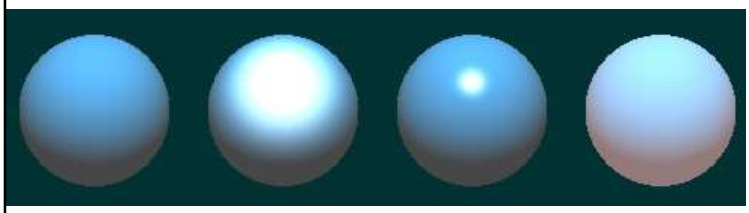
- ✓ Un problema coi modelli di illuminazione visti è che le superfici non raggiunte da alcuna luce (ad esempio, i cui prodotti dot fra normale e dir luce o half-way siano zero) sono perfettamente neri.
- ✓ In una scena realistica, questo è un caso molto raro. Nella maggior parte dei casi, si può assumere che un minimo di luce (detta «ambientale») *raggiunga qualsiasi superficie da qualsiasi direzione*
- ✓ Per simulare questo stato di cose, si può semplicemente aggiungere al modello di illuminazione una componente ambient, una piccola costante additiva (in R, G e B) data dal prodotto dell'intesità di una apposita fonte luce di tipo «ambientale» (una luce pervasiva, globale, che non ha ne' una direzione, ne' posizione) e il colore di una apposita compoente «ambient» del materiale

55

Componente *ambiente*



senza



con

56

Modello di illuminaizione di Phong con luce ambiente

✓ Modello di illuminazione risultante

ripetuto per ciascuna luce direzionale e posizionale

per la luce ambient

termine diffusivo

termine speculare

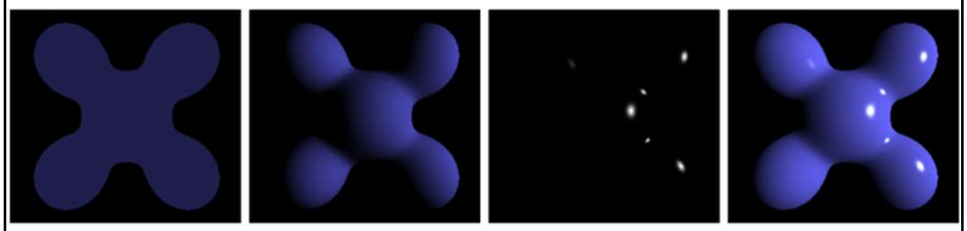
termine ambient

$$\begin{pmatrix} P_R \\ P_G \\ P_B \end{pmatrix} = (\vec{n}_x \cdot \vec{\omega}) \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_R \\ L_G \\ L_B \end{pmatrix} + (\vec{n} \cdot \vec{h})^s \begin{pmatrix} H_R \\ H_G \\ H_B \end{pmatrix} \otimes \begin{pmatrix} L_R \\ L_G \\ L_B \end{pmatrix} + \begin{pmatrix} A_R \\ A_G \\ A_B \end{pmatrix} \otimes \begin{pmatrix} L_R \\ L_G \\ L_B \end{pmatrix}$$

57

Modello di phong esempio (con una luce e componente ambient)

Il risultato finale è ottenuto sommando tutti i termini



ambiente + diffusivo + speculare (highlights) = totale

58

Aggiungiamo la componente ambient in three.js (note)

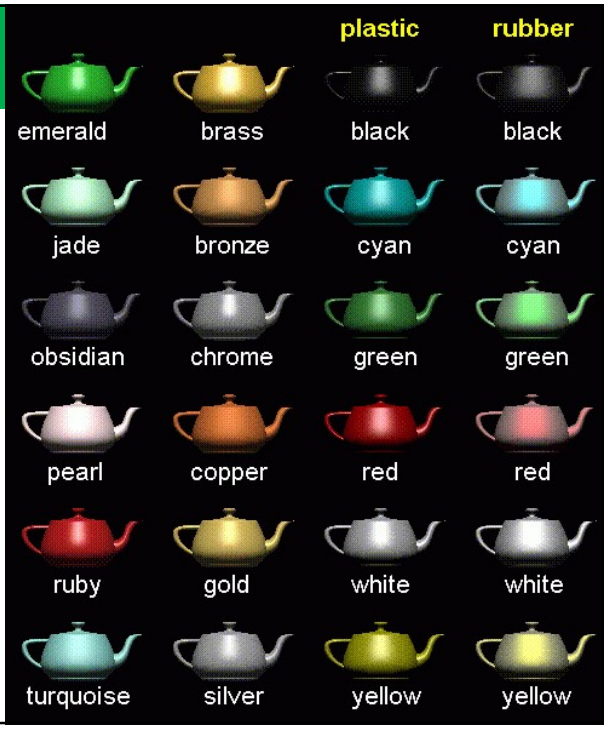
✓ creiamo una sorgente di luce ambient
(tenue, e incolore: R = G = B = 0x10 – cioè 16 su 255)
e aggiungiamola alla scena

```
var luceA = new THREE.AmbientLight(0x040404);
miaScena.add(luceA);
```

59

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 0.458561 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



60

Definizione dei materiali: sommario e alcune note sui limiti

- ✓ Questa lezione ha mostrato una visione semplificata del problema del lighting
- ✓ Il modello visto (di Phong) descrive un materiale come:
  - ⇒ Un colore diffusivo
  - ⇒ Un colore ambient, di solito corrispondente al colore diffusivo
  - ⇒ Un colore speculare
  - ⇒ Uno specular coefficient (da 1 a 100)
- ✓ La scelta di un qualsiasi materiale (velluto, oro, etc) si effettua spaziando fra i possibili valori da assegnare a queste variabili
- ✓ Questo modo di definire i materiali non lascia pochi gradi di libertà, e consente di riprodurre solo materiali in una gamma ristretta di possiblità
- ✓ Inoltre molte delle scelte possibili risultano in materiali dall'aspetto «sbagliato» ritenuti, secondo gli standard attuali della CG, poco realistici
- ✓ Anche la descrizione dell'ambiente di illuminazione vista (cioè, una collezione di luci direzionali o posizionali) è una semplificazione eccessiva, che non consente di riprodurre fedelmente ambiente di illuminazione realistici
- ✓ Modelli di materiale più complessi, che richiedono però computi di lighting più onerosi, consentono di definire materiali in modo più completo, e rendono anche più facile e intuitiva la scelta di parametri atti a riprodurre materiali reali
  - ⇒ ad esempio, da parte di artisti addetti al task, detti material artists

61