

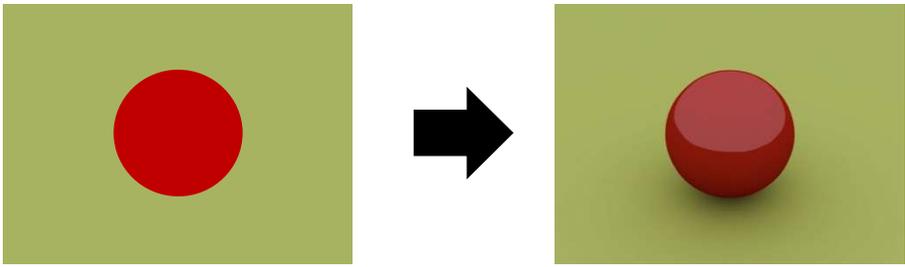
Marco Tarini - Computer Graphics 2022/2023
Università degli Studi di Milano

Lighting



1

Lighting



2

Lighting: intro

- ✓ Dopo la transform, la parte principale del rendering
 - ✓ Ci proponiamo di simulare una serie di fenomeni reali complessi di interazione fra luce (una radiazione elettromagnetica) e scena
 - ✓ Il task concreto: determinare la luce
 - ⇒ quanta luce
 - ⇒ di che colore
- arriva
- ⇒ da ogni punto della scena
 - ⇒ alla camera



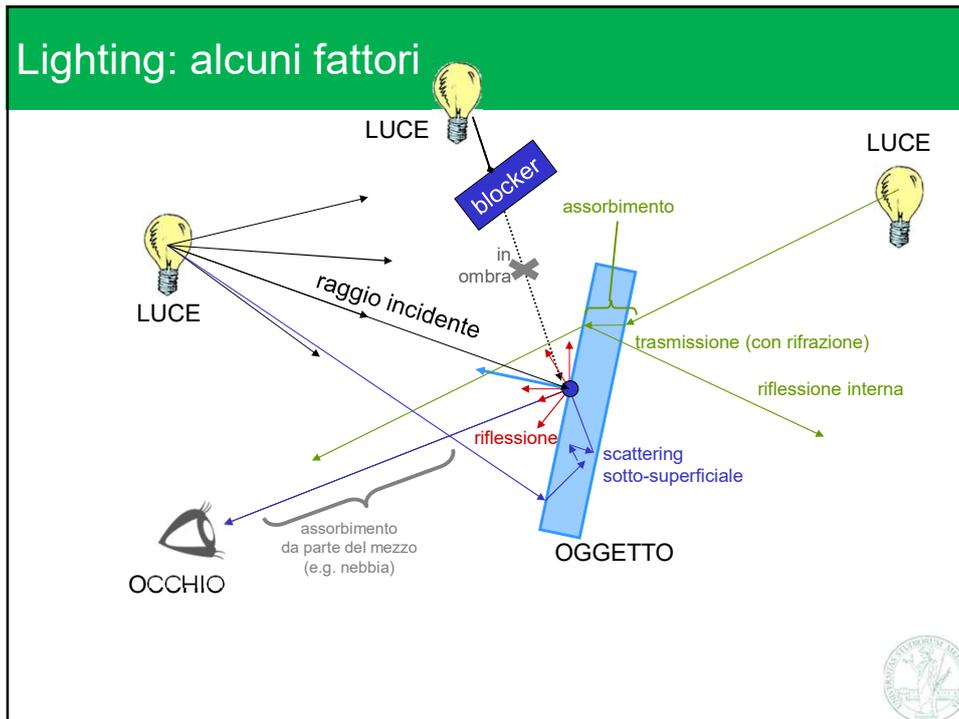
3

Lighting: intro

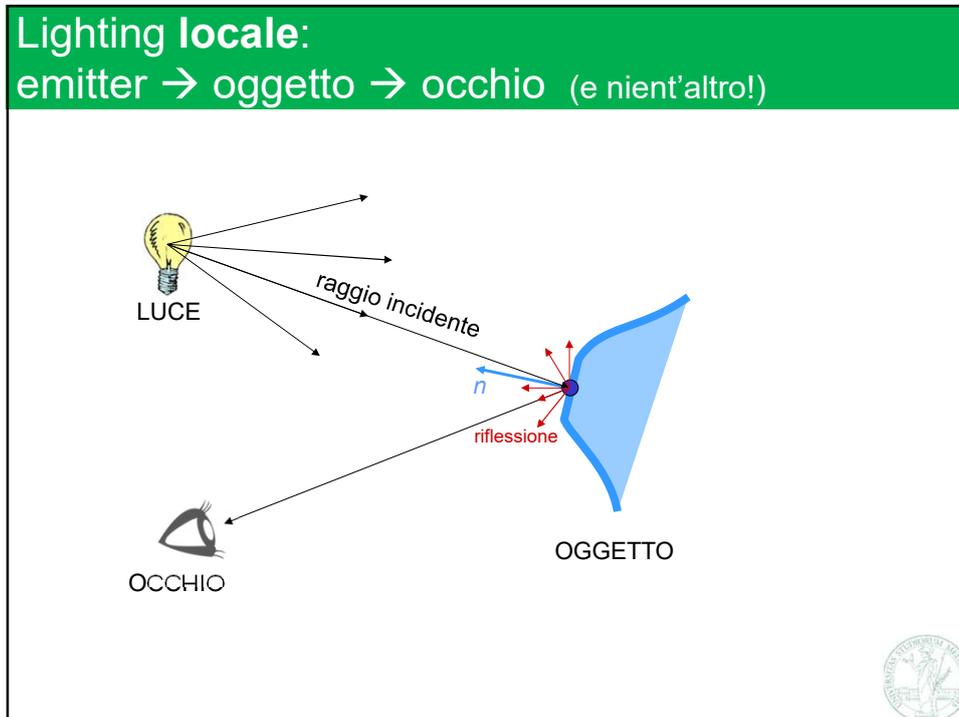
- ✓ Output del lighting:
 - ⇒ colore percepito per un dato punto
- ✓ Input del lighting:
 - ⇒ caratteristiche dell'oggetto illuminato ("materiale")
 - es: carta bagnata, plastica ruvida, ...
 - o banalmente, «di che colore è» - vedi poi
 - ⇒ caratteristiche della luce che illumina
 - che intensità / colore hanno
 - ⇒ e geometria della scena
 - soprattutto, gli orientamenti delle superfici, cioè le normali
 - anche, la posizione degli oggetti, dell'osservatore, della camera...



4



5



8

Materiali in CG

- ✓ In CG, il “materiale” è la una modellazione che descrive in che modo una data superficie reagisce alla luce e comprende cose come
 - ⇒ Quanto è chiaro oppure scuro
 - ⇒ Di che colore si presenta
 - ⇒ Quanto è lucido oppure opaco (privo di riflessi lucidi, o highlight)
 - ⇒ Quanto è trasparente, semitrasparente, traslucido, o permeabile alla luce
 - ⇒ Se e quanto il materiale è “cangiante”
 - ⇒ Se è fosforescente o cmq emette luce propria (oltre a riflettere quella di cui è investito)
- ✓ (nota che in alcune librerie ad alto livello, come three.js o unity, il materiale è una classe che descrive quanto sopra, ma anche alcuni settaggi di rendering come quelli relative a back-face culling o depth test, e le tessiture etc)

9

Materiali in CG, per lighting locale

- ✓ Per quello che riguarda il lighting locale, il materiale è esaustivamente descritto da una funzione che descrive in che modo la superficie riflette la luce che la raggiunge direttamente
- ✓ Cioè quanta della luce che investe un pezzetto di superficie viene riflessa dalla superficie in ogni possibile direzione
- ✓ Tecnicamente, questa funzione prende in input:
 - ⇒ Una possibile direzione di provenienza della luce $\vec{\omega}_i$ (vettore unitario)
 - ⇒ Una possibile direzione di uscita della luce $\vec{\omega}_r$ (vettore unitario)e restituisce in output
 - ⇒ Quanta della luce proveniente da $\vec{\omega}_i$ verrà mandata in $\vec{\omega}_r$,
 - ⇒ Per ogni una distribuzione di probabilità
- ✓ Questa funzione viene detta Bidirectional (due direzioni) Radiance Distribution Function

10

Descrizione di un materiale nel lighting locale: la BRDF

$f_r =$

fotone =

$f_r(\vec{\omega}_i, \vec{\omega}_r) =$ su 100 che arrivano a dalla dir $\vec{\omega}_i$, quante verranno da lui rimbalzate proprio nella dir $\vec{\omega}_r$? (al variare di $\vec{\omega}_i$ e $\vec{\omega}_r$)

Ogni materiale, la sua BRDF:

- metallo cromato
- raso
- legno
- stoffa
- gesso
- carta
- specchio...

Funzione di 4 dimensioni!
(una direzione = 2 dimensioni)

11

La BRDF

- ✓ Questa funzione viene detta BRDF
 - ⇒ Bidirectional (perché prende due direzioni)
 - ⇒ Reflectance (riflettanza, abilità di riflettere la luce)
 - ⇒ Distribution (perché è una distribuzione di riflettanza)
 - ⇒ Function
- ✓ La BRDF descrive in modo esaustivo un materiale (per quello che riguarda il lighting locale)
- ✓ Ad esempio, materiali come

⇒ Legno	⇒ Oro	⇒ Plastica levigata
⇒ Raso	⇒ Acciaio	⇒ Plastica ruvida
⇒ Stoffa	⇒ Argento	⇒ Vetro
⇒ Pietra	⇒ Cobalto	⇒ Diamante
⇒ Ferro arrugginito	⇒ Seta	⇒ Catarifrangenti
⇒ Velluto	⇒ (superficie de) Acqua	⇒ etc

sono caratterizzati da BRDF molto diverse

12

La BRDF

- ✓ Per ora occupiamoci della più semplice classe delle BRDF: quelle costanti
 - ⇒ dette puramente diffuse, o Lambertiane (da Lambert, v. dopo)
- ✓ I materiali che esibiscono questa BRDF sono detti diffusivi o Lambertiani
 - ⇒ Esempio di materiali che si avvicinano a questo: gesso, carta ruvida
 - ⇒ (in realtà è sempre solo un'approssimazione)
- ✓ Un materiale Lambertiano riflette la stessa percentuale di luce in tutte le direzioni possibili di uscita, indipendentemente da quale sia la direzione di arrivo della luce
 - ⇒ Quindi è privo di riflessi, esibisce lo stesso colore visto da tutti gli angoli
 - ⇒ Il lighting è *view-independent*: non dipende dalla direzione di luce
- ✓ Dobbiamo occuparci di rispondere ad alcune domande:
 - ⇒ Quanto è questa percentuale di luce costante? (dipende dal materiale)
 - ⇒ Quanta luce raggiunge il punto sulla superficie? (dipende dalle luci)

13

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

14

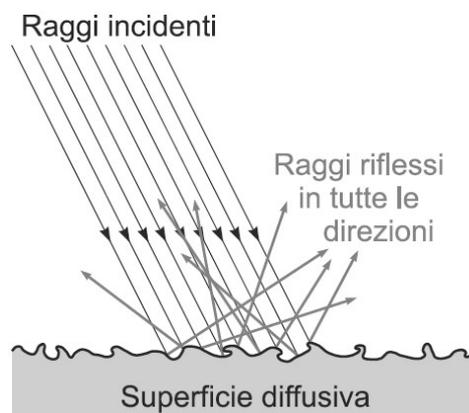
Materiali «lambertiano» (o diffusivi) reali

✓ Si verificano quando...

⇒ a livello **microscopico**,
la superficie presenta
micro-sfaccettature
disposte in modo
molto irregolare, caotico

✓ Nota:

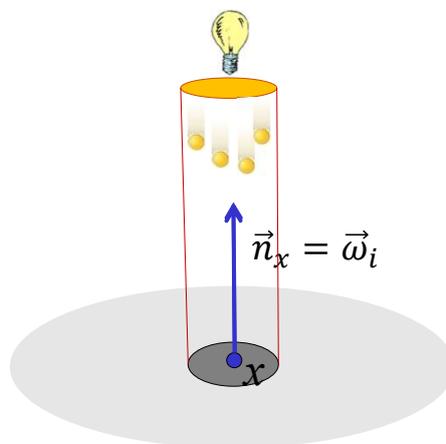
⇒ le BRDF dei materiali
dipendono dalle
conformazioni
microscopiche della sup
(e altro, come le proprietà
elettromagnetiche)



15

Sottoproblema

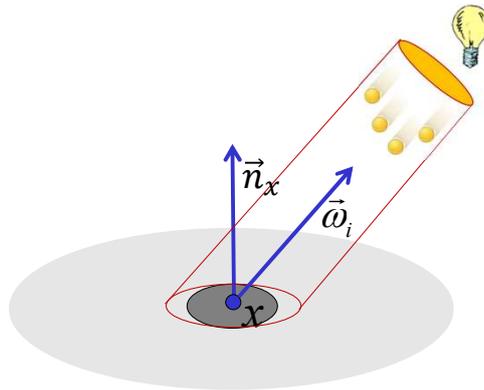
✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens,
quante ne riceve un intorno di x con normale \vec{n}_x ?



16

Sottoproblema

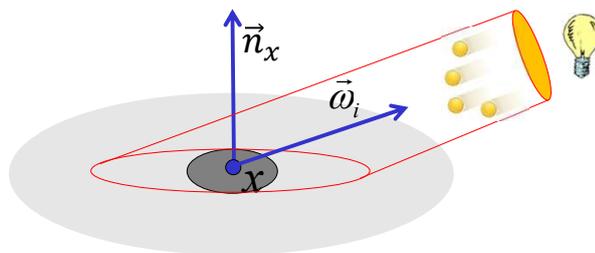
- ✓ Se dalla dir $\vec{\omega}_i$ arrivano L lumens, quante ne riceve un intorno di x con normale \vec{n}_x ?



17

Sottoproblema

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



18

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

20

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

22

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)

23

Equazione di lighting per materiali lambertiani (o puramente diffusivi)

✓ L'equazione di lighting risultante è

Prodotto dot,
ma zero se negativo

Luminosità
finale
del pixel

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

Intensità della luce

albedo del
materiale
(quanto è chiaro o
scuro il suo
aspetto)

normale della sup

direzione della luce
(vettore unitario che
va verso la luce)

24

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.»



25

Lighting lambertiano... a colori

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

canale rosso del pixel risultante

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

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

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

“albedo”, (per così dire) ma solo per quella che riguarda la luce rossa

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

Idem, per il verde

Idem, per il blue



26

Lighting... a colori: una riscrittura

Scalatura del vettore
RGB (nessun simbolo)

$$\begin{pmatrix} P_R \\ P_G \\ P_B \end{pmatrix} = (\vec{n}_x \cdot \vec{\omega}_i) \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_R \\ L_G \\ L_B \end{pmatrix}$$

pixel
finale

Qui usiamo questo
simbolo per donotare il
"prodotto componente
per componente"

Diffuse color
 o **Lambertian color**
 o **Base color**.
 In pratica, la risposta
 alla domanda
 "di che colore è l'oggetto"
 (nel punto x)
 (a prescindere dalla luce usata per illuminarlo)

Intensità /
colore della luce
(luci potenzialmente
colorata)



27

Modelli (o equazioni) di lighting

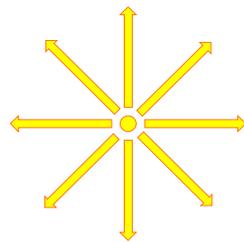
- ✓ L'equazione di lighting vista è molto semplice
 - ⇒ consiste esclusivamente di una componente di riflessione diffusa
 - ⇒ però è physically based – il fenomeno ottico modellato è accuratamente riprodotto (eccetto che per la tricromia RGB)
- ✓ E' un esempio di modello di illuminazione locale
- ✓ Il 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



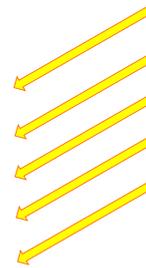
28

Modellazione della sorgente di luce

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



LUCE POSIZIONALE



LUCE DIREZIONALE



29

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



30

Luci posizionali e direzionali

- ✓ Una luce **direzionale** ha un'intensità/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

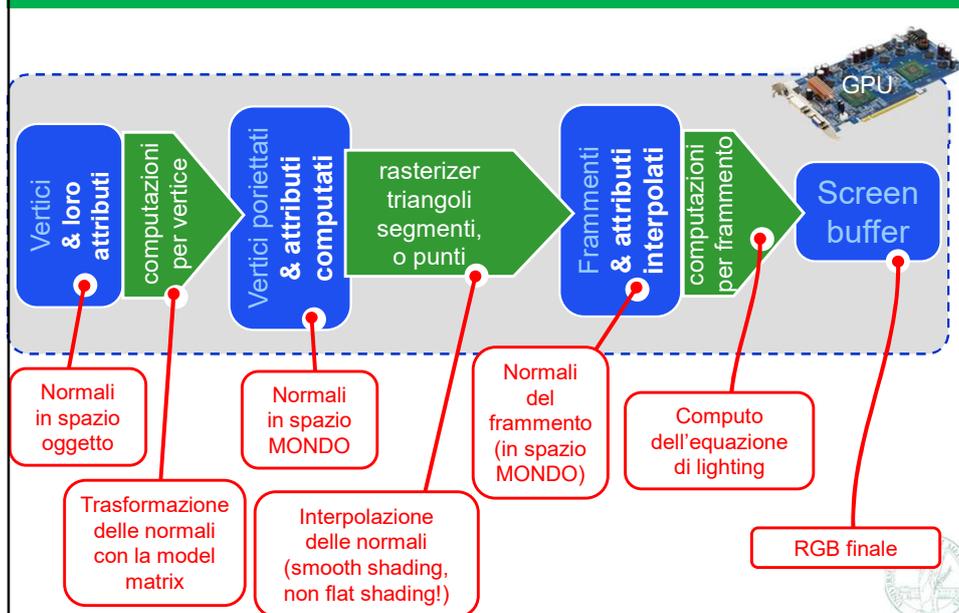
$$f_{affievolimen} = \frac{1}{\|P_L - Q\|^2}$$

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



31

Calcolo del lighting nel pipeline di rendering



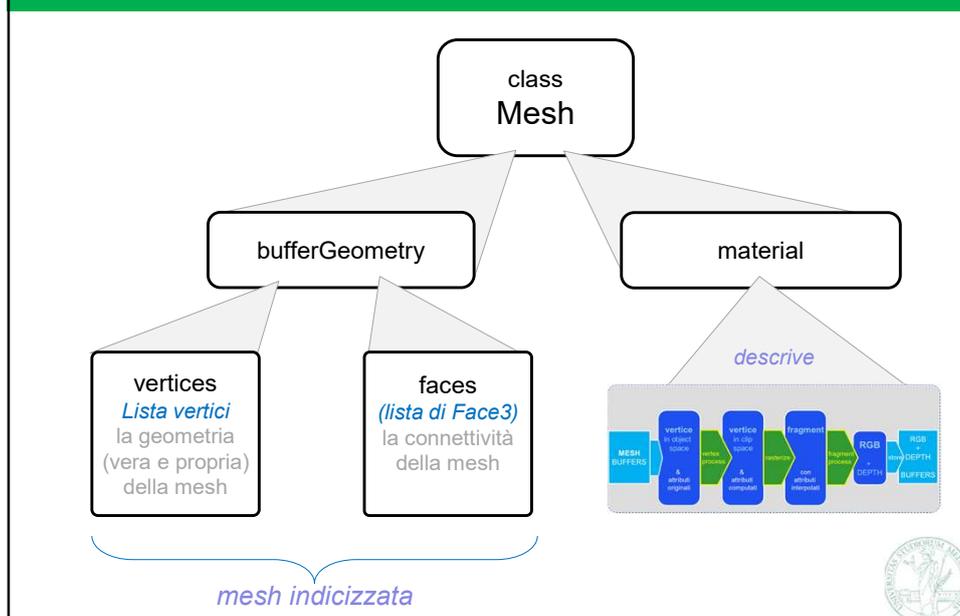
32

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 mondo:
 - a questo scopo, le normali della mesh (originalmente espresse, ovviamente, in spazio oggetto) devono essere trasformate in spazio mondo, nella fase di trasformazione per vertice, usando la matrice model
 - ⇒ Anche lo spazio vista può essere usato (quale matrice occorre usare, allora)?
 - ⇒ (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

33

Struttura per le Mesh su Three.js



34

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 );
```

 (questo istruisce three.js ad usare il pipeline come descritto sopra)
- ✓ 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 miaLuce = new THREE.DirectionalLight();
miaScena.add(miaLuce);
```
- ✓ 4: settiamo la direzione (chiamata, per confondere, "position") e l'intensità-colore della luce (chiamato color), ad esempio, una luce bianca (il default)


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

Vedere lez04.html per un esempio (righe marcate con "NUOVO")



36

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

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


```
var luce2 = new THREE.PointLight();
miaScena.add(luce2);
```
- ✓ 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


```
luce2.color.set(0xDD6666);
luce2.position.set(2,1.5,5);
```



37

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

38

Lighting con più sorgenti di luce: osservazioni

- ✓ L'aggiunta di ogni luce comporta un aggravio di computazione del lighting
 ⇒ Dobbiamo computare un altro termine,
 su ogni pixel della scena (o ogni elemento da calcolare)
- ✓ 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 fiavole

39

Componente ambiente

- ✓ Un problema del modello di illuminazione visto è che le superfici non raggiunte da alcuna luce (ad esempio, i cui prodotti dot fra normale e dir luce o siano zero o negativi) 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 ambiente, una piccola costante additiva (in R, G e B) data dal prodotto dell'intensità di una apposita fonte luce di tipo «ambientale» (una luce pervasiva, globale, che non ha né una direzione, né posizione)



40

Componente *ambiente*



senza



con



41

Modello di illuminazione Lambertiano con luce ambiente

✓ Equazione di lighting risultante

termine diffusivo, ripetuto per ciascuna luce direzionale o posizionale i

legge del coseno

$$\begin{pmatrix} P_R \\ P_G \\ P_B \end{pmatrix} = \left(\sum_i (\vec{n}_x \cdot \vec{\omega}_i) \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_{iR} \\ L_{iG} \\ L_{iB} \end{pmatrix} \right) + \begin{pmatrix} D_R \\ D_G \\ D_B \end{pmatrix} \otimes \begin{pmatrix} L_{AR} \\ L_{AG} \\ L_{AB} \end{pmatrix}$$

termine ambient

↑
pixel finale

↑
direzione luce i

↑
intensità & colore della luce i

↑
intensità & colore della luce ambiente

Base color (caratteristica del materiale)
 (in realtà, si possono usare anche due colori diversi per ambient e diffusivo)

42

Aggiungiamo la componente ambient in three.js (note)

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

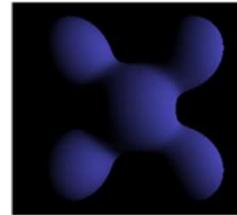
```
var luceAmbiente = new THREE.AmbientLight(0x101010);
miaScena.add(luceAmbiente);
```

Vedere lez04.html per un esempio (righe marcate con "NUOVO")

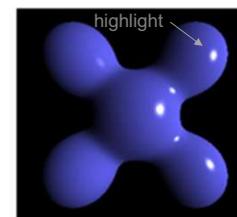
43

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
- ✓ La prossima lezione, 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 (con specular reflections, o highlights)
- ✓ Il modello di Phong, tuttavia...
 - ⇒ non è ispirato da considerazioni fisiche
 - ⇒ non è confermato da misurazione radiometriche.
 - ⇒ è basato su una costruzione geometrica molto approssimata ma efficace



Lambert

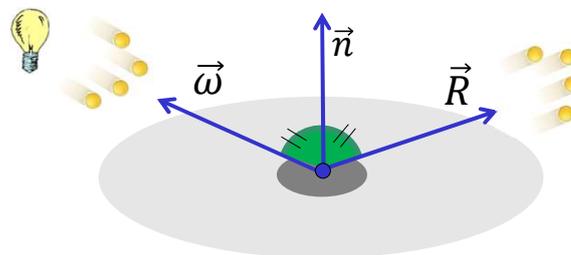


Phong

44

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

45