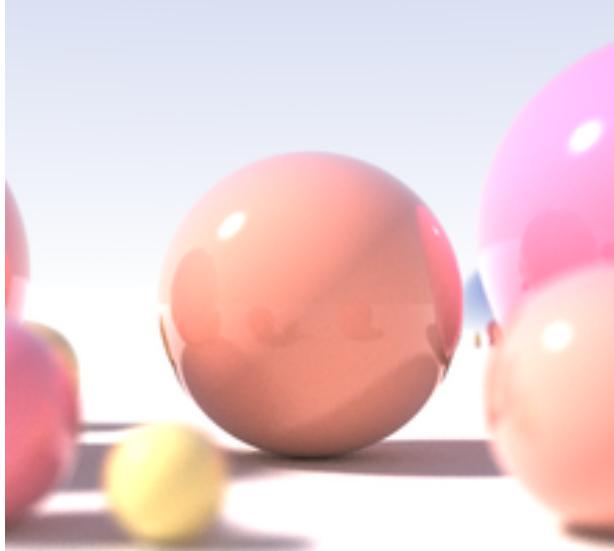


Gràfics i Visualització de Dades



T2b: RayTracing

Anna Puig

Índex

2.1. Introducció: algorisme principal

2.2. Càlcul de Raig Primari

2.3. Càlcul del color:

2.3.1. Interseccions amb objectes

2.3.2. Materials i Llums

2.3.2.1. Interacció de les llums i els materials

2.3.2.2. Llums

2.3.2.3. Materials

2.3.2.4. El model de Blinn-Phong

2.3.2.5. Textures

2.4. Ombres

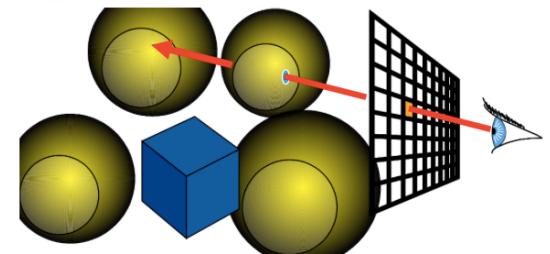
2.5. Reflexions i transparències

NOTA: Aquestes transparències es corresponen a les explicacions del tema 5 del llibre de referència bàsica

[Angel2011] Edward Angel, Dave Shreiner, **Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, 6/E**, ISBN-10: 0132545233. ISBN-13: 9780132545235, Addison-Wesley, 2011

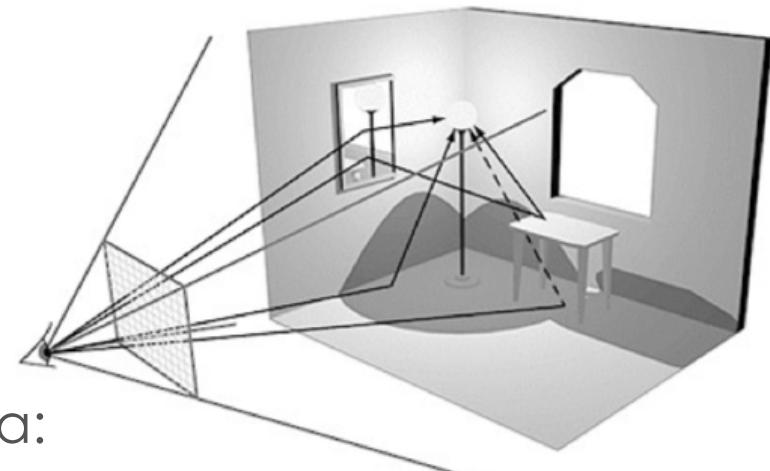
2.1. Introducció

- **Raycast()** // generar una imatge per cada pixel x,y
 $r = \text{getRay}(x, y)$
 color(pixel) = **PixelRay(ray)**
 fper
- **PixelRay(ray)** // llançar un raig, retorna RGB
 hitInfo= **hit(ray)** // intersection (ray, tmin, tmax, hitinfo)
 si object_point retorna **Shade(hitInfo, lookFrom)**
 sino retorna Background_Color // o Intensitat ambient global
- **hit(ray)**
 per cada objecte en l'escena
 hit(ray, surface) //intersection (ray, tmin, tmax, hitinfo)
 retorna el punt més proper a l'observador (+normal, +material)
- **Shade(hitInfo, lookFrom)** // retorna la il·luminació en el point
 retorna color



2.3.2. Materials i Llums

Òptica geomètrica (la llum es considera ona i partícula)

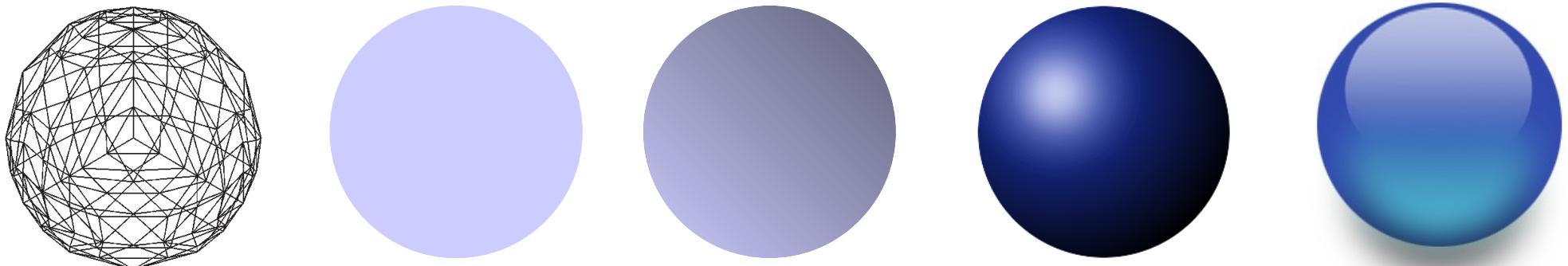


Algunes regles de l'òptica geomètrica:

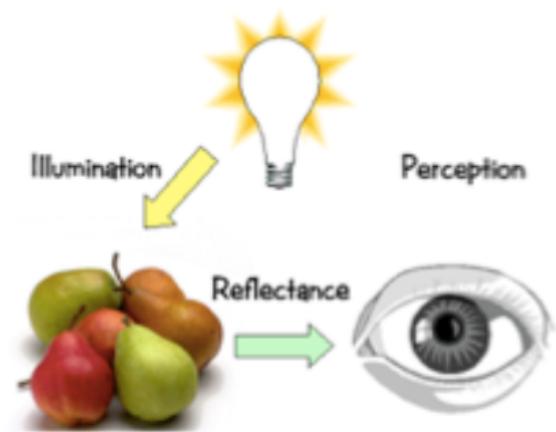
- La llum és un flux de fotons amb longitud d'ona que anomenarem "**rajos de llum**"
- Els rajos de llum viatgen en **trajectòries rectilínies**
- Els rajos de llum no interfereixen entre ells quan es creuen
- Els rajos de llum obeyeixen les **lleis de reflexió i refracció**
- Els rajos de llum viatgen des de les fonts de llum fins els objectes i la física és invariant en sentit invers (**principi de reciprocitat**)

2.3.2. Materials i Llums

Suposeu que construïm una esfera definida amb molts polígons:



- Les **interaccions** entre les **llums** i el **material** de l'esfera fa que cada punt que veu l'observador tingui un color diferent.
- Per això és necessari considerar:
 - Les propietats de les **llums**
 - Les propietats del **material** de l'objecte
 - La posició de **l'observador**
 - L'orientació de la **superficie** de l'objecte



2.3.2. Materials i Llums

La **llum** es pot considerar com petits paquets d'energia (**fotons**)

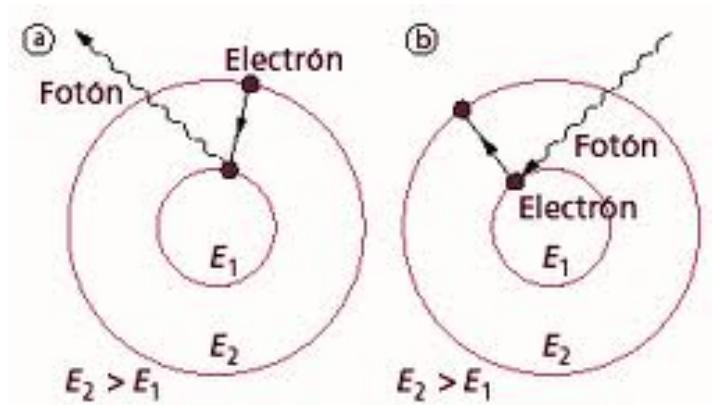
- Cada àtom té un nucli i electrons circulant a diferents òrbites
- Quan un electró salta d'òrbita alta a una baixa s'emet energia (**Llei de Plank**) -> **emet llum**
- Quan un fotó col·lisiona amb l'àtom i un electró puja d'òrbita -> s'**absorbeix** la llum.



<http://www.overclock.net/art-graphics/251218-kerkythea-shaded-lightsource-test.html>

La **longitud d'ona (color)** de la llum emesa o absorbida correspon al canvi d'orbital.

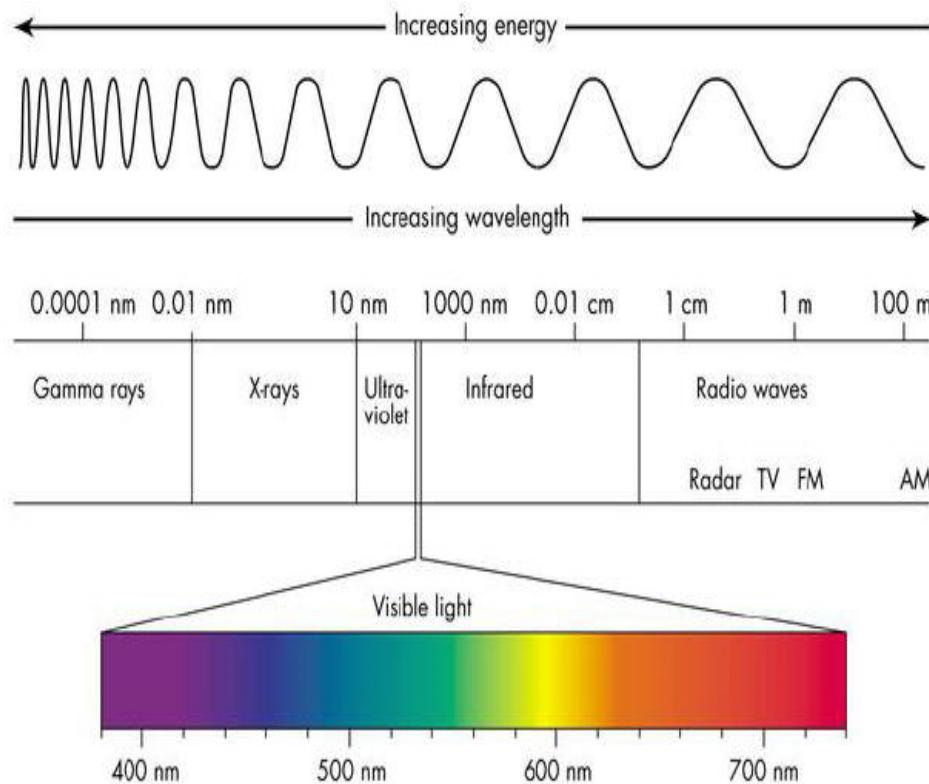
La **natura** dels àtoms i els seus electrons (**material**) controla el tipus d'emissió/scatter o reflexió/absorció de llum



2.3.2. Materials i Llums

La **llum** també es pot considerar com a un **conjunt d'ones**:

- La **longitud d'ona** és la distància entre un màxim i un altre de l'ona
- La longitud d'ona visible va de 400nm a 700nm aproximadament



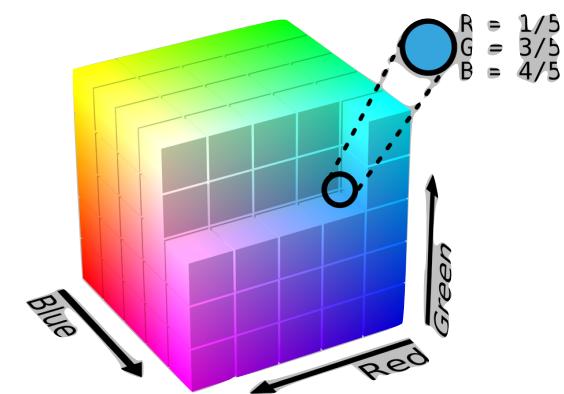
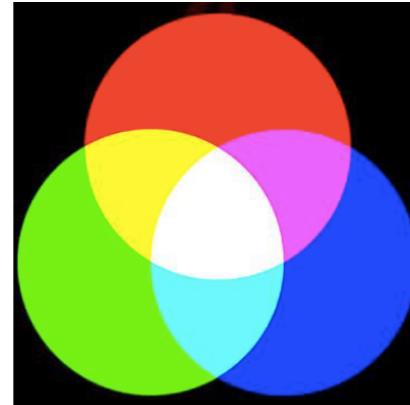
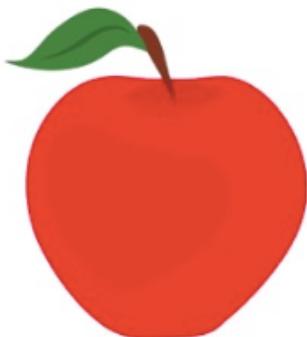
La llum visible té una longitud d'ona a 400nm – 700nm, una porció petita de l'espectre de la llum

Només ens interessa en l'espectre visible i definir/ agrupar rangs amplis de longitud d'ona, aproximadament:

Vermell (R), Verd (G) I Blau (B)

2.3.2.1. Interacció de la llum i els materials

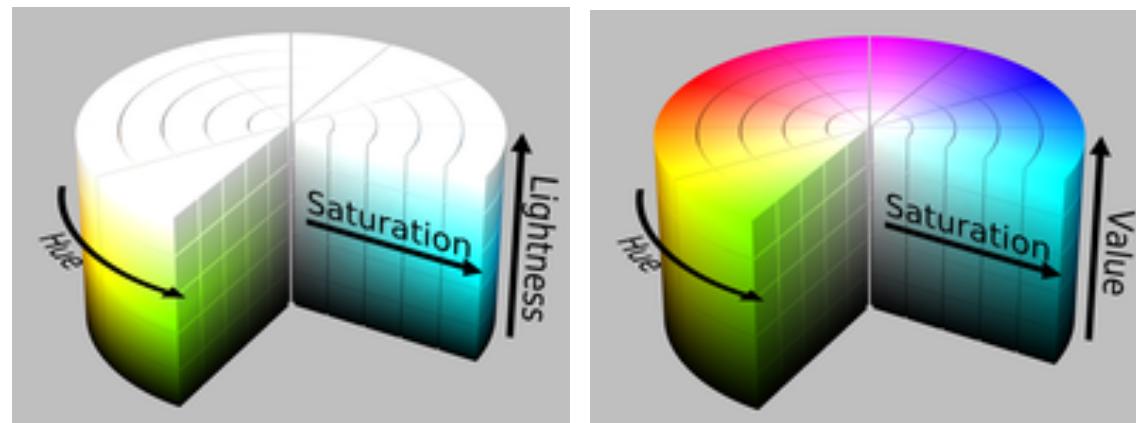
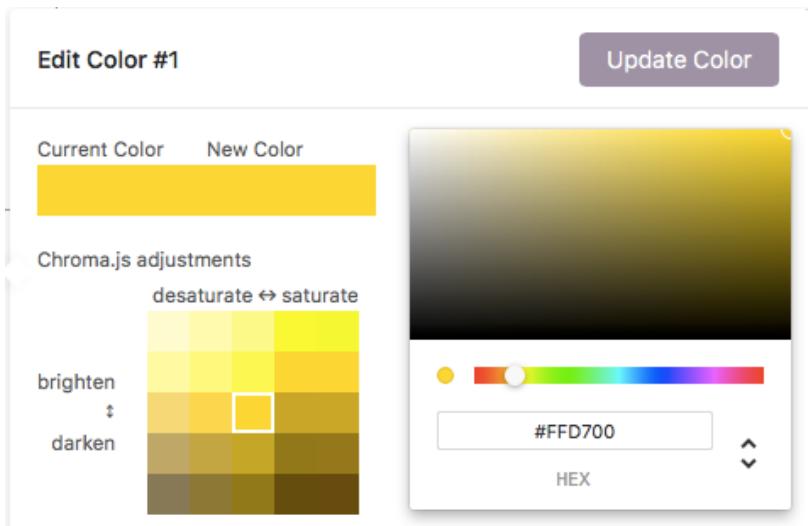
- Codificació RGB del color
 - Additiu
 - Tres canals
 - Cada canal està entre
 - 0 i 1 (float)
 - 0..255 (integer)
- Què fa veure que una poma és vermella?



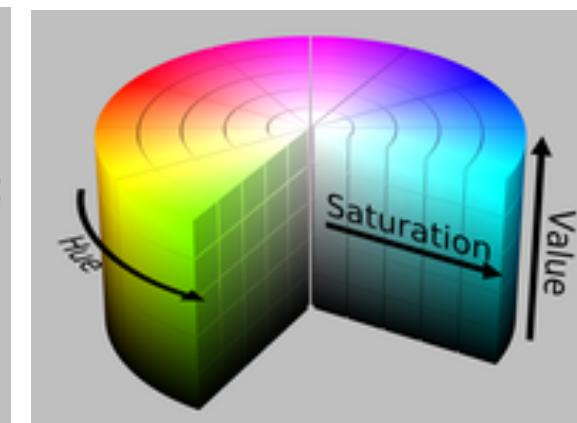
Si li arriba llum blanca, el **material** absorbeix la llum dels canals G i B i reflecteix només llum pel canal R.

2.3.2.1. Interacció de la llum i els materials

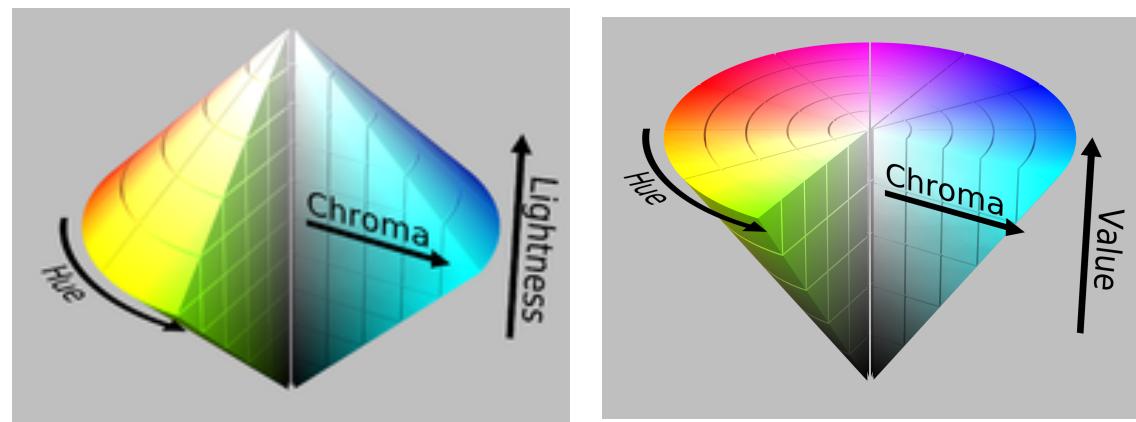
- Codificació HSL i HSV
 - Hue
 - Saturation
 - Ligthness (Value)
- Més intuïtius
- Usats en paletes



HSL



HSV



<https://projects.susielu.com/viz-palette>
<http://colorbrewer2.org/>

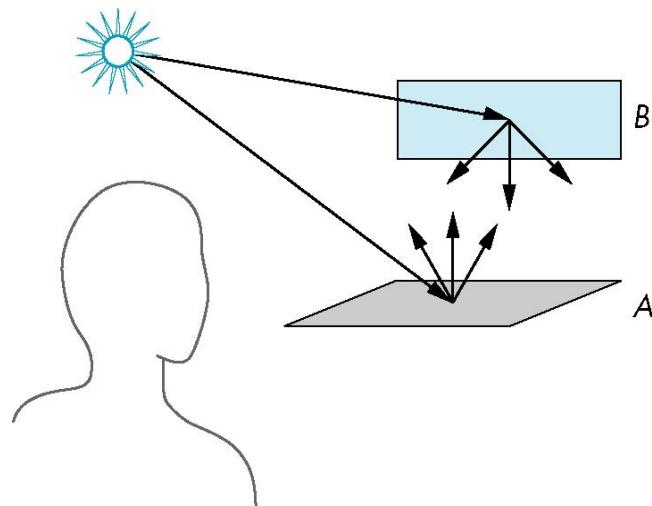
2.3.2.1. Interacció de la llum i els materials

Quina llum arriba a l'ull?

1. Quan la llum impacta en la superfície A:

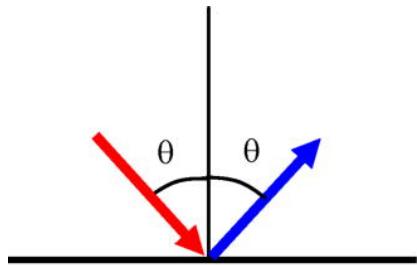
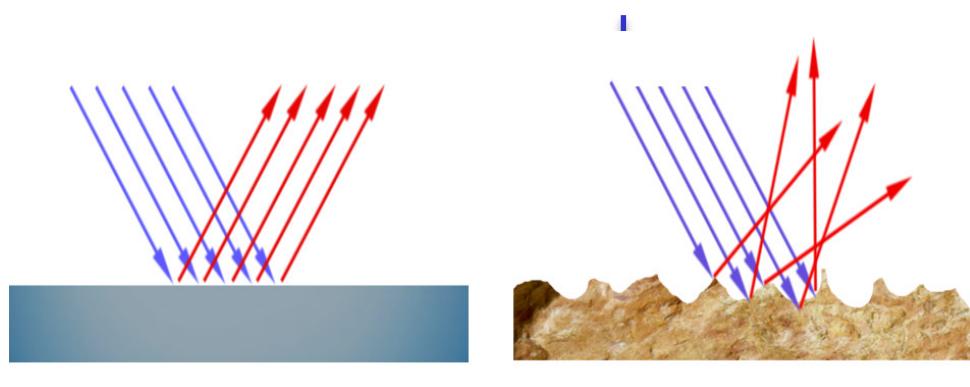
- Una part rebota (**scattering**)
- Una part s'absorbeix (**absorption**)
- Una part es pot **transmetre** (si l'objecte és transparent)

2. La superfície A pot també emetre llum (**emission**)

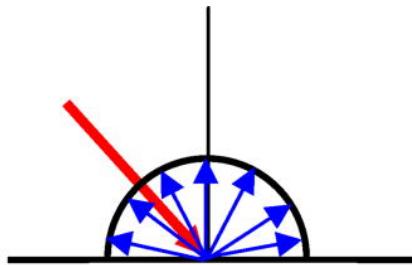


2.3.2.1. Interacció de la llum i els materials

Tipus de reflexions de la llum (**scattering**):



Reflexió especular



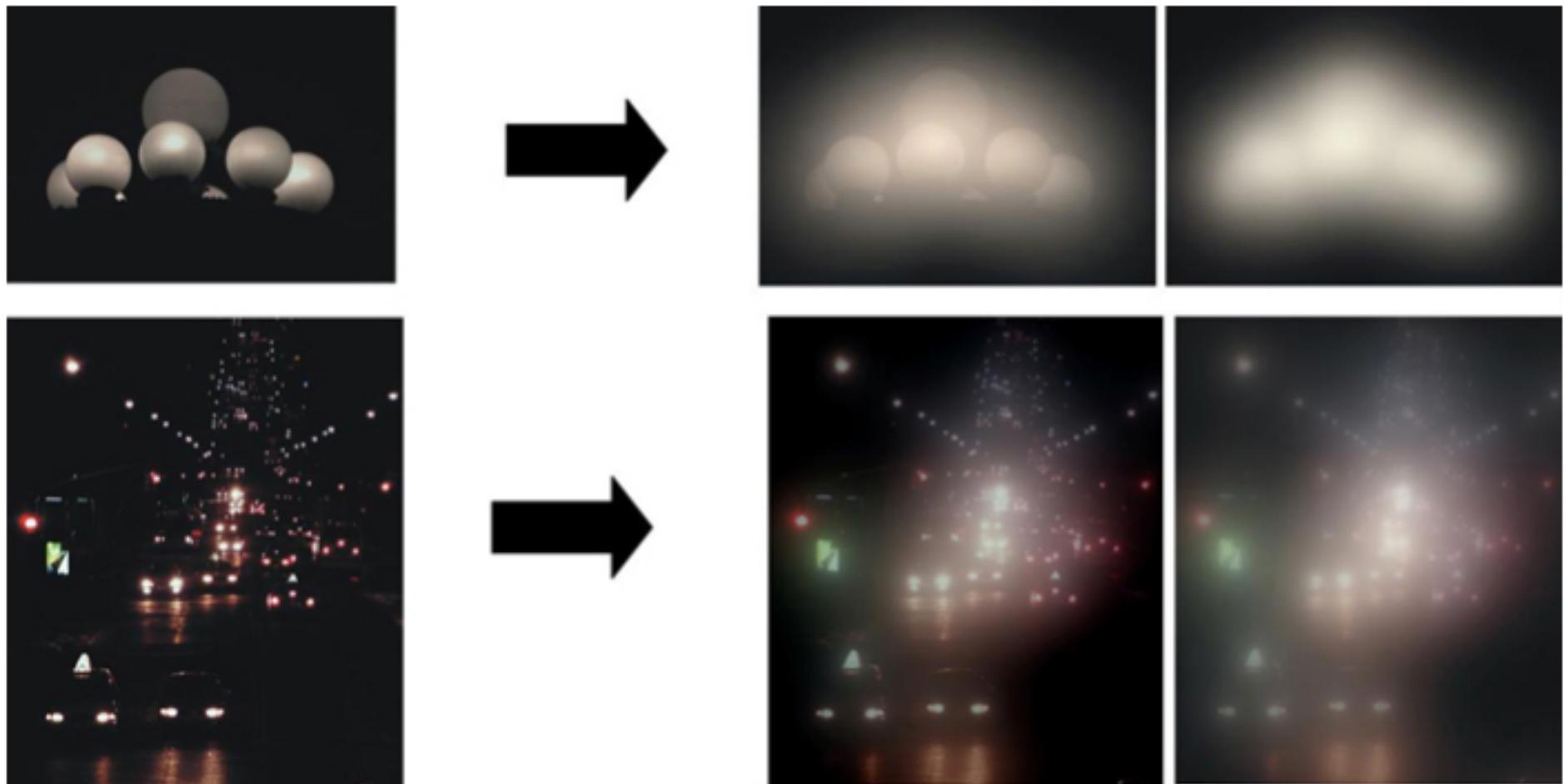
Reflexió difusa



<https://www.khanacademy.org/science/physics/geometric-optics/reflection-refraction/v/specular-and-diffuse-reflection>

2.3.2.1. Interacció de la llum i els materials

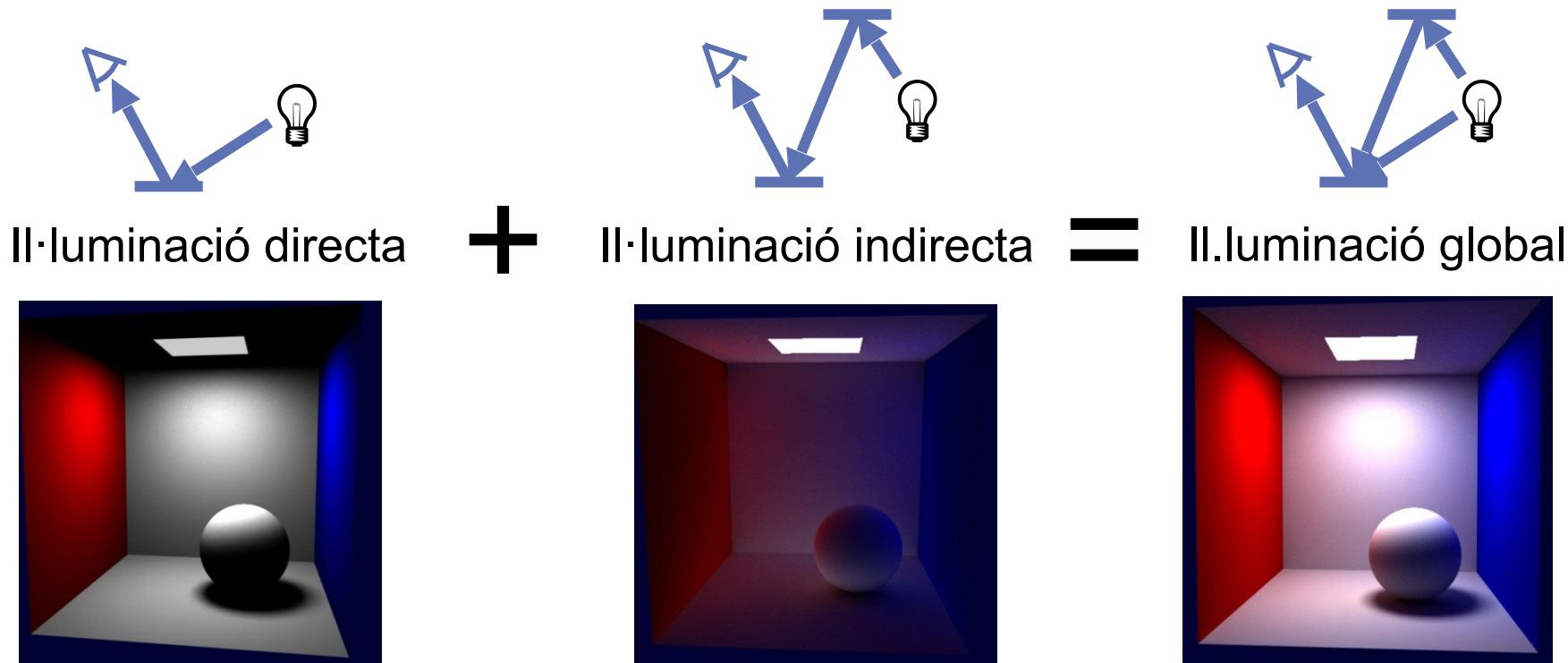
- Simple/Multiple scattering



http://www.cs.columbia.edu/CAVE/projects/ptping_media/

2.3.2.1. Interacció de la llum i els materials

- La il·luminació es pot calcular amb dos termes:



2.3.2.1. Interacció de la llum i els materials



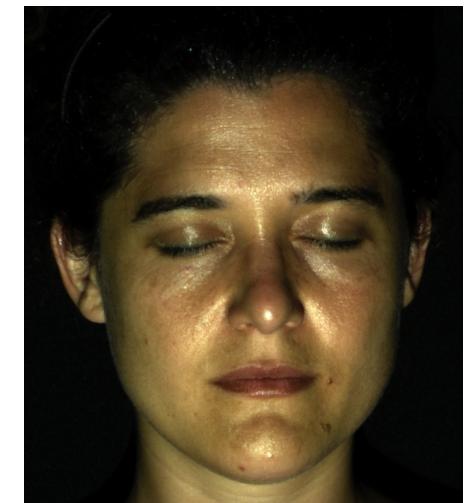
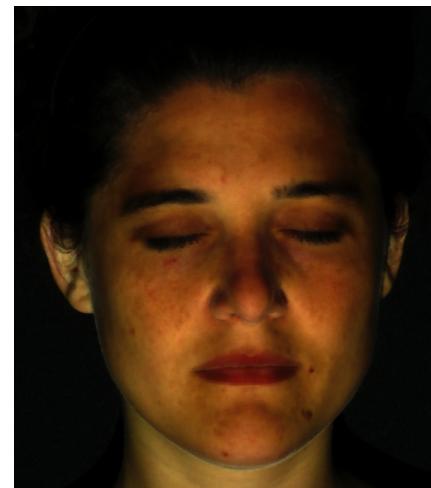
Il·luminació directa



Il·luminació indirecta



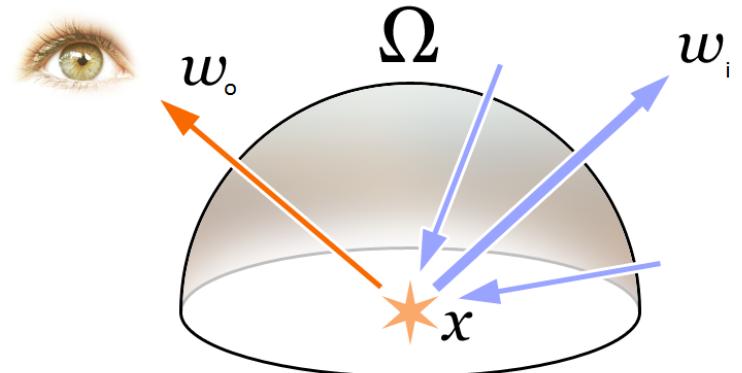
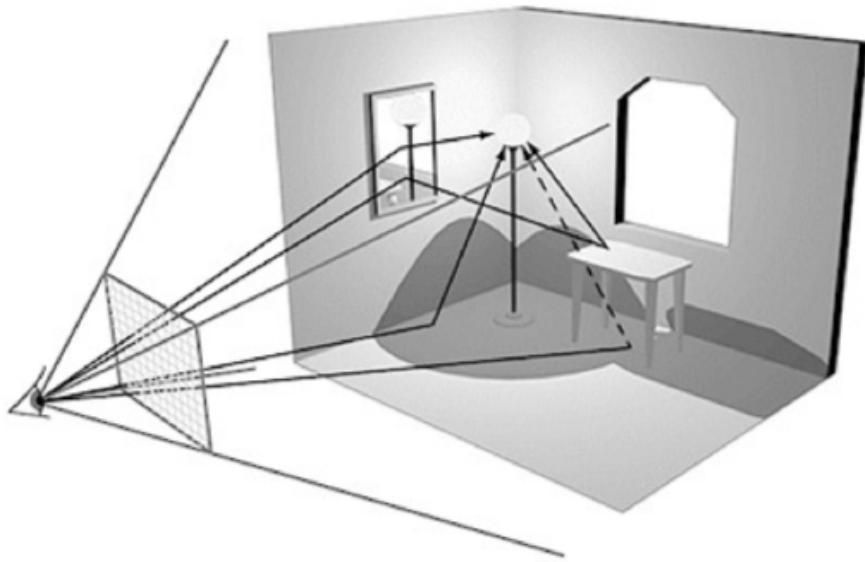
Il·luminació global



S. Nayar, G. Krishnan, M. Grossberg, and R. Raskar. "Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination," SIGGRAPH '06

<http://www.cs.columbia.edu/CAVE/projects/separation/videos/Separation.mov>

2.3.2. Materials i Llums



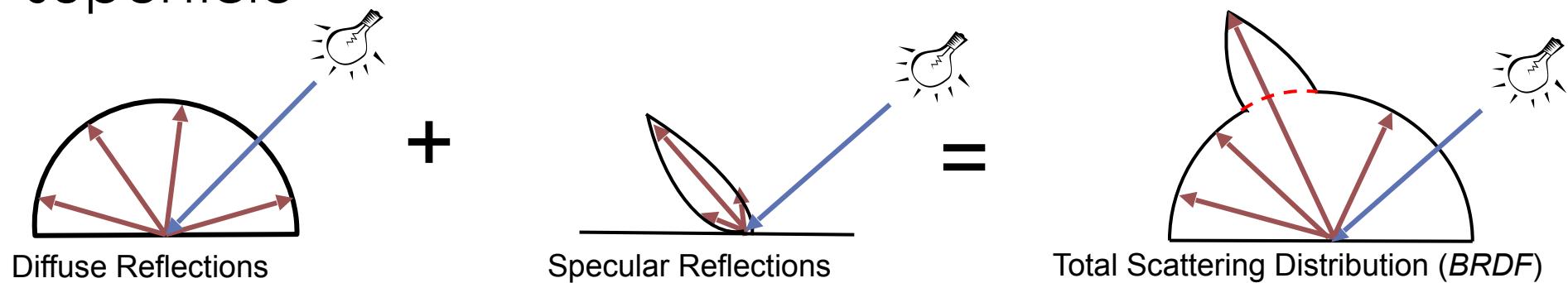
http://en.wikipedia.org/wiki/Rendering_equation

Equació del rendering: equació que serveix per a calcular la il·luminació a tot punt de l'escena

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

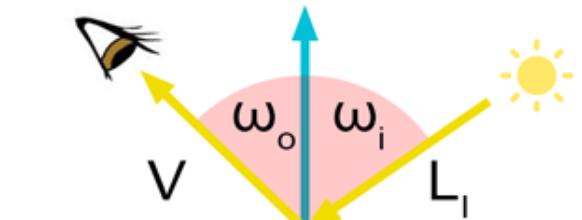
2.3.2.1. Interacció de la llum i els materials

Comportament de la llum quan impacta en una superfície



$$BRDF(\omega_o, \omega_i) \rightarrow$$

Quantitat de llum reflectida
en la direcció de visió per
una determinada direcció
de llum incident



- BRDF positiva, recíproca conservativa
- Exemples: **Lambert**, **Phong**, Cook-Torrance, Oren-Nayar, etc.

2.3.2.1. Interacció de la llum i els materials

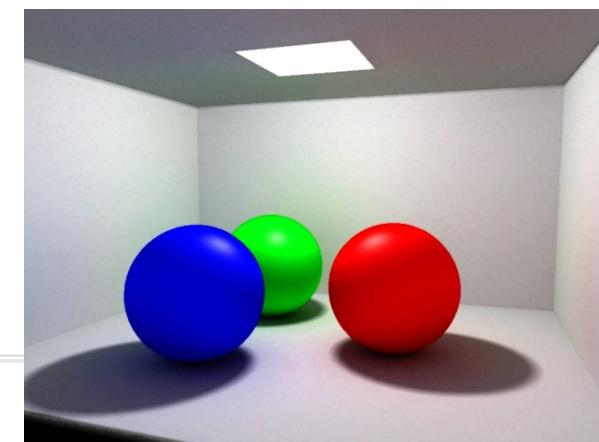
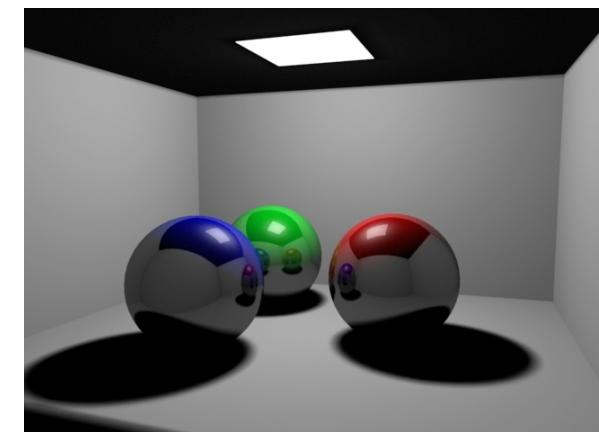
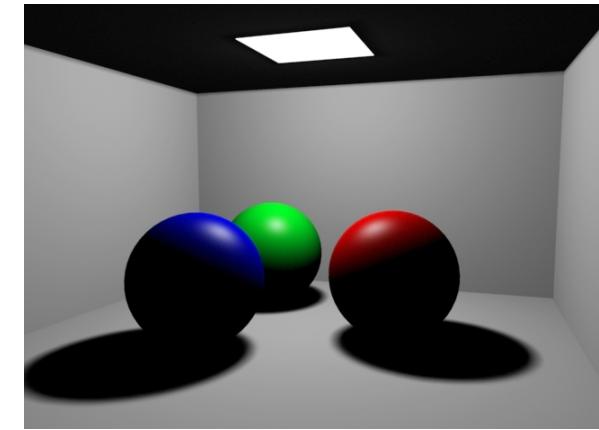
Tipus de càlculs:

Rendering local: té en compte la llum directa de les fonts de llum i modela alguns efectes indirectes, com les ombres:

- Ràpid i fàcil d'adaptar al pipeline gràfic
- Menys realisme

Rendering global: té en compte la il·luminació amb totes les superfícies i les llums de l'escena.

- Permetombres, reflexions entre objectes, refraccions, càustiques, efectes de mitjans participatius (boira, aigua, etc.)

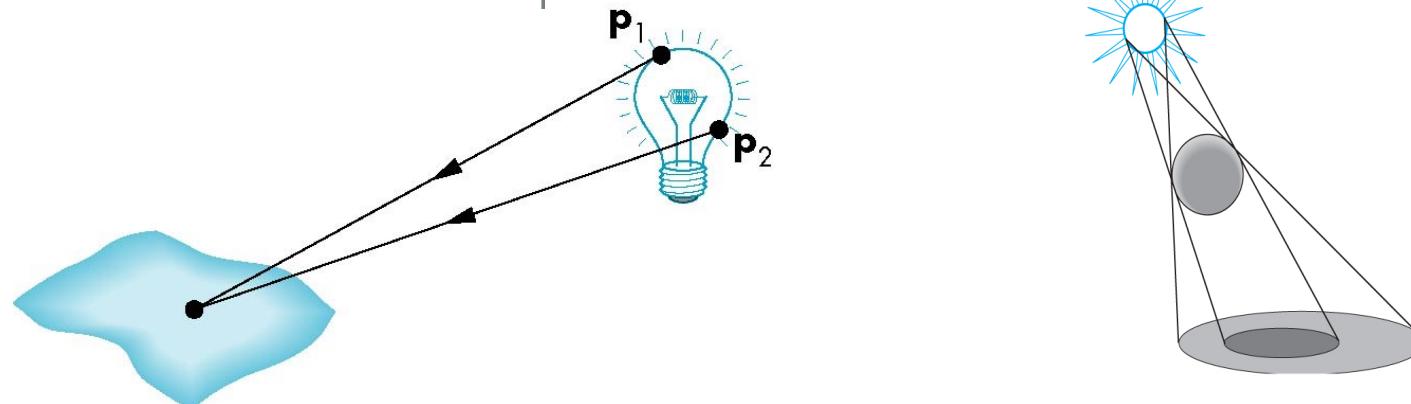


http://www.spot3d.com/vray/help/200R1/examples_GI.htm

2.3.2. Llums

Un **focus de llum** és un objecte que emet llum.

Les llums reals són difícils de modelar per què cal integrar l'emissió de la llum de tota la seva superfície



Tota llum emet en les tres components de color $\mathbf{I} = [R \ G \ B]$

Una llum emet tres tipus de comportaments:

- comportament ambient I_a ,
- comportament difús I_d
- comportament especular I_s

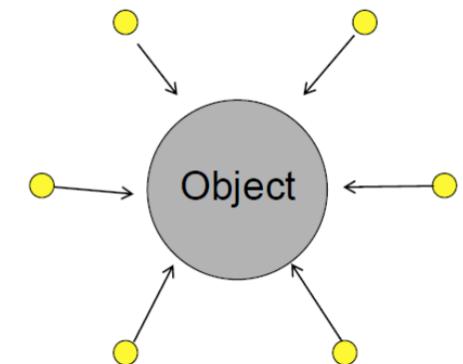
2.3.2.2. Llums

Tipus de llums:

Tipus	Descripció	Exemples / Imatges
Ambient global	Quantitat de llum constant en tots els punts de l'escena. Modela part de la component difusa de la llum.	Color de les parts no il.luminades 
Puntual	No tenen dimensió, ni direcció (emeten en forma radial). Es caracteritzen per la posició i la seva emissió (o intensitat)	Bombeta 
Direccional	Focus a l'infinít, es caracteritzen per la seva direcció i la seva intensitat	Sol, Lluna 
Spot light	Llum de tipus focus, localitzada en una posició, emet en una certa direcció i només en un conus de llum	Focus 

2.3.2.2. Llums

Llum ambient global:



És el resultat de múltiples interaccions entre llums i els objectes de l'entorn

El seu càlcul exacte és molt costós. S'aproxima.

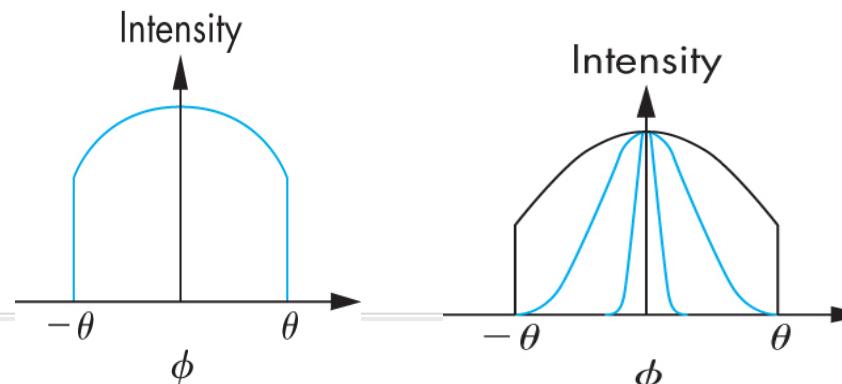
Es caracteritza amb les tres components de color R, G, B

$$I_a = [I_{aR} \ I_{aG} \ I_{aB}]$$

2.3.2. Llums

Atenuació en profunditat:

- La quantitat de llum rebuda per una superfície és inversament proporcional a la distància de la superfície al focus de llum $1/d^2$
- Es pot aplicar un factor de $1/(a + bd + cd^2)$ a les components de la llum, tan difusa com especular, que arriben a la superfície, on a, b, c són paràmetres que es determinen segons l'escena.
- en spotlights l'atenuació pot ser concentrada al centre del con, exponencial



2.3.2.3. Materials

Els materials dels objectes són les propietats òptiques dels objectes que defineixen el seu comportament amb la llum.

Defineixen l'aspecte de l'objecte



Plastic



Metal



Matte

2.3.2.3. Materials

- Propietats que defineixen un material

- Component **difusa** K_d (R, G, B)
- Component **ambient** K_a (R, G, B)
- Opacitat **alfa** = 0 (transparent) 1 (opac)
- Component **especular** $K_s(R, G, B)$
- Exponent de **reflexió especular** (shininess) $a_s = 1...500$



2.3.2.4. Model de Phong

Model d'il·luminació **local** que es pot calcular de forma molt eficient integrat dins del pipeline gràfic o al raycasting.

Utilitza quatre vectors:

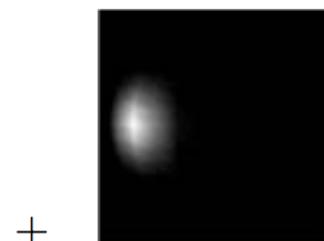
- del punt de la superfície a la llum \vec{L}
- del punt de la superfície a l'observador \vec{V}
- la normal al punt \vec{N}
- el vector reflectit \vec{R}

Modela tres components:

- ambient
- difusa
- especular



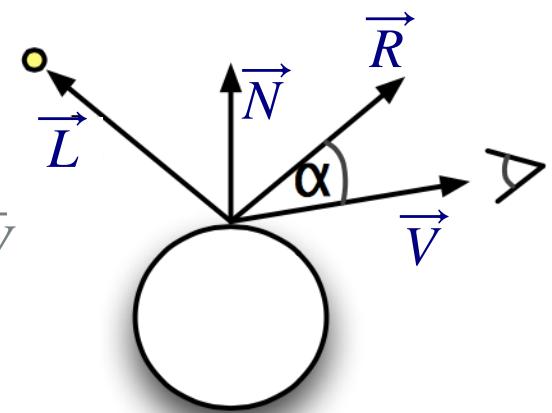
Diffuse



Specular



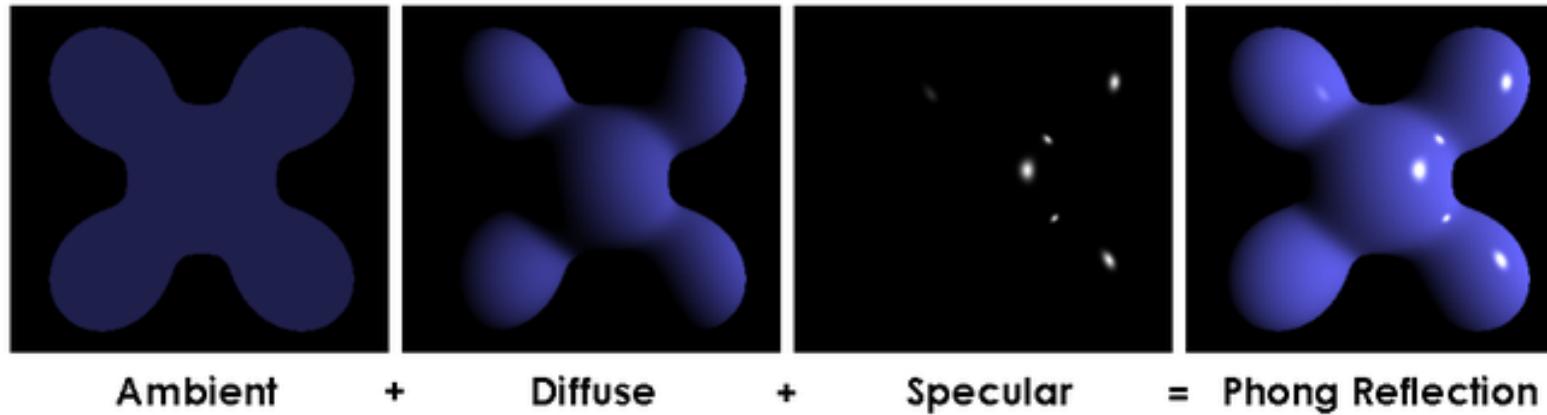
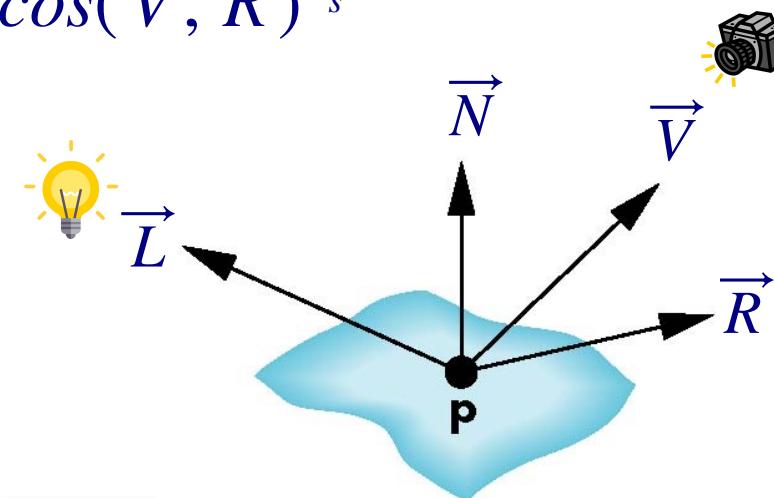
Ambient



2.3.2.4. Model de Phong

Per a una llum, el model de Phong es pot escriure com:

$$I = k_a I_a + k_d I_d \cos(\vec{L}, \vec{N}) + k_s I_s \cos(\vec{V}, \vec{R})^{\alpha_s}$$



<http://multivis.net/lecture/phong.html>

2.3.2.4. Model de Phong

Notació:

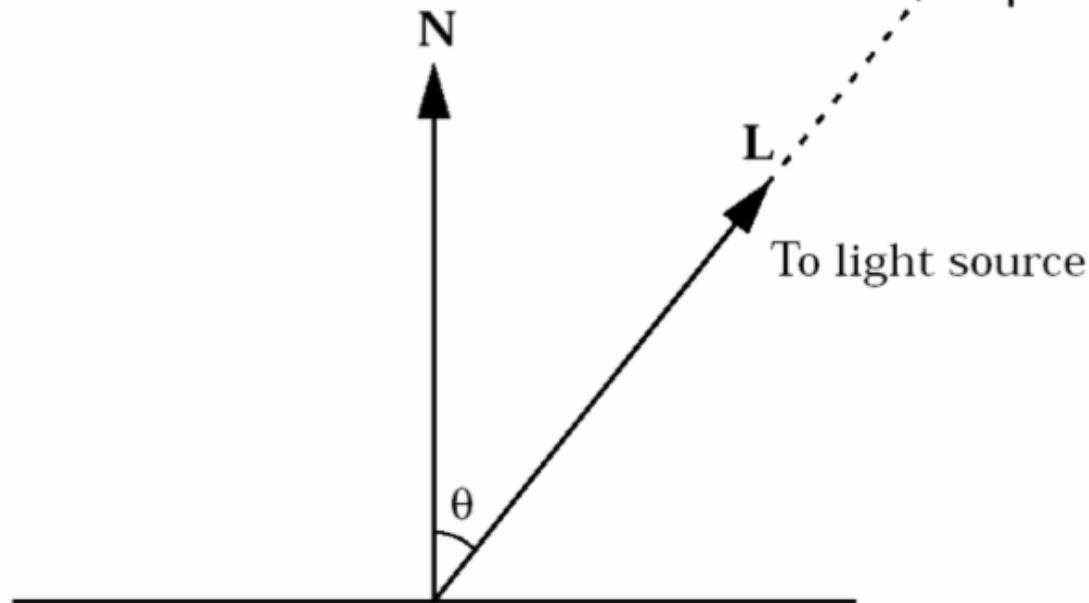
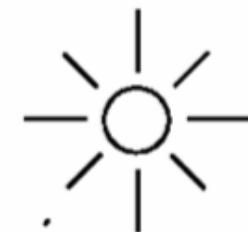
\vec{N} vector Normal a la superfície

\vec{L} vector des de la superfície a la llum

\vec{N} és unitari

θ és l'angle d'incidència

Normal vector



2.3.2.4. Model de Phong: llum ambient

La **llum ambient** modela la part de la llum difusa que està implicada en les reflexions indirectes.

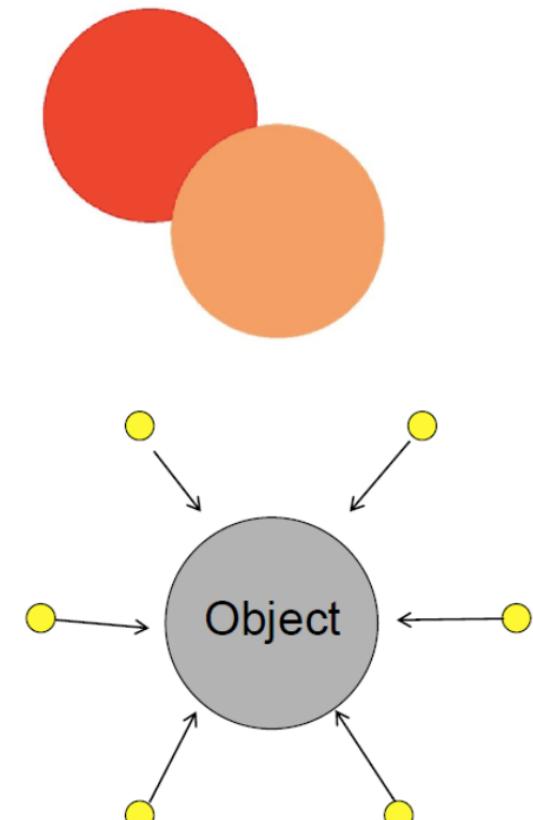
S'utilitza una aproximació molt senzilla:

$$I = I_a k_a$$

On,

I_a és la Intensitat ambient de la llum

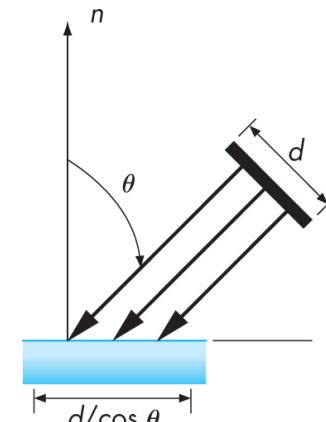
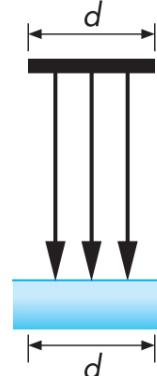
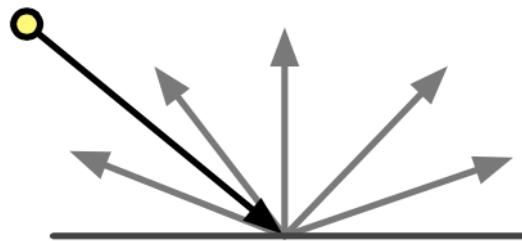
k_a és el component ambient del material



- Tot punt de l'objecte tindrà la mateixa intensitat
- No es té sensació 3D

2.3.2.4. Model de Phong: part difusa

Llei de Lambert: Modela la llum **difusa** pura, on la llum es reflexa igualment a totes les direccions



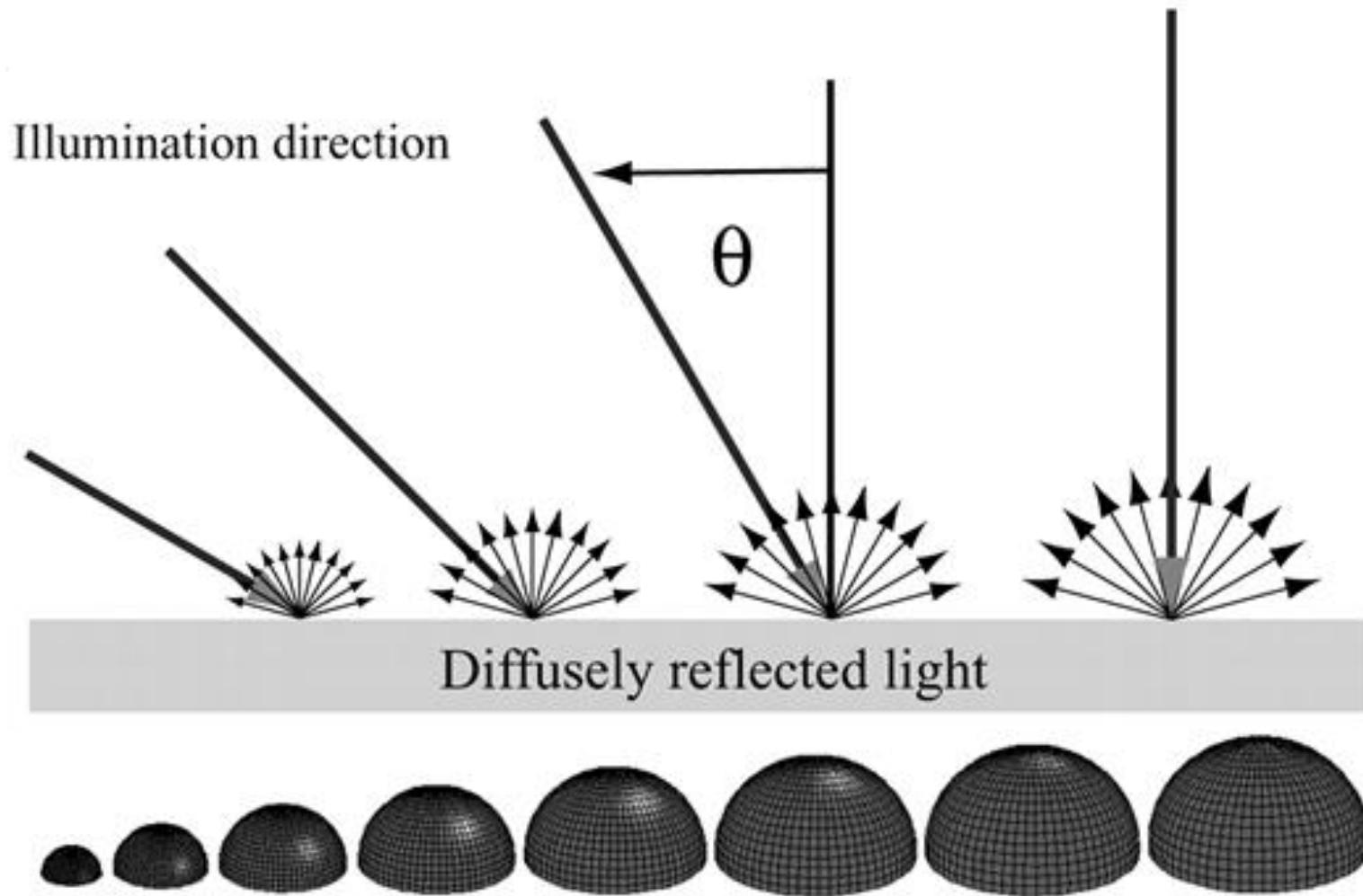
$$I = I_d \ k_d \cos \theta_i$$

El coeficient **k_d** del material indica quanta llum de cada component (R, G, B) es reflexa en cada canal

La **I_d** és la intensitat difusa de la llum

$$\cos \theta_i = \vec{L} \cdot \vec{N} \text{ (amb els vectors normalitzats)}$$

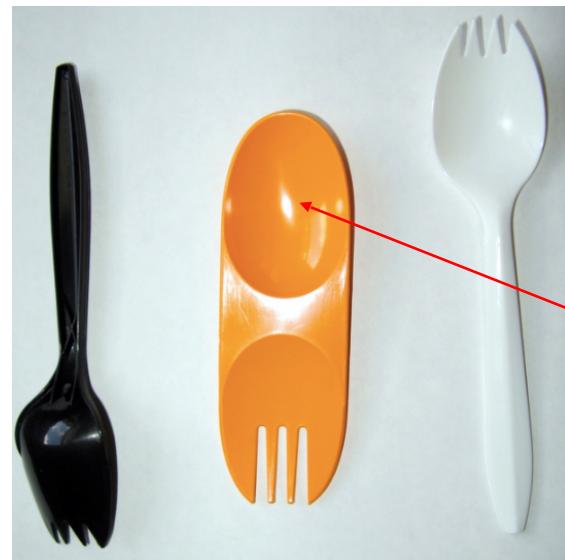
2.3.2.4. Model de Phong: part difusa



2.3.2.4. Model de Phong: part espectral

La majoria de superfícies no són idealment difuses ni idealment especulars

Les superfícies suaus mostren brillants (**highlights**) especulars produïdes per que la llum incident es reflecteix en direccions concentrades prop de la direcció de la reflexió pura



Reflexe espectral

2.3.2.4. Model de Phong: part espectral

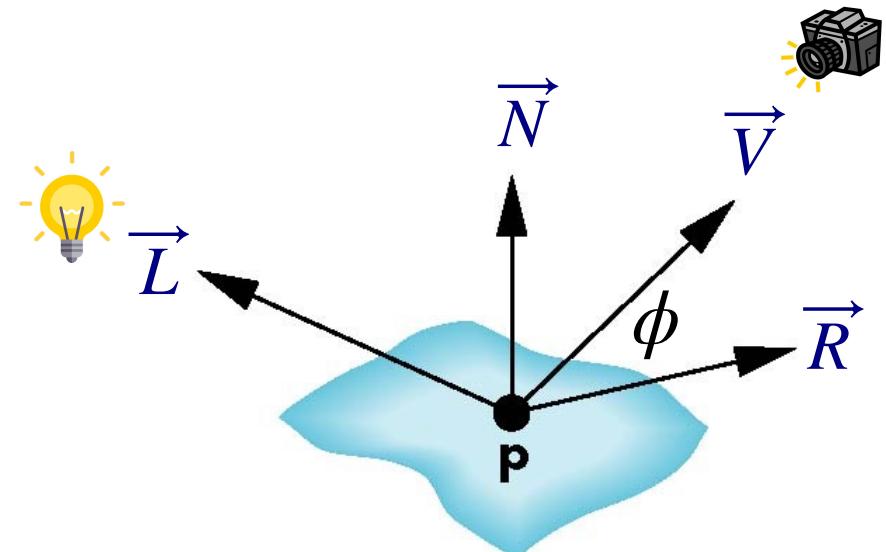
Terme que decrementa quan l'angle entre l'observador i la reflexió ideal augmenta.

$$I \sim k_s * I_s * \cos^\alpha \phi$$

Exponent de reflexió espectral



$$\cos \phi = \vec{R} \cdot \vec{V} \quad (\text{amb els vectors normalitzats})$$



2.3.2.4. Model de Phong: part espectral

Com es calcula el **raig reflectit R**?

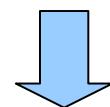
A partir del raig incident (L) i la normal a l'objecte en el punt p (N)

$$\vec{R} = \vec{L} + 2\vec{U}$$

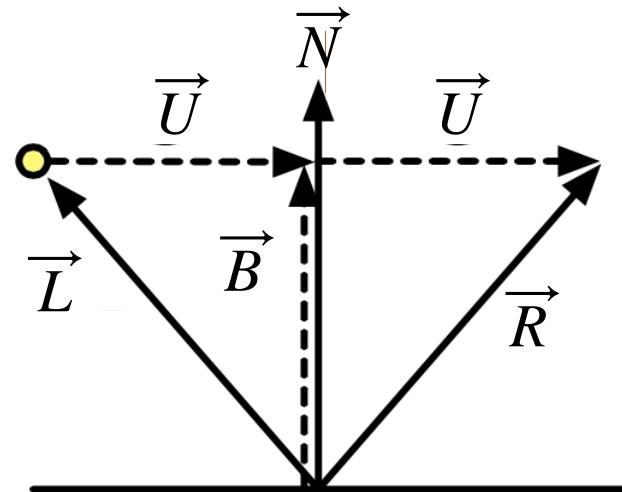
$$\vec{R} = \vec{L} + 2(-\vec{L} + \vec{B})$$

$$\vec{R} = \vec{L} + 2(-\vec{L} + \vec{N}(\vec{N} \cdot \vec{L}))$$

$$\vec{R} = 2\vec{N}(\vec{N} \cdot \vec{L}) - \vec{L}$$



$$\boxed{\vec{R} = 2\vec{N}(\vec{N} \cdot \vec{L}) - \vec{L}}$$

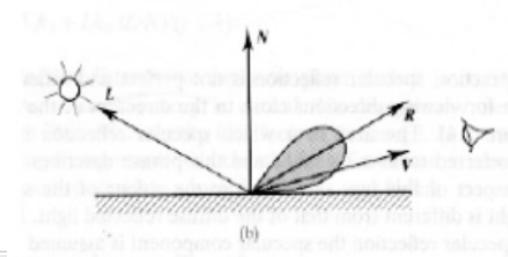
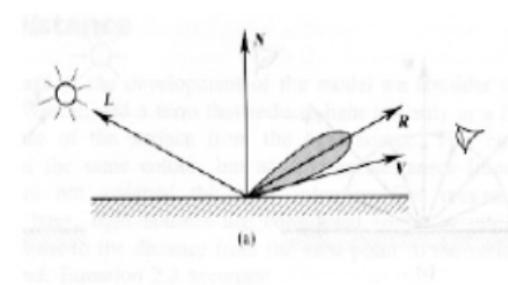
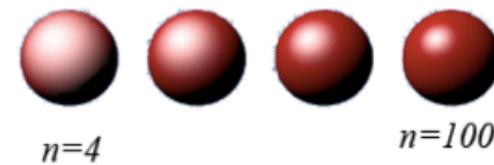
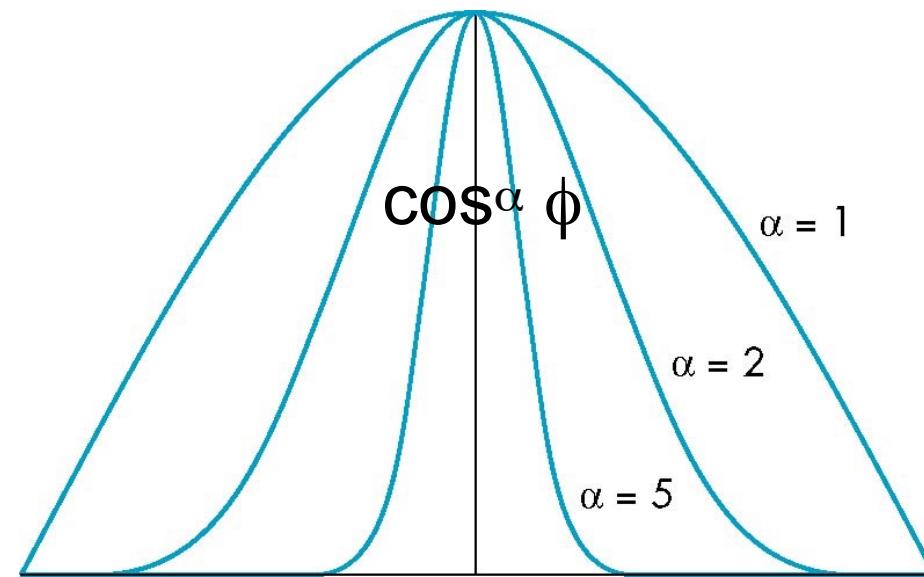


2.3.2.4. Model de Phong: part espectral

L'exponent de reflexió espectral:

Els valors de α_s entre 100 i 200 corresponen als metalls.

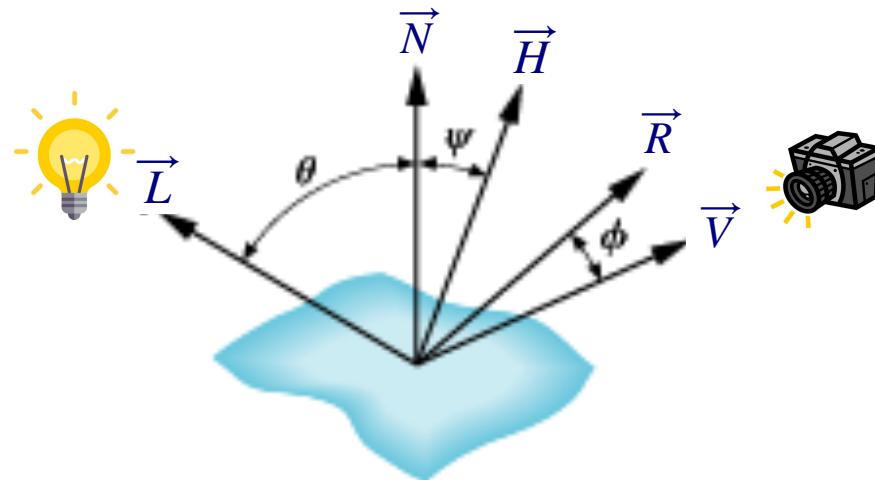
Valors entre 5 i 10 donen una aparença de plàstic.



2.3.2.4. Model de Phong: optimització

El terme espectral de Phong és problemàtic ja que s'ha de recalcular un nou vector de **reflexió** i el vector de visió a cada vèrtex.

Blinn va proposar una aproximació usant el vector mig, que és més **eficient**.



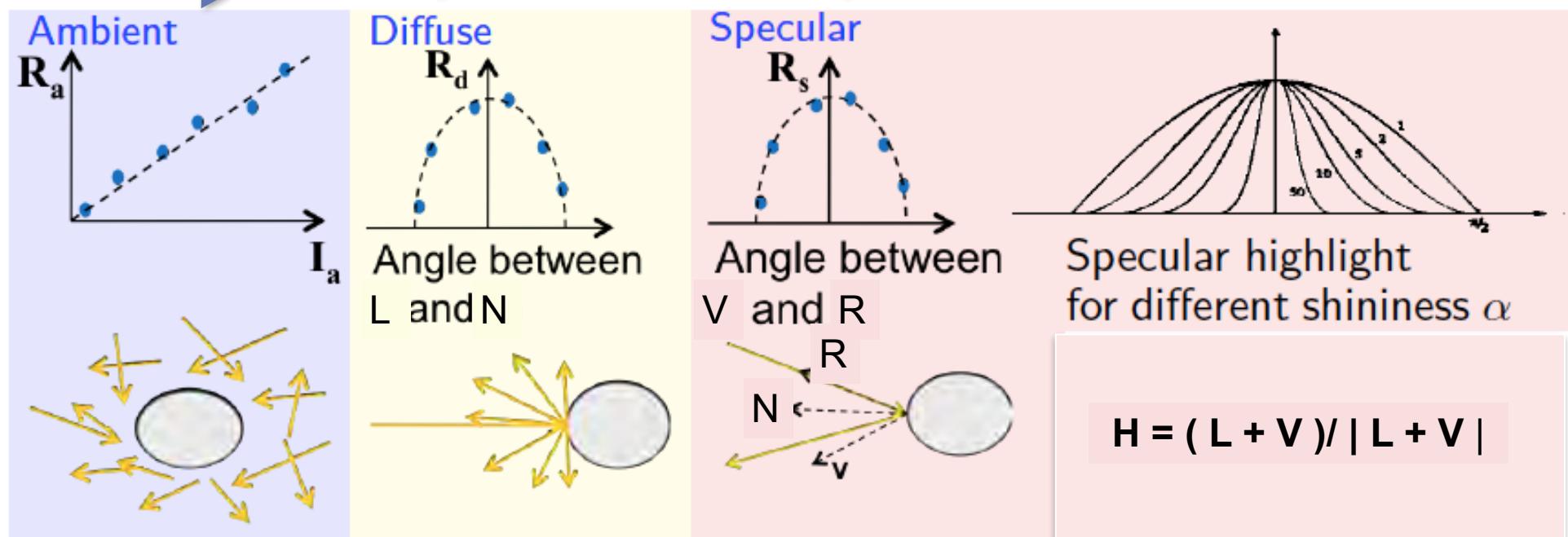
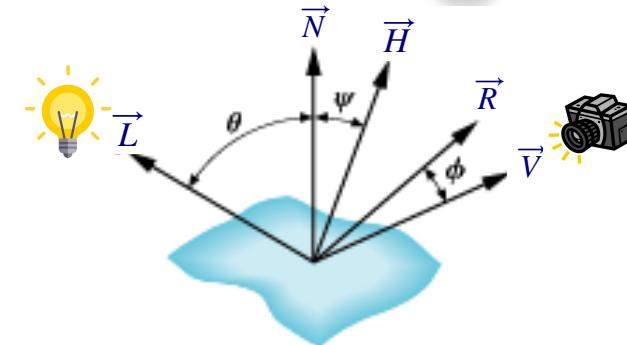
$$\vec{H} = (\vec{L} + \vec{V}) / |\vec{L} + \vec{V}|$$

\vec{H} és el vector mig normalitzat entre \vec{L} i \vec{V} , amb \vec{L} i \vec{V} normalitzats.

2.3.2.4. Model de Blinn-Phong

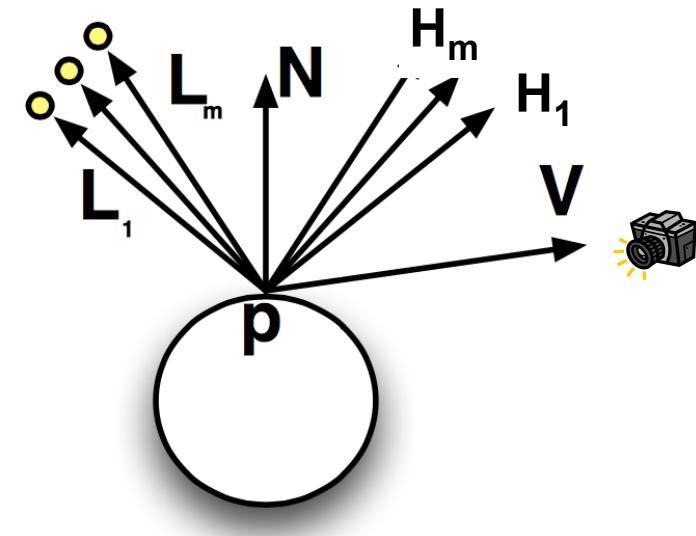
Model de **Blinn-Phong**:

$$I = k_a I_a + k_d I_d \cos(\vec{L}, \vec{N}) + k_s I_s \cos(\vec{N}, \vec{H})^\beta$$



2.3.2.4. Model de Blinn-Phong

La intensitat total en un punt es calcula com la contribució de **totes** les llums de l'escena en el punt més la intensitat ambient global de l'escena.



$$I_{total} = I_{a_{global}} K_a + \sum_{i=1}^{numLlums} \frac{1.0}{a_i + b_i d_i + c_i d_i^2} (I_{d_i} K_d \max(L_i \cdot N, 0.0) + I_{s_i} K_s \max((N \cdot H_i), 0.0)^\beta) + I_{a_i} K_a$$

On,

L és el vector normalitzat entre el punt i la llum,

N és la normal al punt,

V és el vector normalitzat entre el punt i l'observador

H es calculen a partir dels vectors anteriors

d_i és la distància del punt a la llum

Índex

2.1. Introducció: algorisme principal

2.2. Càlcul de Raig Primari

2.3. Càlcul del color:

2.3.1. Interseccions amb objectes

2.3.2. Materials i Llums

2.4. Ombres

2.3.2.1. Interacció de les llums i els materials

2.3.2.2. Llums

2.3.2.3. Materials

2.3.2.4. El model de Phong

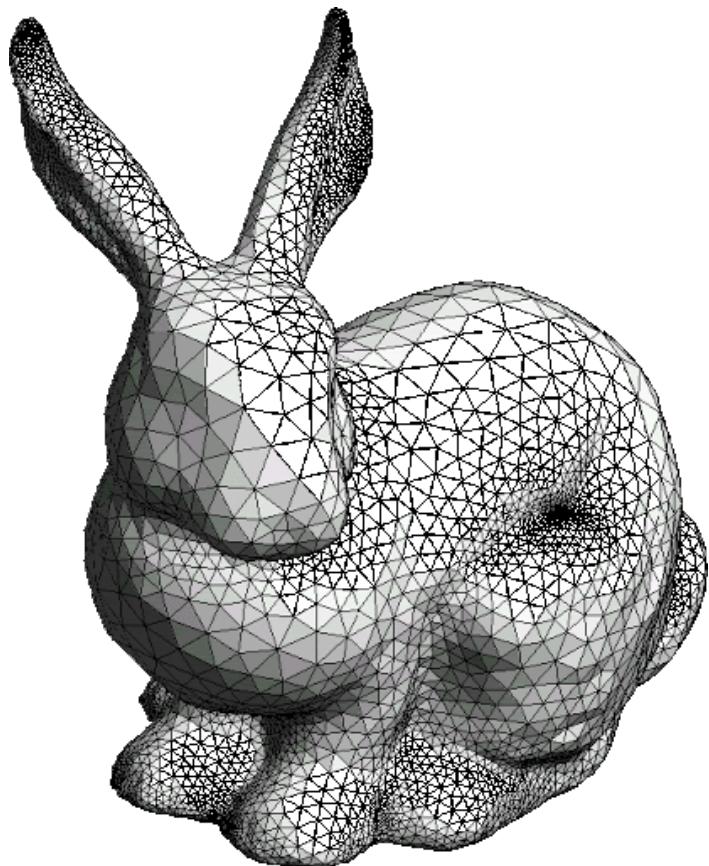
2.3.2.5. Textures

2.5. Reflejos i traços arènics

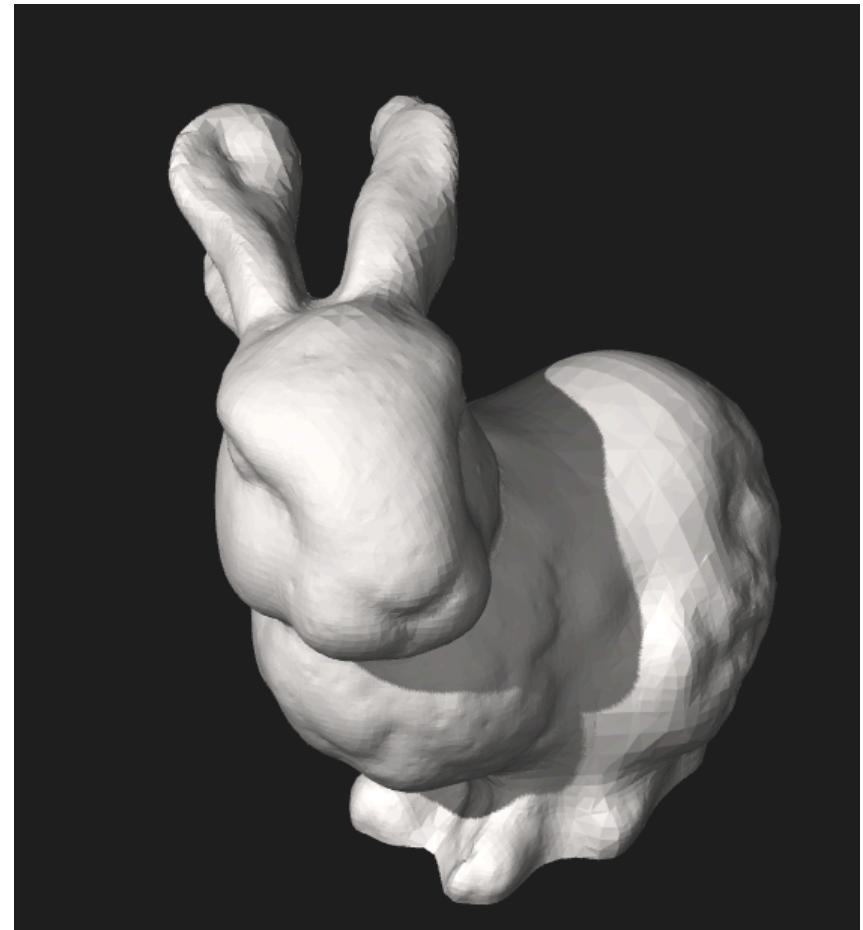
NOTA: Aquestes transparències es corresponen a les explicacions del tema 7.5 del llibre de referència bàsica (pag. 368 - 374)

[Angel2011] Edward Angel, Dave Shreiner, **Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, 6/E**, ISBN-10: 0132545233. ISBN-13: 9780132545235, Addison-Wesley, 2011

2.3.2.5. Textures



Stanford Bunny Mesh, Source:



https://en.wikipedia.org/wiki/Stanford_bunny

2.3.2.5. Textures



Using Uv arrays from Mesh Data
to map a texture on the Stanford
Bunny, Source:



El realisme requereix la introducció de detalls:

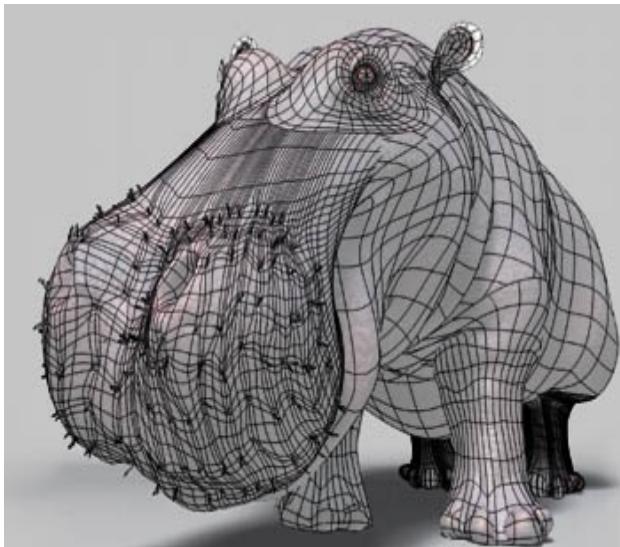
- modelant
- simulant diferents propietats d'il.luminació a cada punt

Els objectes reals no són uniformes en termes de color i normals a tots els punts: existeixen detalls amb altres freqüències que no són directament modelables.

<https://blenderartists.org/t/uv-unwrapped-stanford-bunny-happy-spring-equinox/1101297>

2.3.2.5. Textures

Modelatge (filferros)



Il.luminació + shading



Texture Mapping



<http://www.3drender.com/jbirn/productions.html>

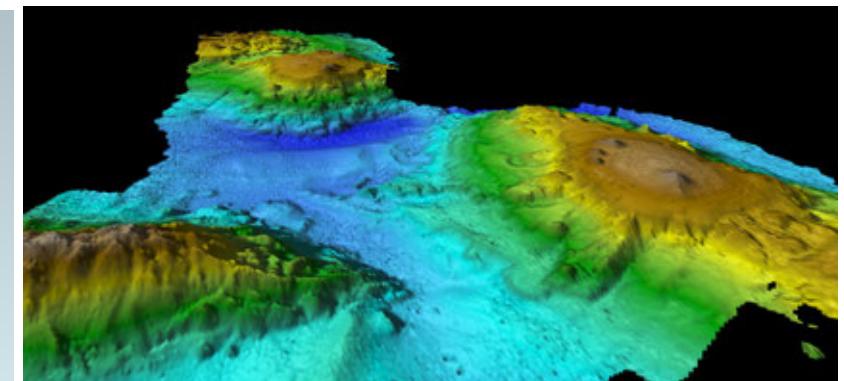


De menys a més realisme

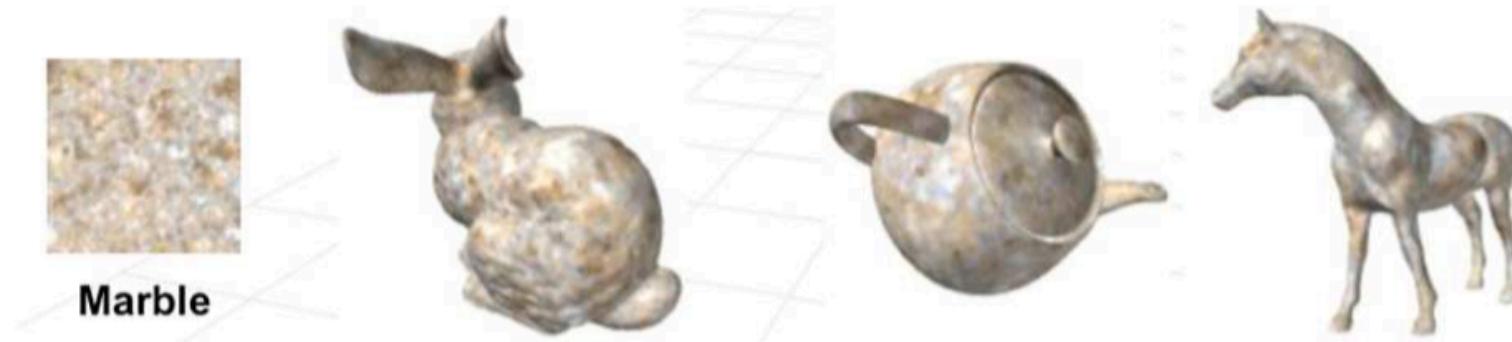
2.3.2.5. Textures

Per a què s'usen?

- Aplicacions: videojocs, mons virtuals, mapes,
 - Pocs polígons
 - Detalls creats essencialment amb mapes de textures:
 - Detalls de superfícies
 - Fum, danys, ...
 - Il·luminació,ombres
 - Càlculs locals d'il·luminació
 - Re-utilitzables
 - GPU



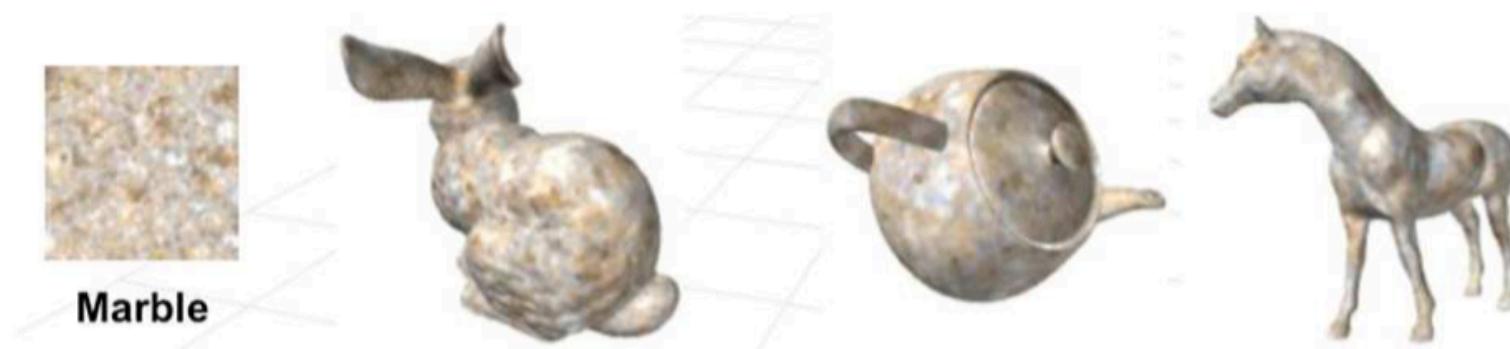
2.3.2.5. Textures



Objectiu:

Obtenir objectes amb aparences més **realistes i variables** usant diferents propietats òptiques en diferents punts de l'objecte

2.3.2.5. Textures

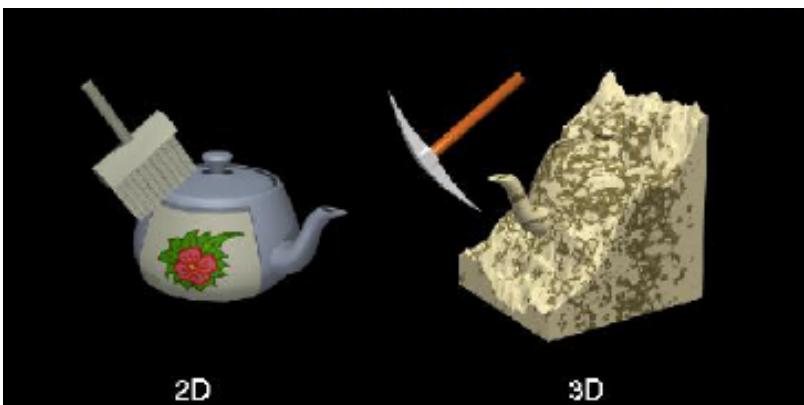
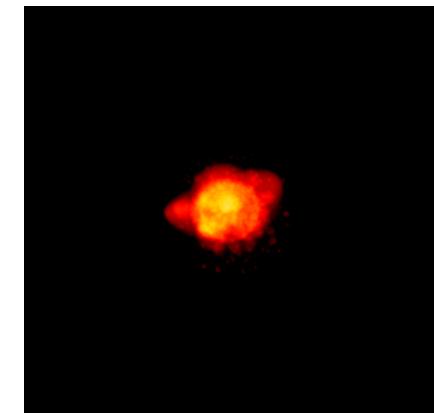


Qüestions bàsiques:

1. Què són i d'on venen les textures?
2. Com es mapejen les textures en els objectes?
3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?
4. Com s'integren en l'algorisme de RayTracing?

1. Què són i d'on venen?

- **Textures:** característiques superficials i volumètriques de detall d'un objecte.
- Poden ser 1D, 2D, 3D i 4D



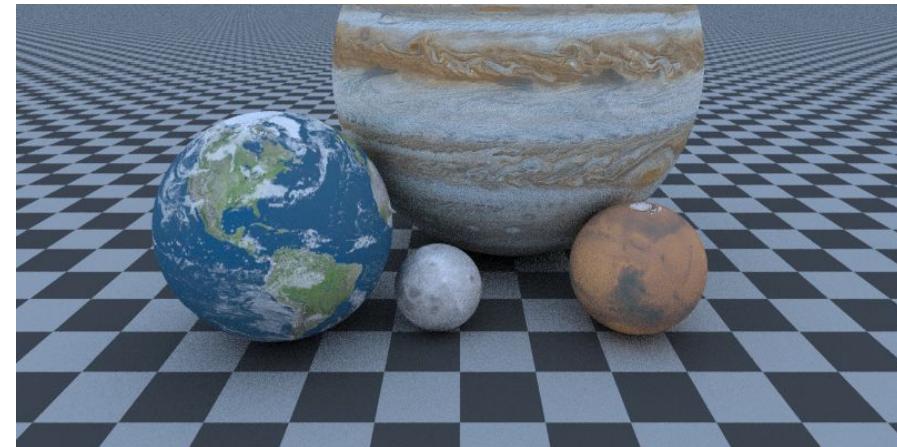
1. Què són i d'on venen?

- **Anàlisi:**

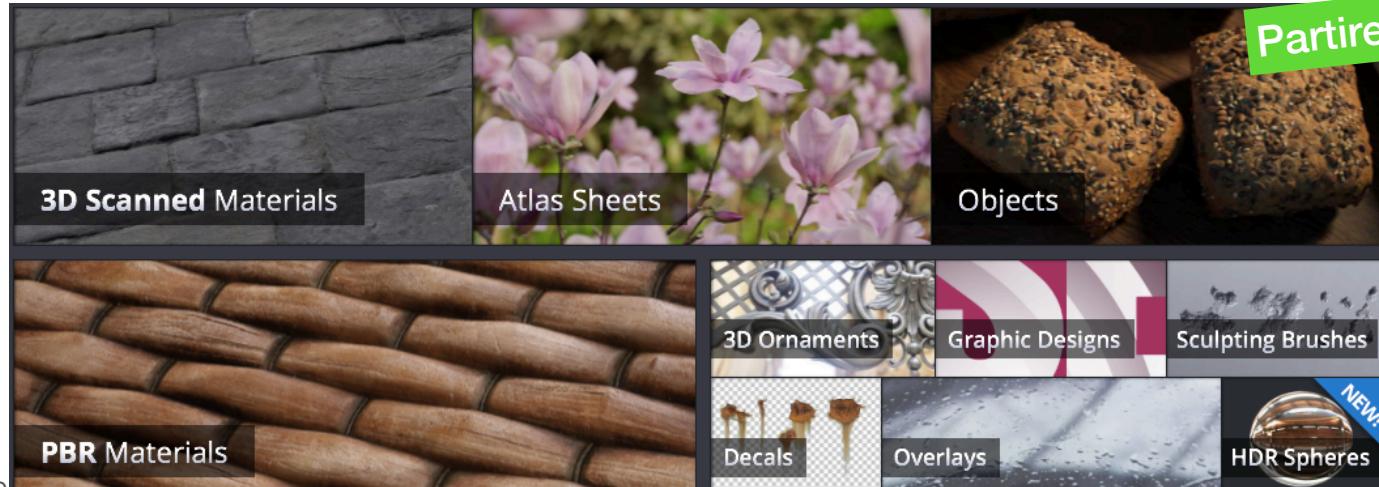
- Observació de les propietats físiques dels objectes texturats
- Deducció de les regles que determinen l'estruccuració dels detalls

- **Síntesi:**

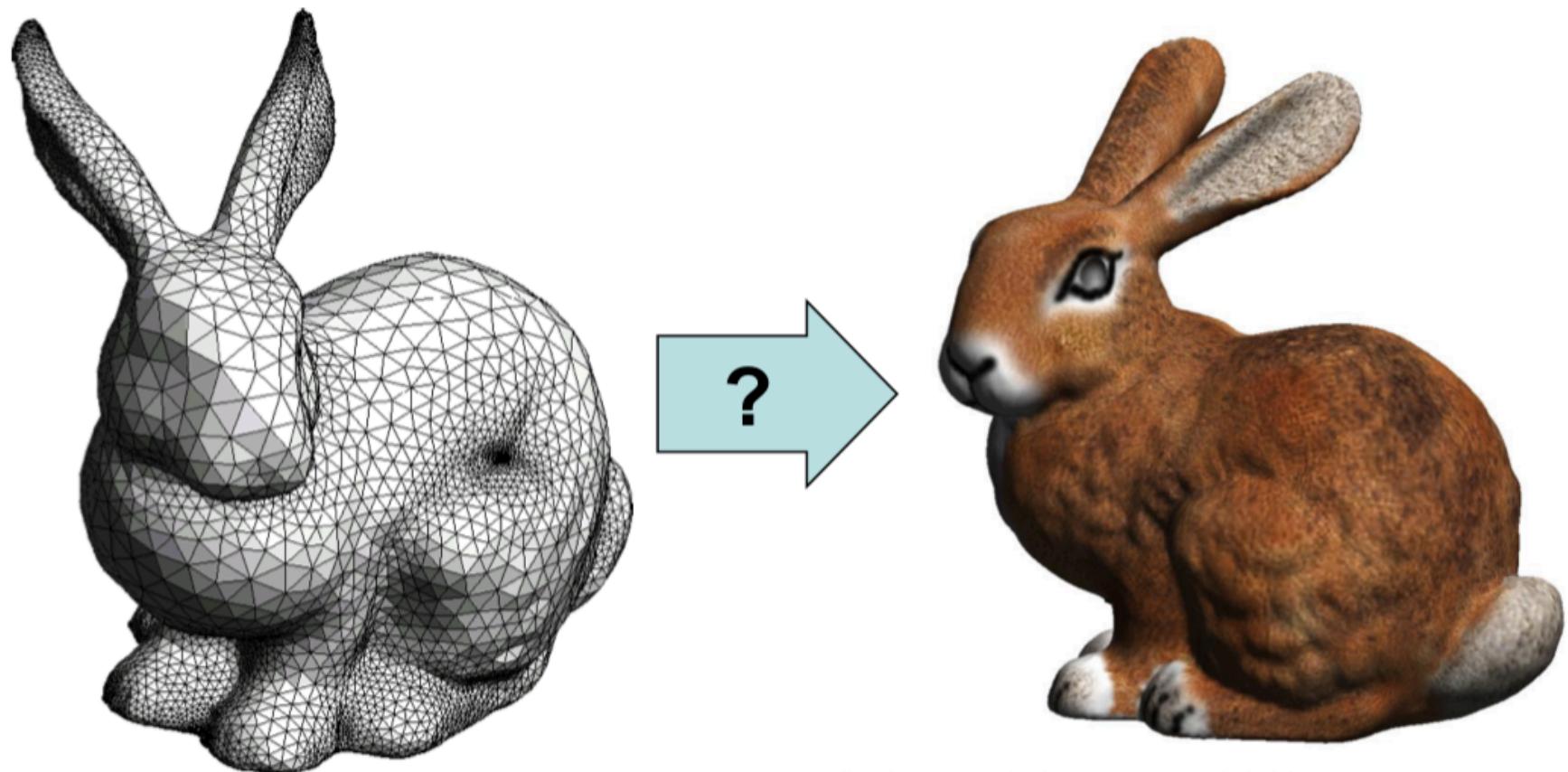
1. Procedents de dades
2. Procedurals (programes que donat un punt calculen un color/informació)



<https://www.textures.com>

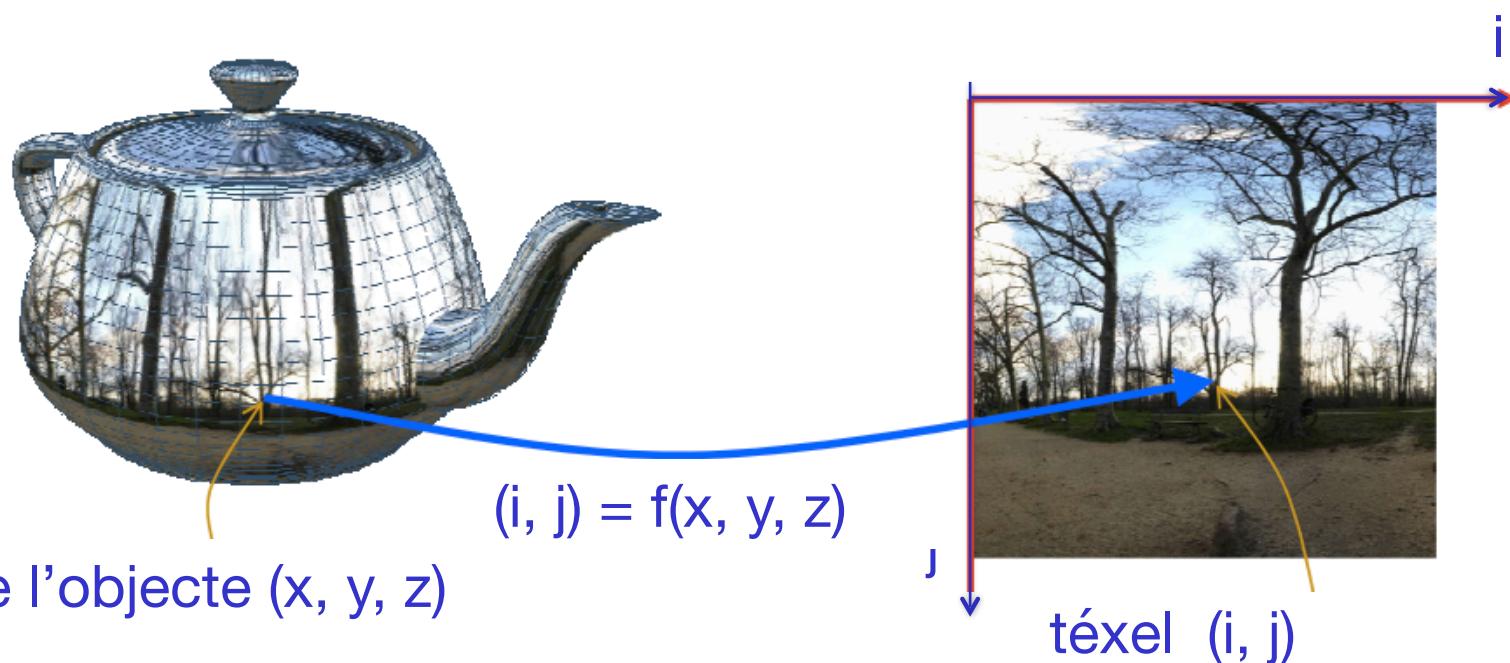


2.Com es mapejen les textures als objectes?



2. Com es mapejen les textures als objectes?

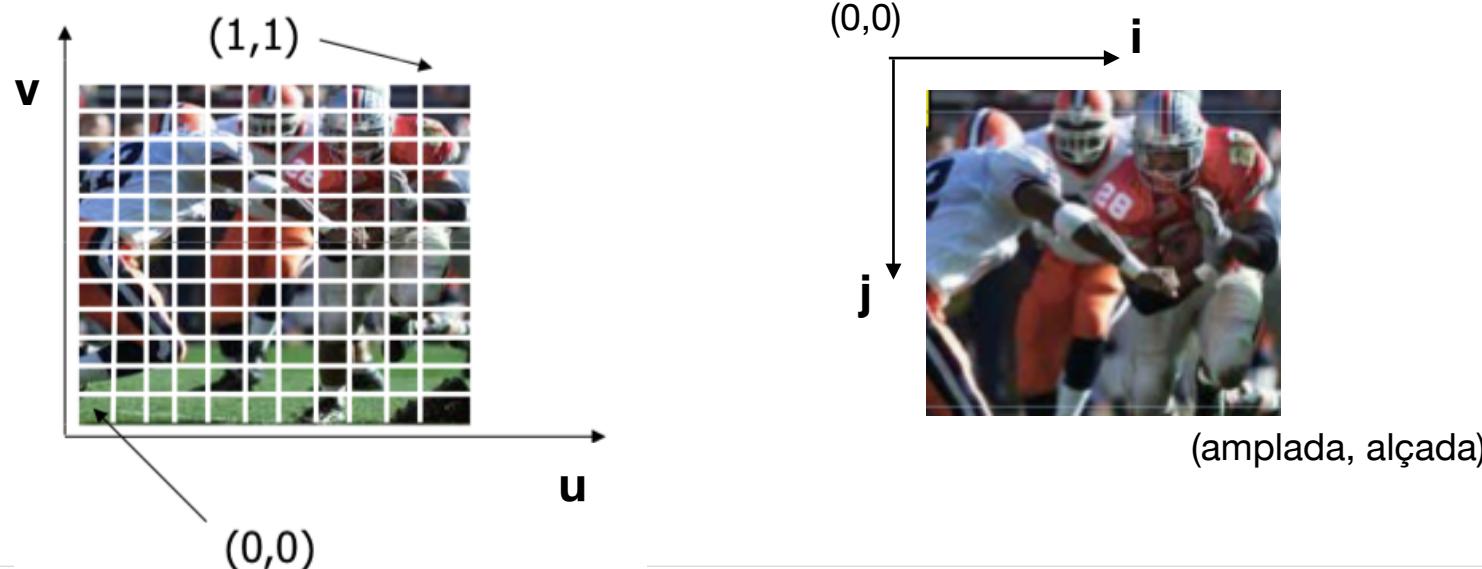
- 'Enganxar' o **mapear** (fer mapping) una textura sobre un objecte consisteix en definir una funció unívoca (**funció de mapping**) que a cada punt d'un objecte (x, y, z) li fa corresponder un punt d'una imatge (i, j).
- El mapping pot ser **directa** o en dues fases



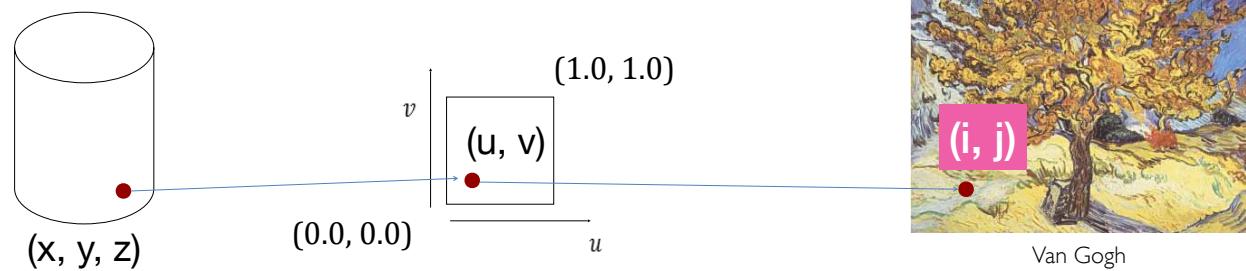
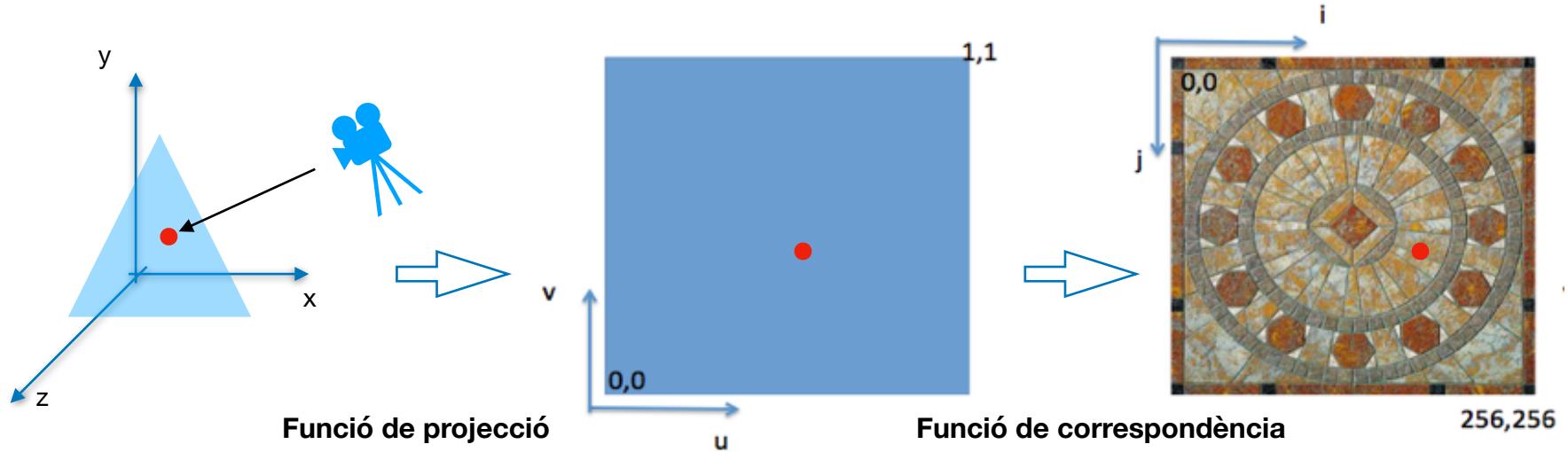
2. Com es mapejen les textures als objectes?

Notació:

- Suposem una textura com una imatge 2D [amplada x alçada]
- cada píxel (**téxel**) o (i, j) conté un color (R, G, B)
- les **coordenades de textura** (u, v) estan en el rang [0, 1]
- Per qualsevol (u, v) dins del rang [0, 1], només hi ha un únic (R, G, B) de la imatge



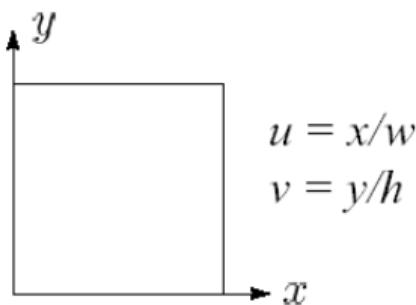
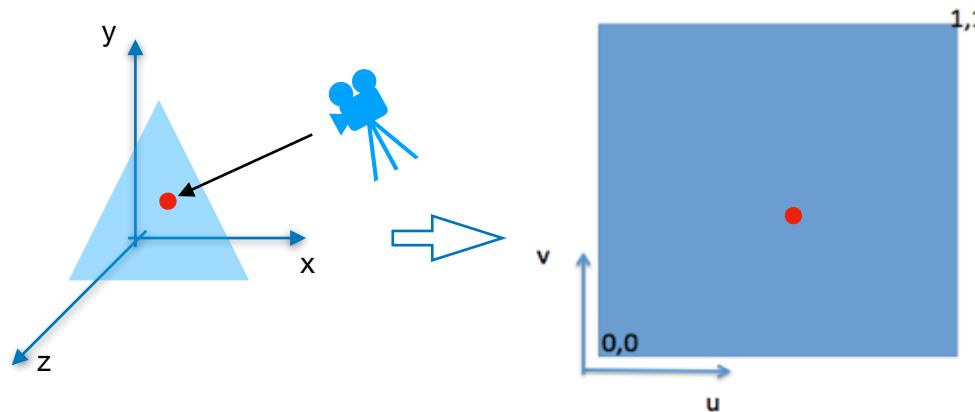
2. Com es mapejen les textures als objectes?



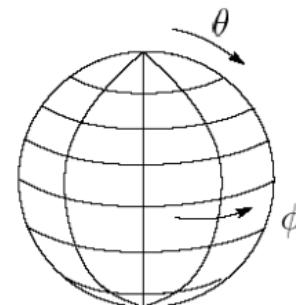
Funció de projecció

Funció de projecció: $(x, y, z) \rightarrow (u, v)$

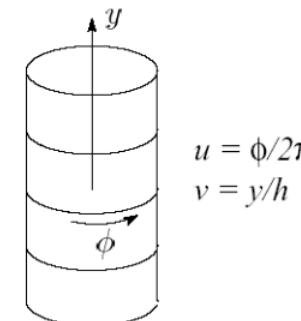
Per a cada punt de cada polígon de l'objecte, es calculen les coordenades (u, v) : Depèn del tipus d'objecte



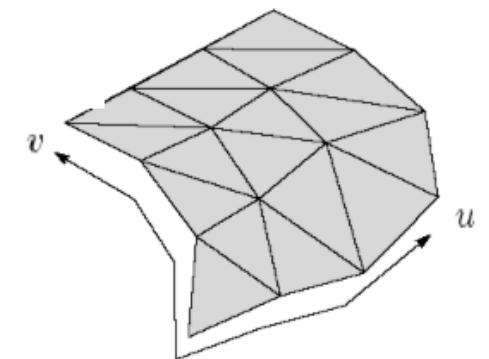
$$u = x/w$$
$$v = y/h$$



$$u = \phi/2\pi$$
$$v = \theta/\pi$$

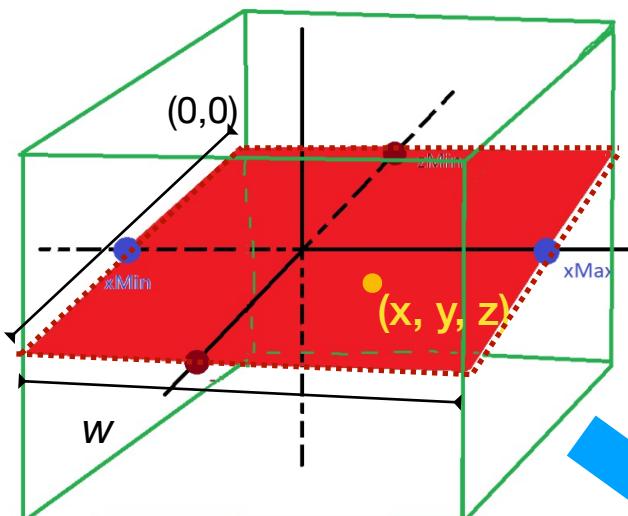


$$u = \phi/2\pi$$
$$v = y/h$$

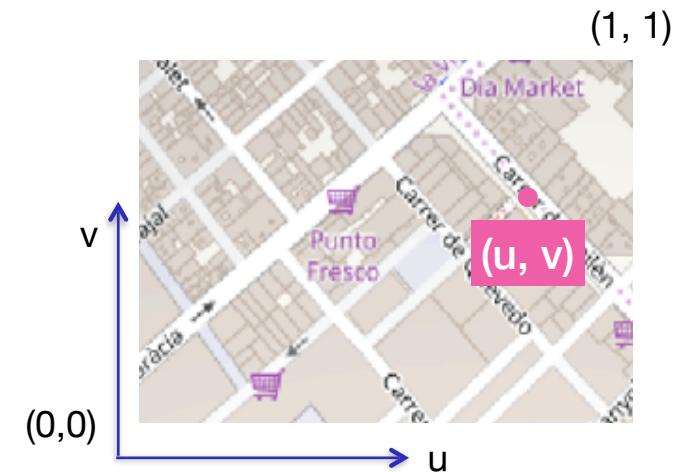
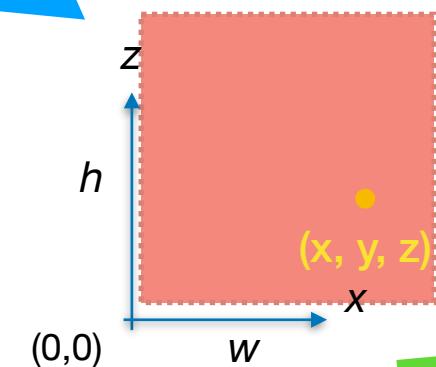


Funció de projecció

- **Pla:** $(x, y, z) \rightarrow (u, v)$



Thanks to Arnau Rovira



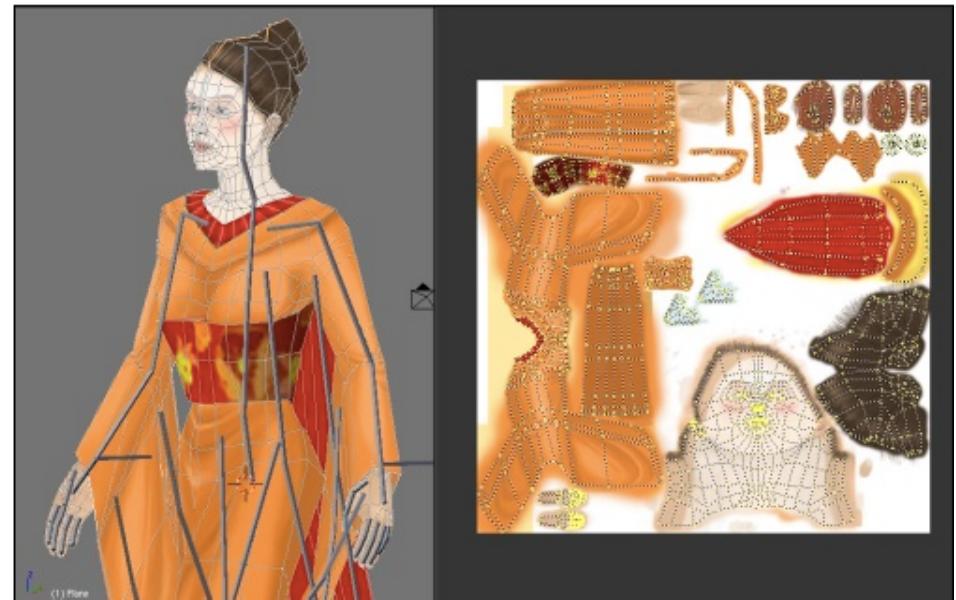
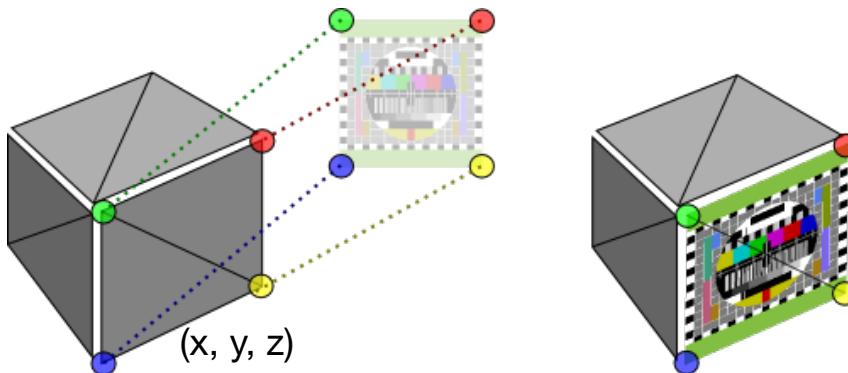
$$\begin{aligned} u &= x/w \\ v &= z/h \end{aligned}$$

Com seria si l'origen no fós el $(0, 0, 0)$?

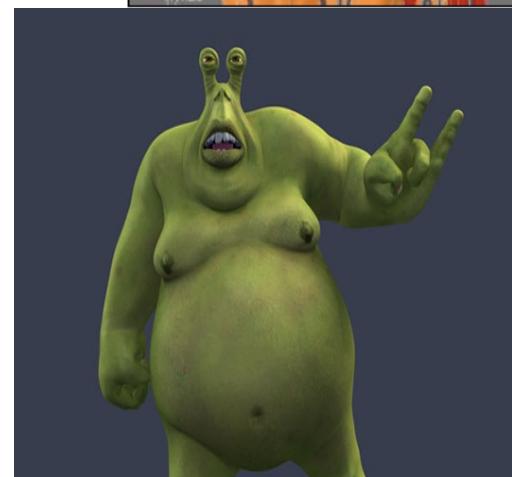
Funció de projecció

- **Pla:** $(x, y, z) \rightarrow (u, v)$

Traslació + Rotació

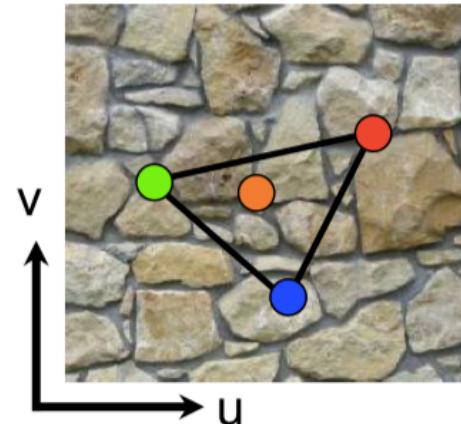
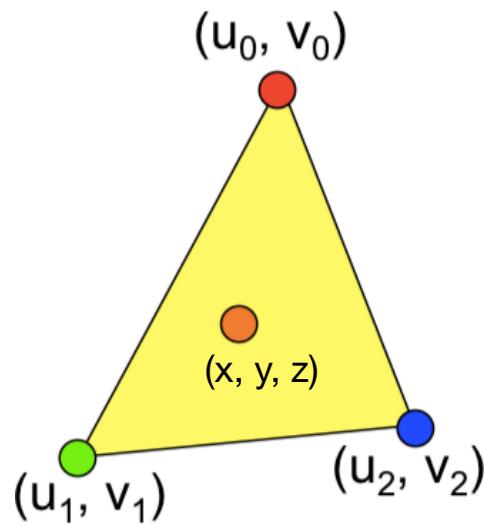


UV wrapping:
mètode automàtic per a
desplegar una superfície
en l'espai paramètric UV



Funció de projecció

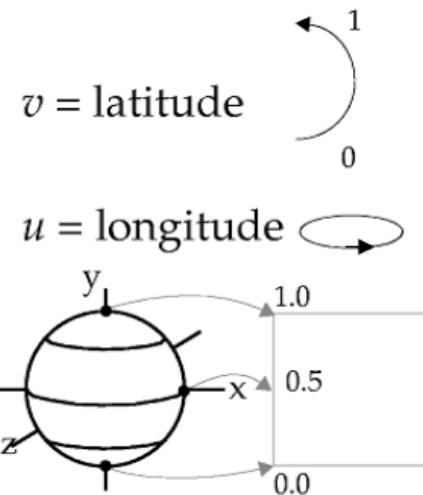
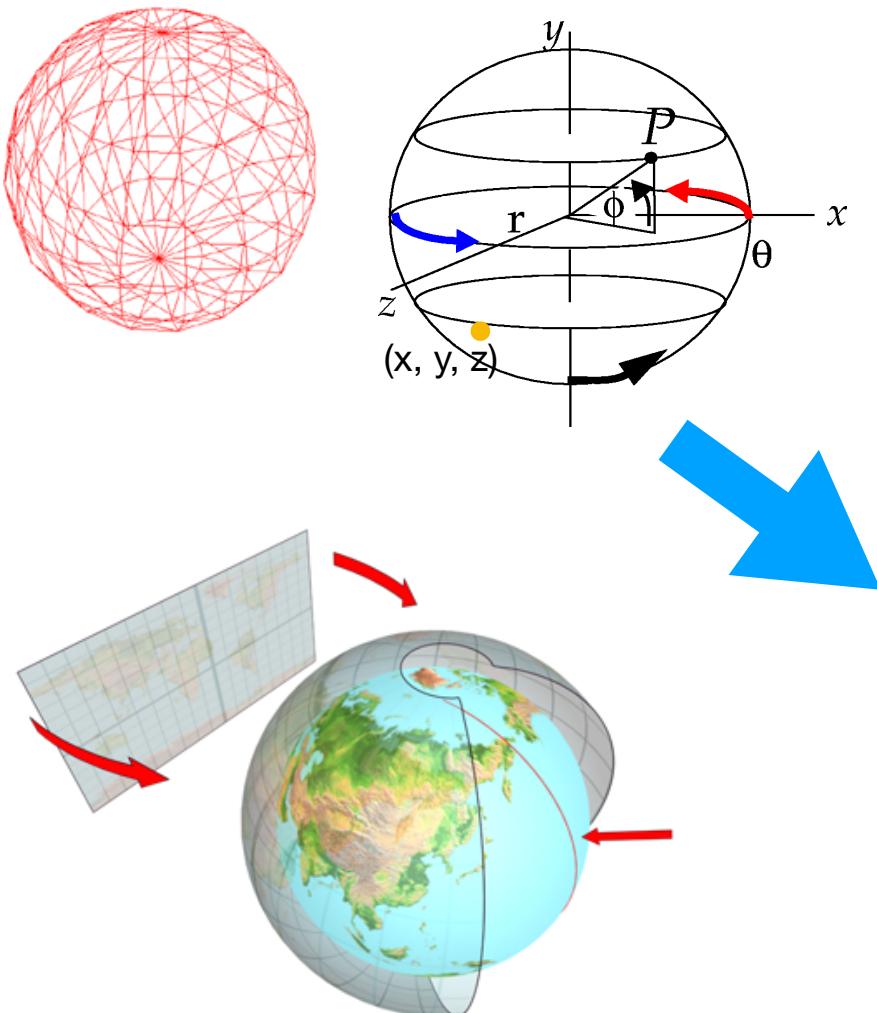
- **Triangle:** $(x, y, z) \rightarrow (u, v)$
- Per a cada vèrtex del triangle es tenen definides les coordenades (u, v)
- A cada punt interior del triangle s'interpol la seva coordinada (u, v) segons les seves coordenades baricèntriques



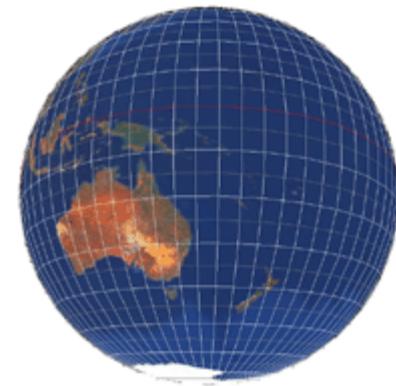
<https://www.geogebra.org/m/sUuDYe85>

Funció de projecció

- **Esfera:** $(x, y, z) \rightarrow (u, v)$

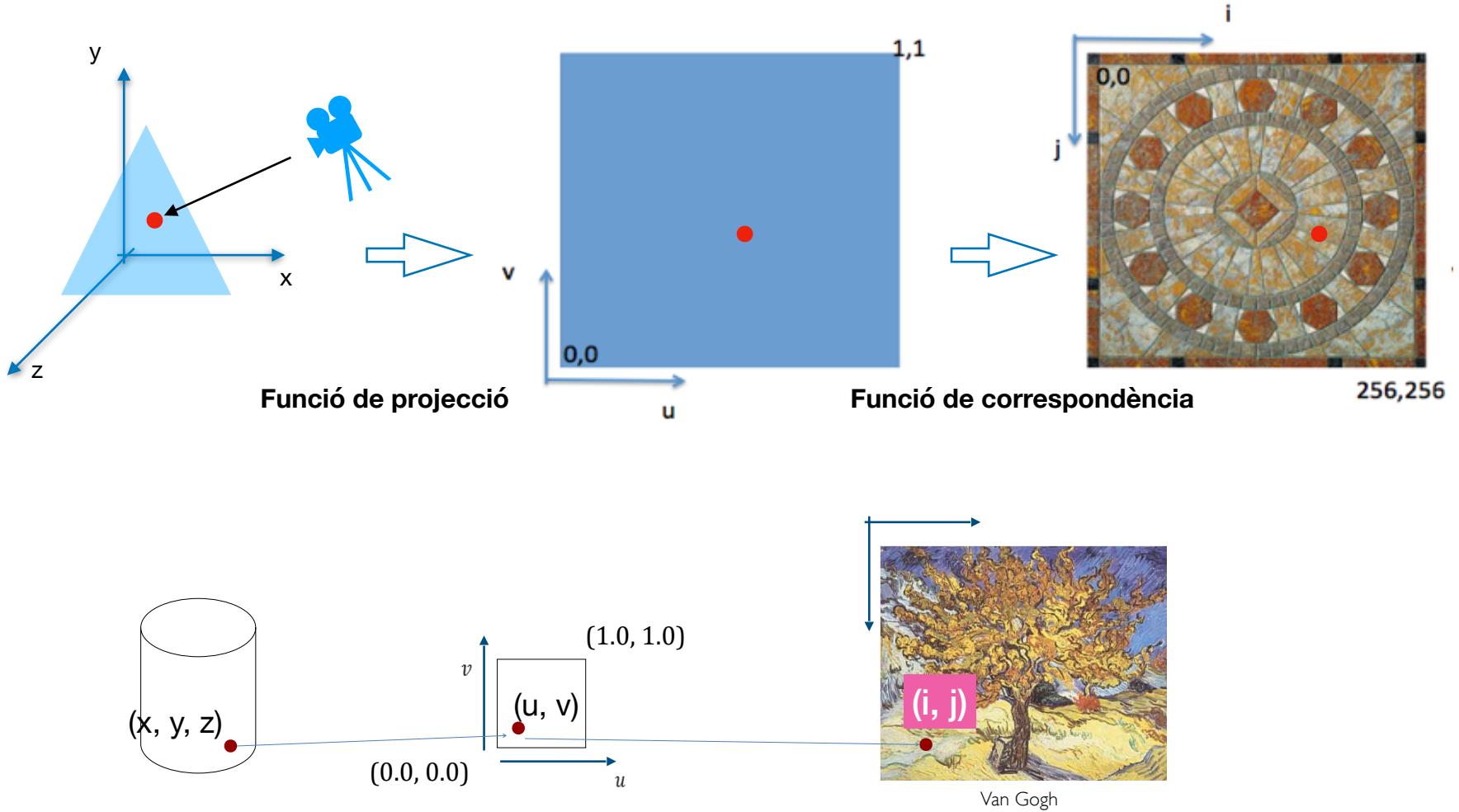


$\theta = \text{atan}2(z/r, x/r)$
 $u = 0.5 + \text{atan}2(z, x)/2\pi$ ó
 $u = 0.5 - \text{atan}2(z, x)/2\pi$?
 $\varphi = \text{asin}(y/r) \quad -\pi/2 \leq \varphi \leq \pi/2$
 $v = \varphi / \pi + 0.5$



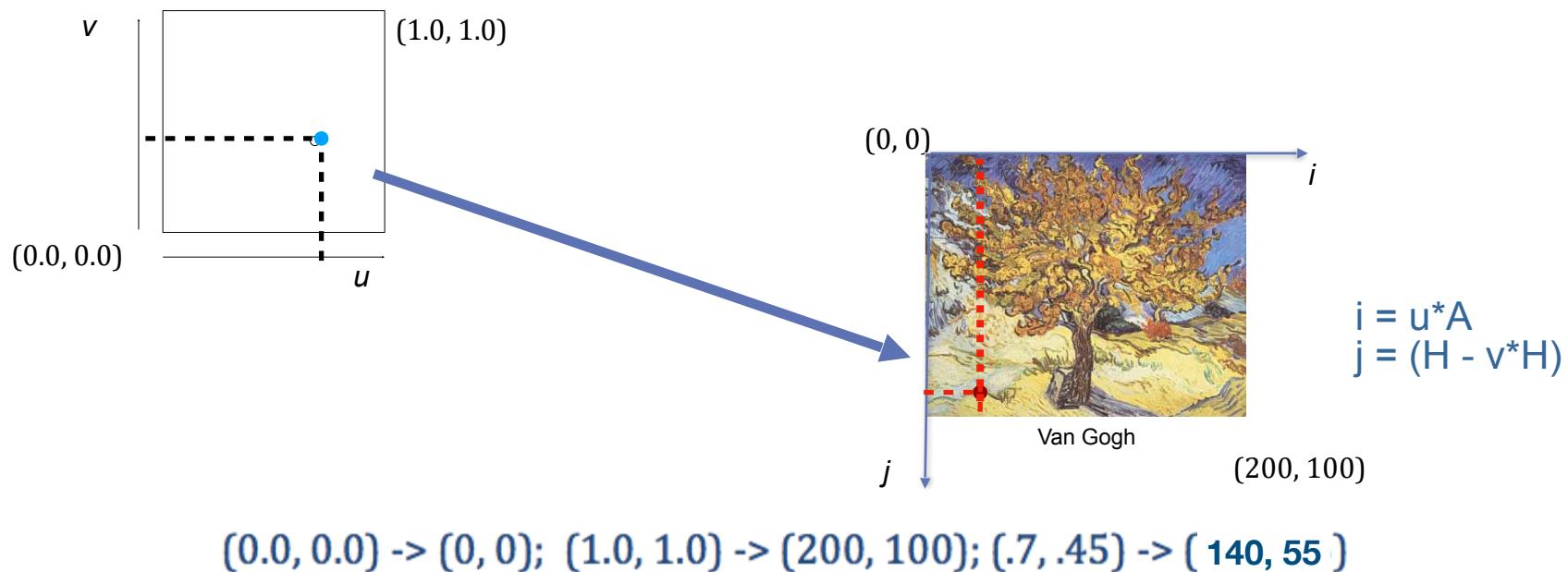
<https://www.geogebra.org/m/H5SAVTXy>

2. Com es mapejen les textures als objectes?



Funció de correspondència

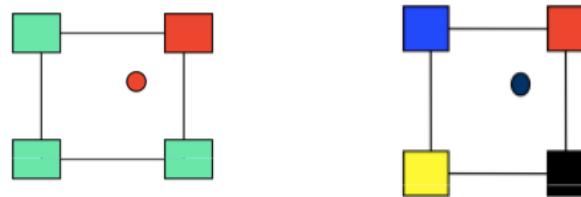
Funció de correspondència: $(u, v) \rightarrow (i, j)$



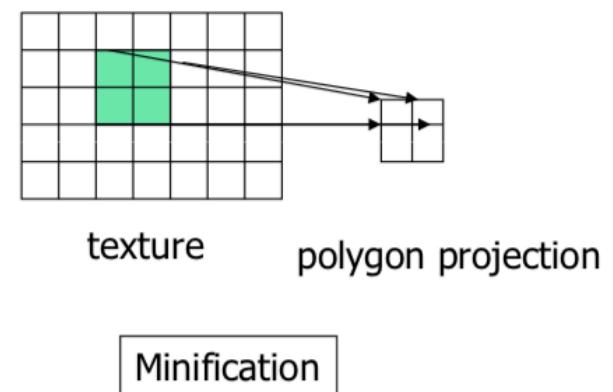
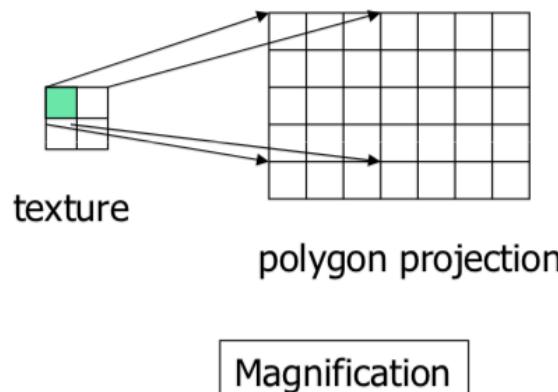
Funció de correspondència

Correspondència entre l'espai continu (u, v) i discret (i, j)

- Selecciona el píxel més proper
 - Promig dels 4 píxels veïns al punt

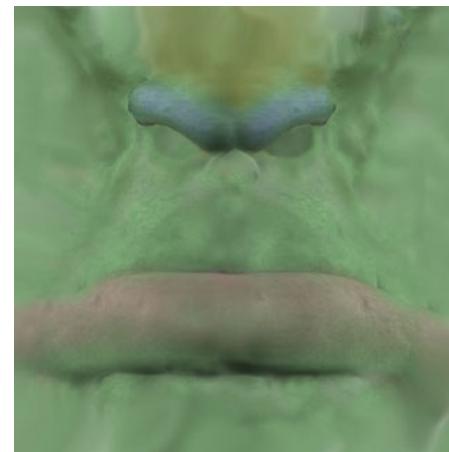


- Aquest fet provoca problemes **d'aliasing**:
 - Varis píxels de pantalla corresponen a un mateix téxel ($\text{t\'exels/p\'ixels} < 1$) (MAGNIFICATION)
 - Varis téxels cauen en un píxel de pantalla ($\text{t\'exels/p\'ixels} > 1$) (MINIFICATION)

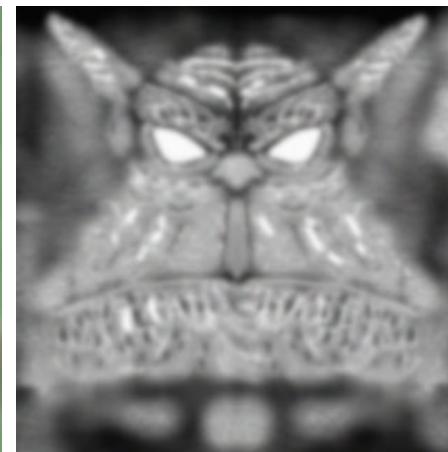


3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?

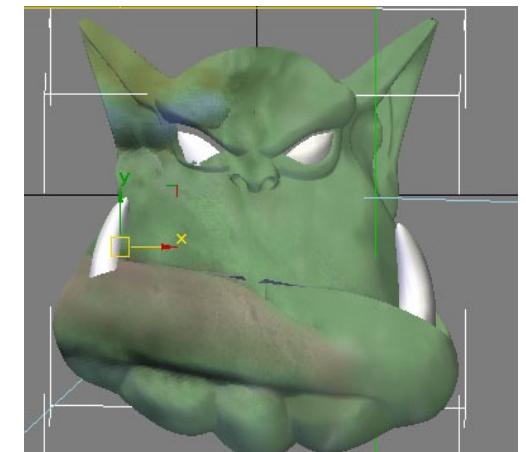
Amb el RGB de la textura es poden variar diferents propietats òptiques



Color



Especularitat



Resultat



Normals



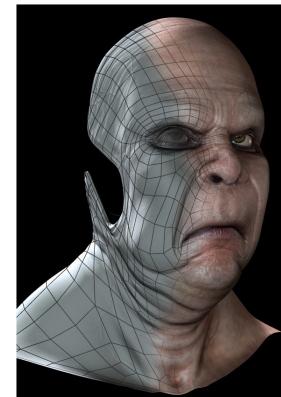
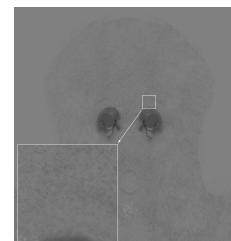
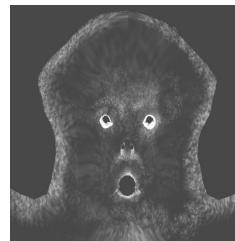
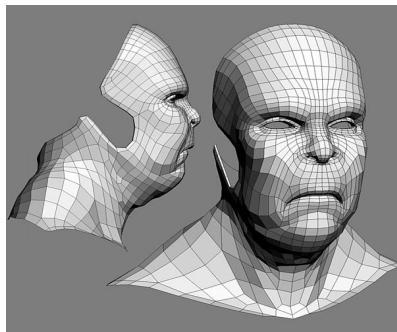
Resultat final
Semestre Primavera 2022-23 • 58

3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?

- Aplicació del valor de textura a la il·luminació

Funció de transformació del valor

Modificar valor en l'equació d'il·luminació



3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?

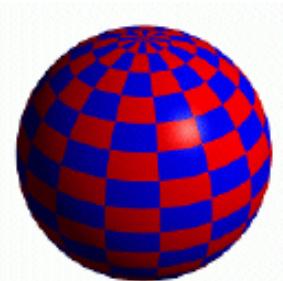
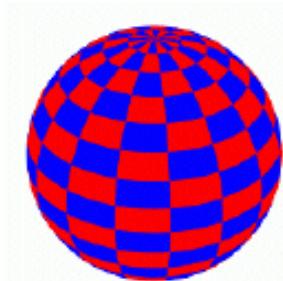
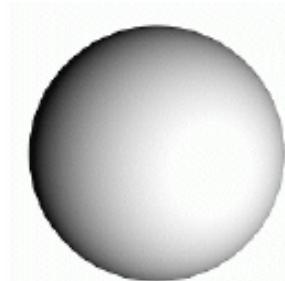
- Aplicació del valor de textura a la il·luminació

Funció de transformació del valor

Modificar valor en l'equació d'il·luminació

- Funcions de transformació

- Reemplaçar
 - Modular o blending
 - (AND, OR, ...)



- Serveixen per fer la composició de diferents textures amb diferents materials o propietats

3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?

- Aplicació del valor de textura a la il·luminació

Funció de transformació del valor

Modificar valor en l'equació d'il·luminació

- Quins components de l'equació de Blinn-Phong es poden modificar amb el valor de la textura?

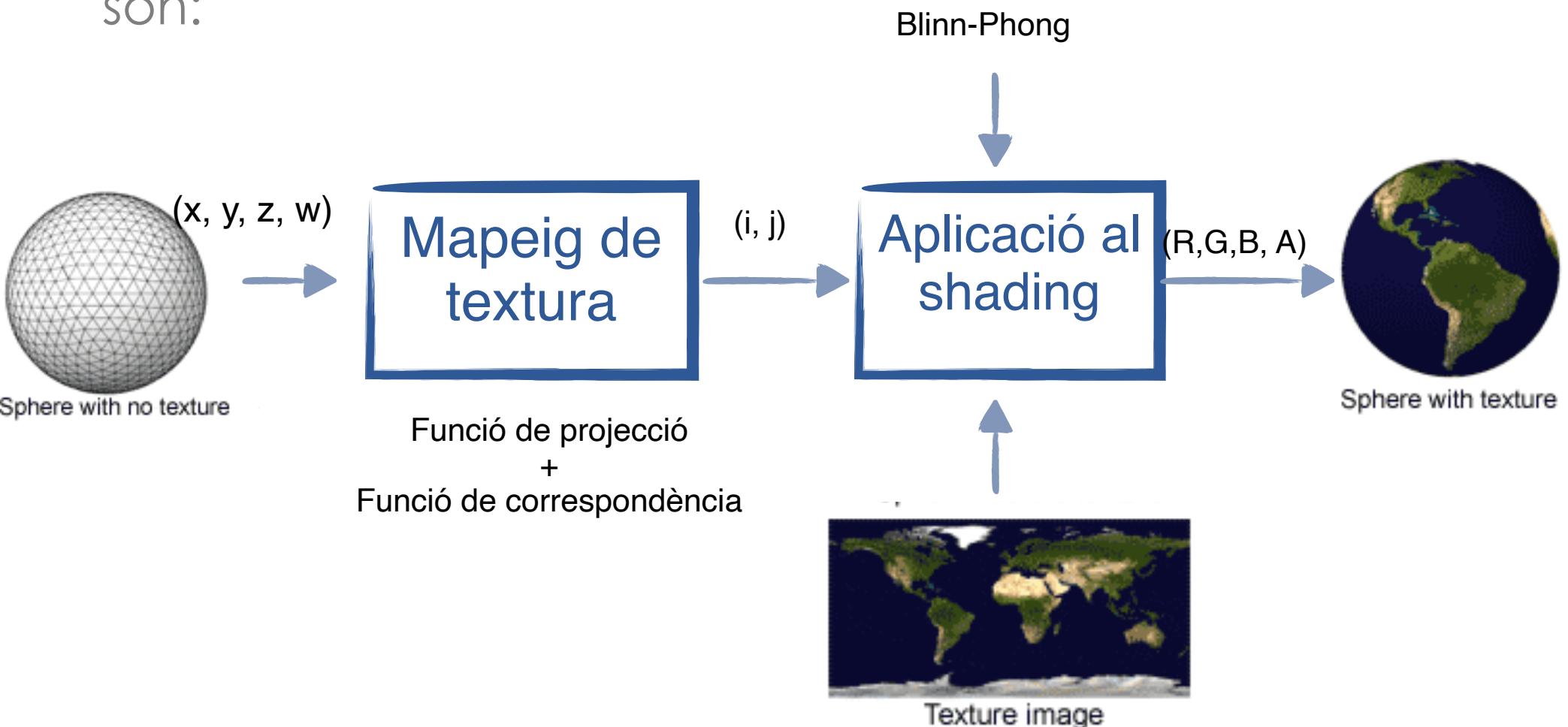
$$I_{total} = I_{a_{global}}K_a + \sum_{i=1}^{numLlums} \frac{1.0}{a_i + b_id_i + c_id_i^2} (I_{d_i}K_d \max(L_i \cdot N, 0.0) + I_{s_i}K_s \max((N \cdot H_i), 0.0)^\beta) + I_{a_i}K_a$$

- La component difusa K_d (color)
- El vector normal (tècnica de bump-mapping)
- La component especular K_s

2.3.2.5. Textures

Resum

Les etapes a realitzar per a aplicar textures a un objecte són:

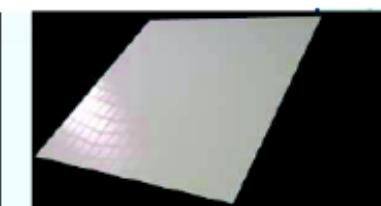
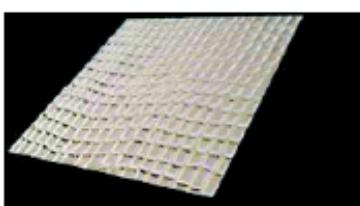
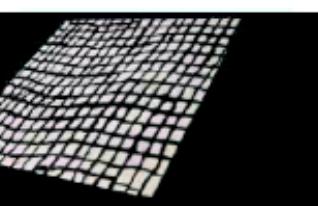
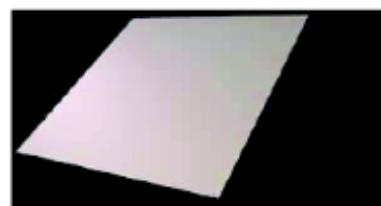


3. Com es poden usar les textures per a controlar l'aspecte de l'objecte?

- Aplicació del valor de textura a la il·luminació

Funció de transformació del valor

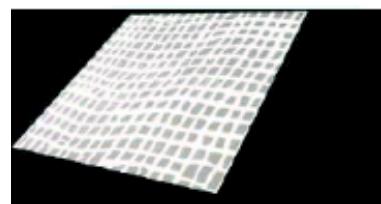
Modificar valor en l'equació d'il·luminació



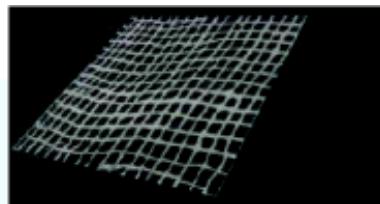
Kd

Normal

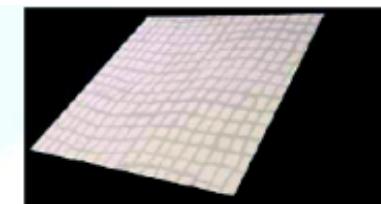
coef. especular



Emitància

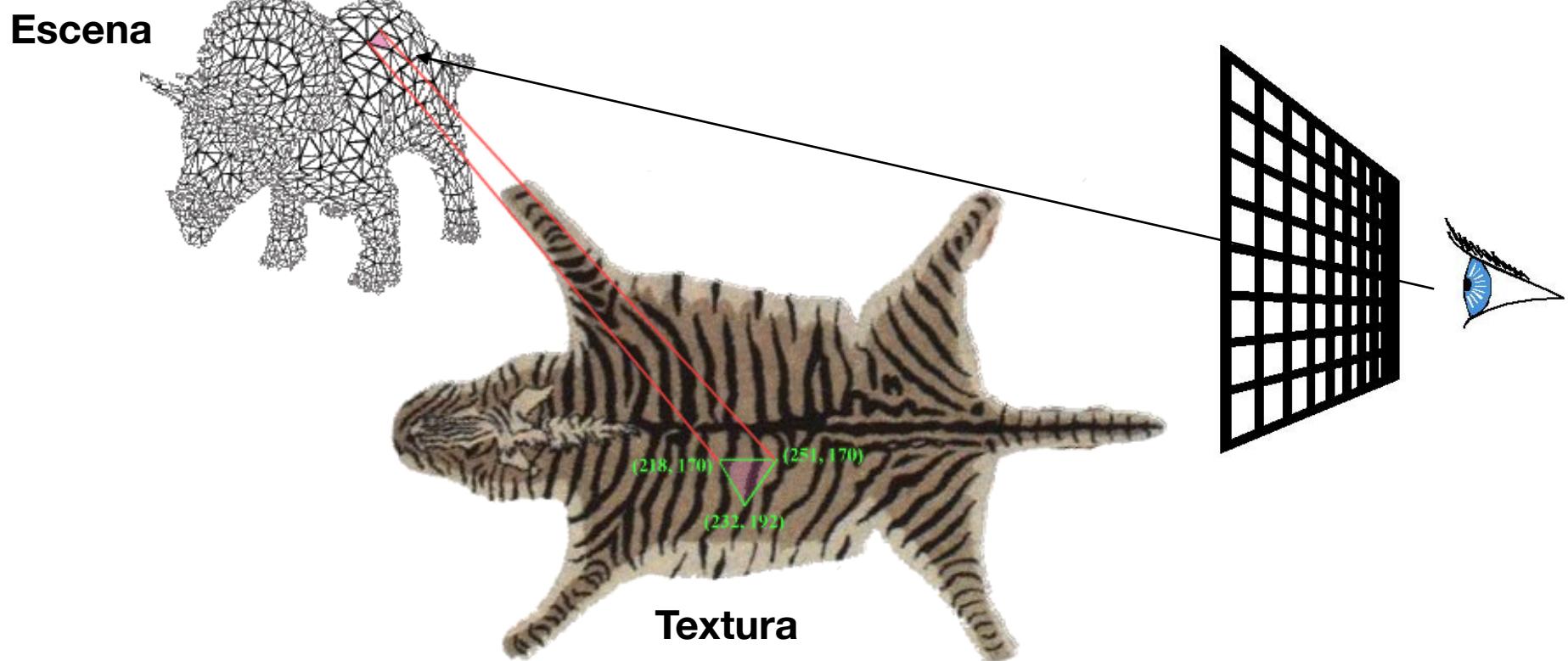


Kt (alfa)

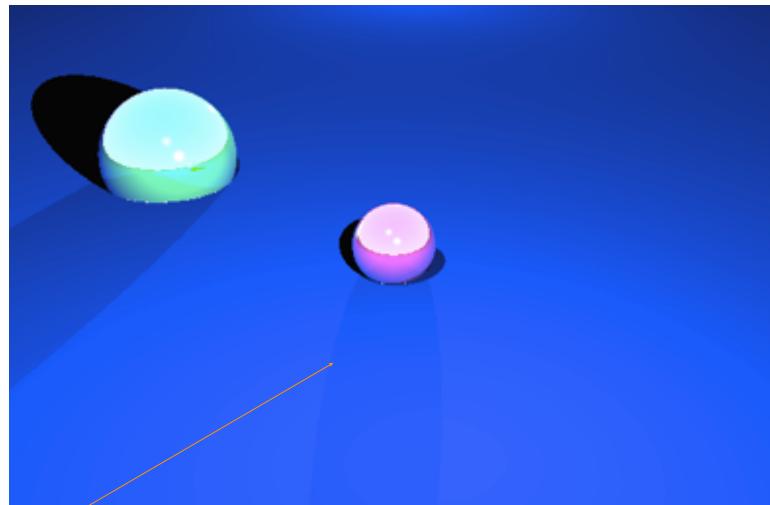


Ks

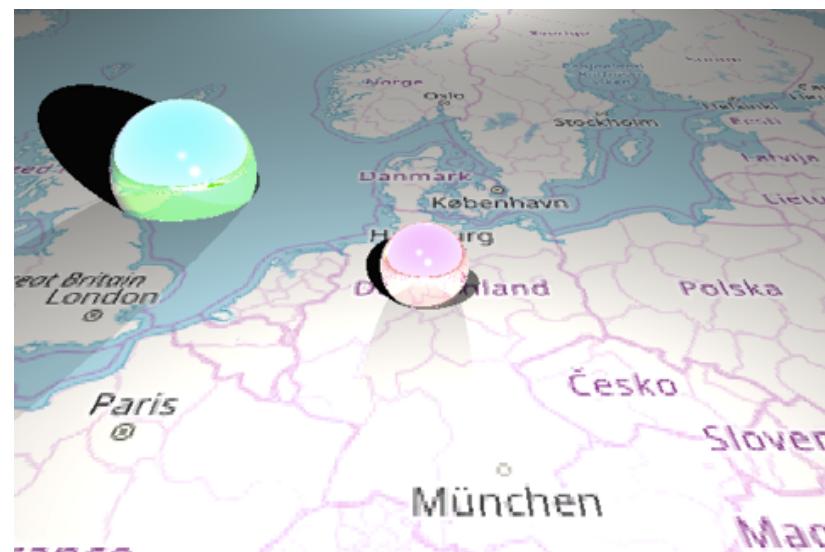
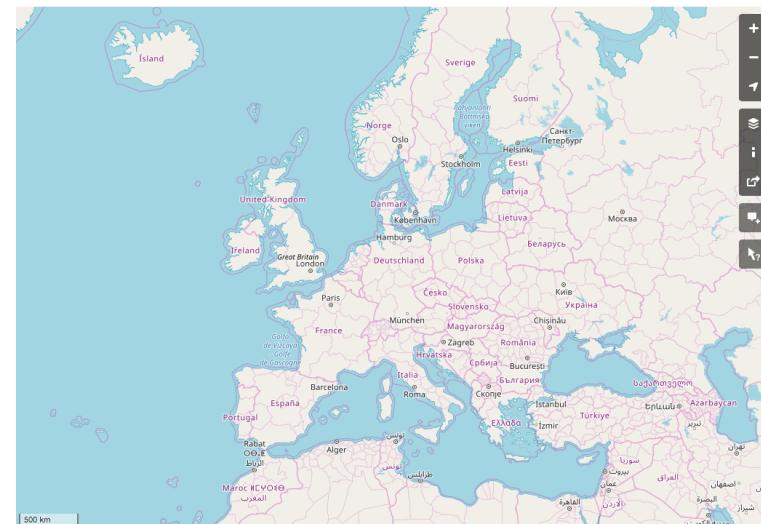
4. Com s'integren en l'algorisme de Raytracing?



4. Com s'integren en l'algorisme de Raytracing?

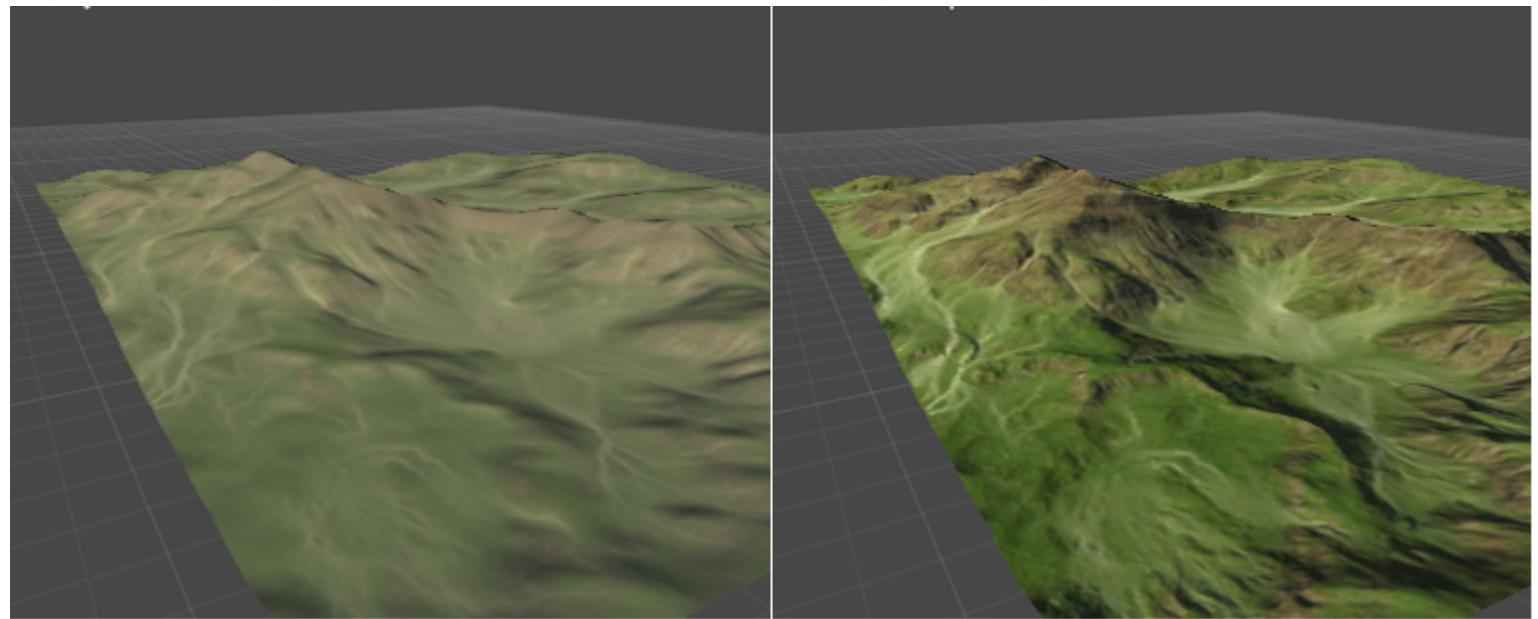
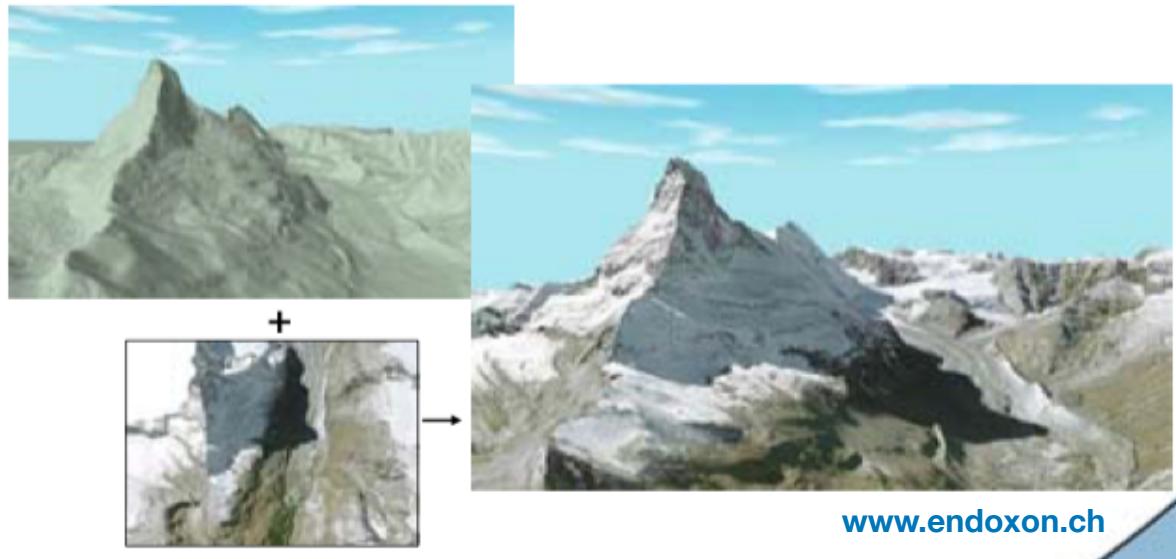


+



MaterialTextura:

té un mètode
`getDiffuse(u, v)` que retorna el
color del pixel de la textura
correspondent al valor (u,v). El seu
mètode `getOneRayScattered`
no retorna cap raig de sortida





Sintel
Blender Open Movie