

Gràfics i Visualització de Dades

T3: Mètodes projectius: Textures

Anna Puig

NOTA: Aquestes transparències es corresponen a les explicacions del tema **7.5** del llibre de referència bàsica. En la secció **7.6** trobareu l'explicació de textures en OpenGL.

[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

Índex

3.1. Z-Buffer: Pipeline de visualització



3.2. Pipeline integrat a OpenGL

3.3. Il·luminació usant shaders

3.4. Textures



3.5. Reflexions i trac

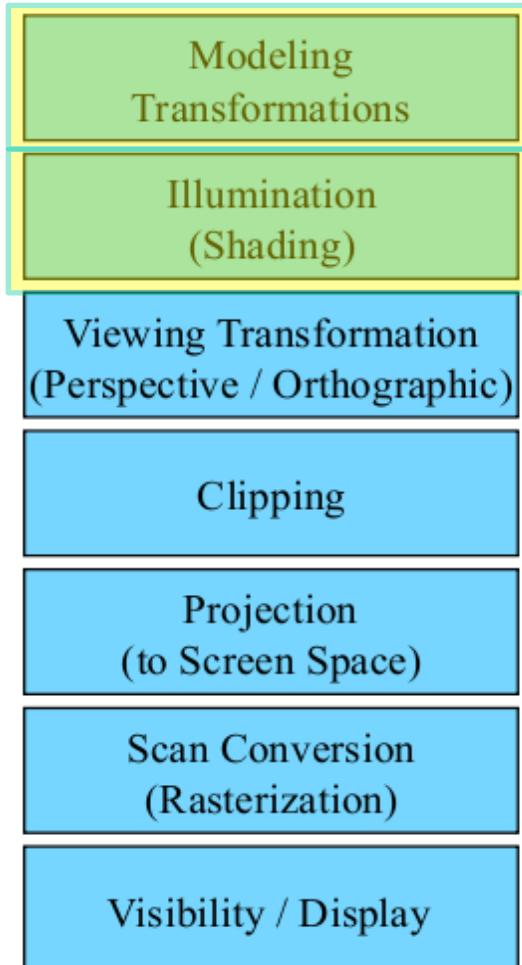
3.6. Usos avançats



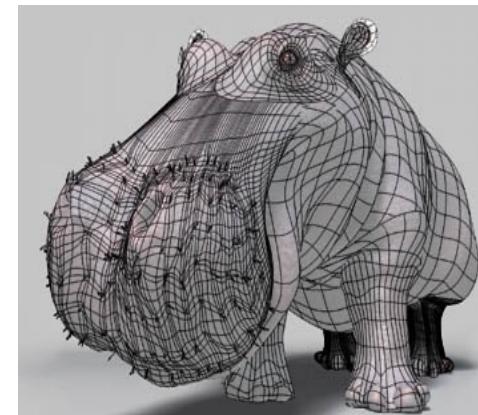
by Rami Ali Al-ashqar

3.4. Textures

Objectiu: incrementar el grau de realisme amb petits detalls a la superfície. Com es pot aconseguir?



- en la fase de **modelatge**: incrementant punts i triangles a la superfície

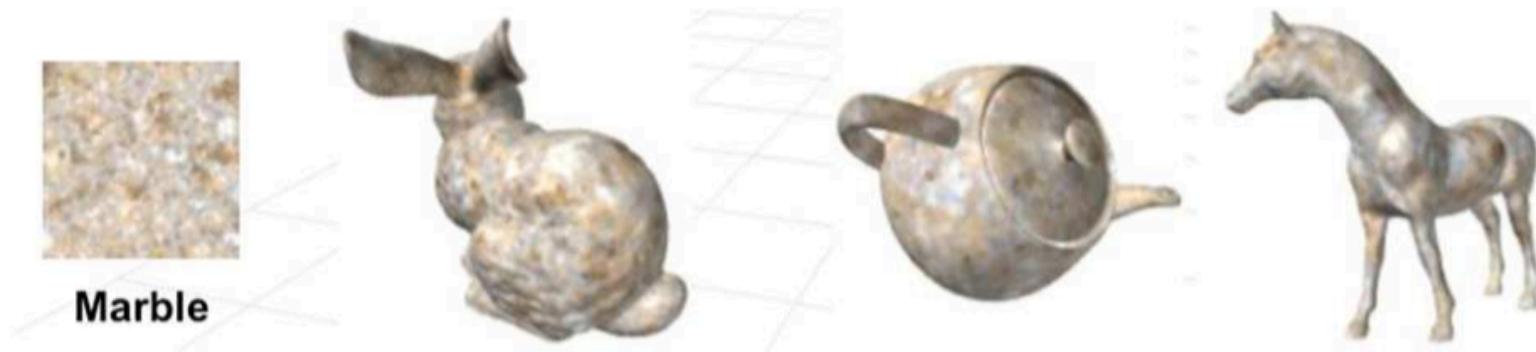


- en la fase de **shading**: ús de materials i textures



Els objectes reals no són uniformes en termes de color i normals: existeixen detalls amb altes freqüències que no són directament modelables.

3.4. Textures



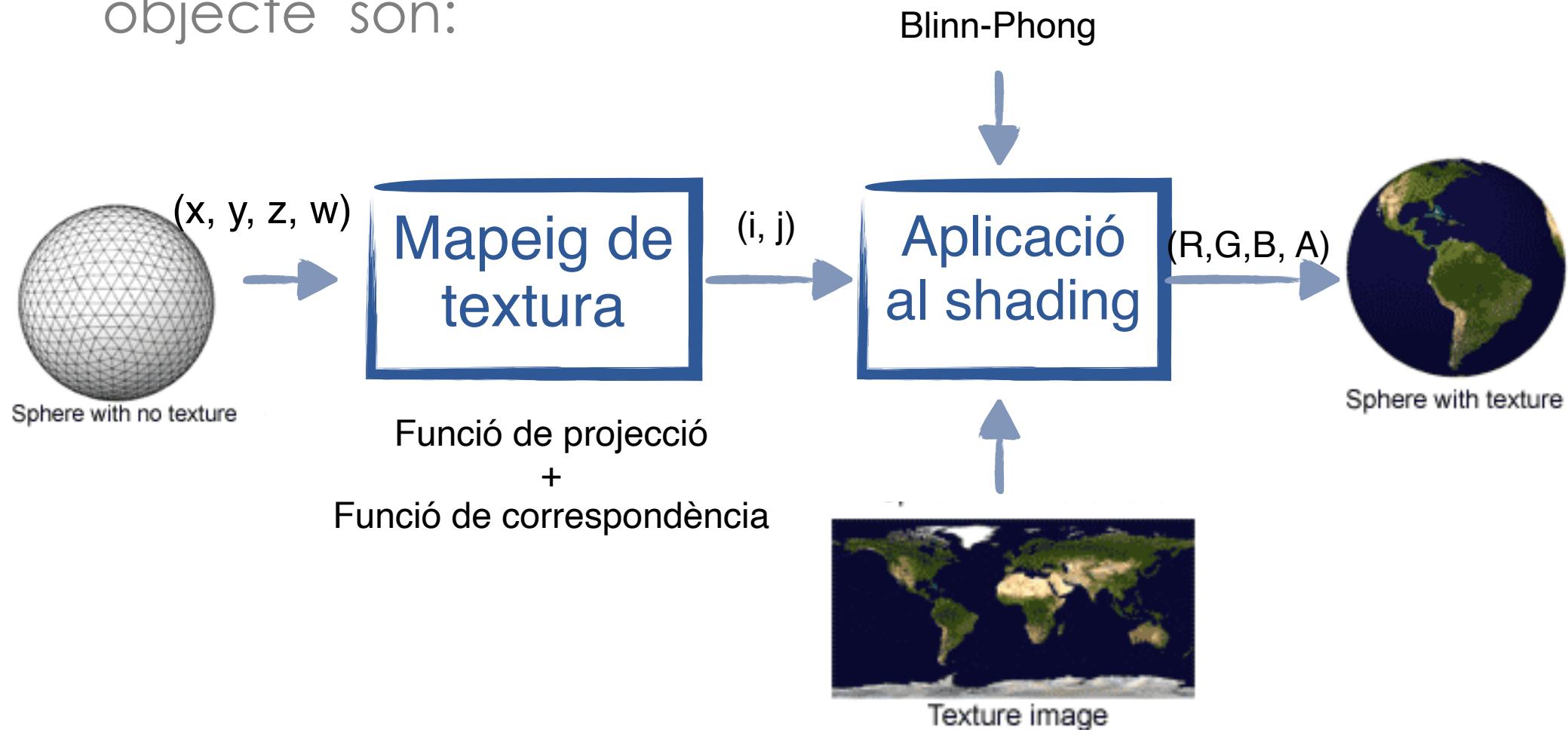
Qüestions bàsiques:

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

3.4. Textures

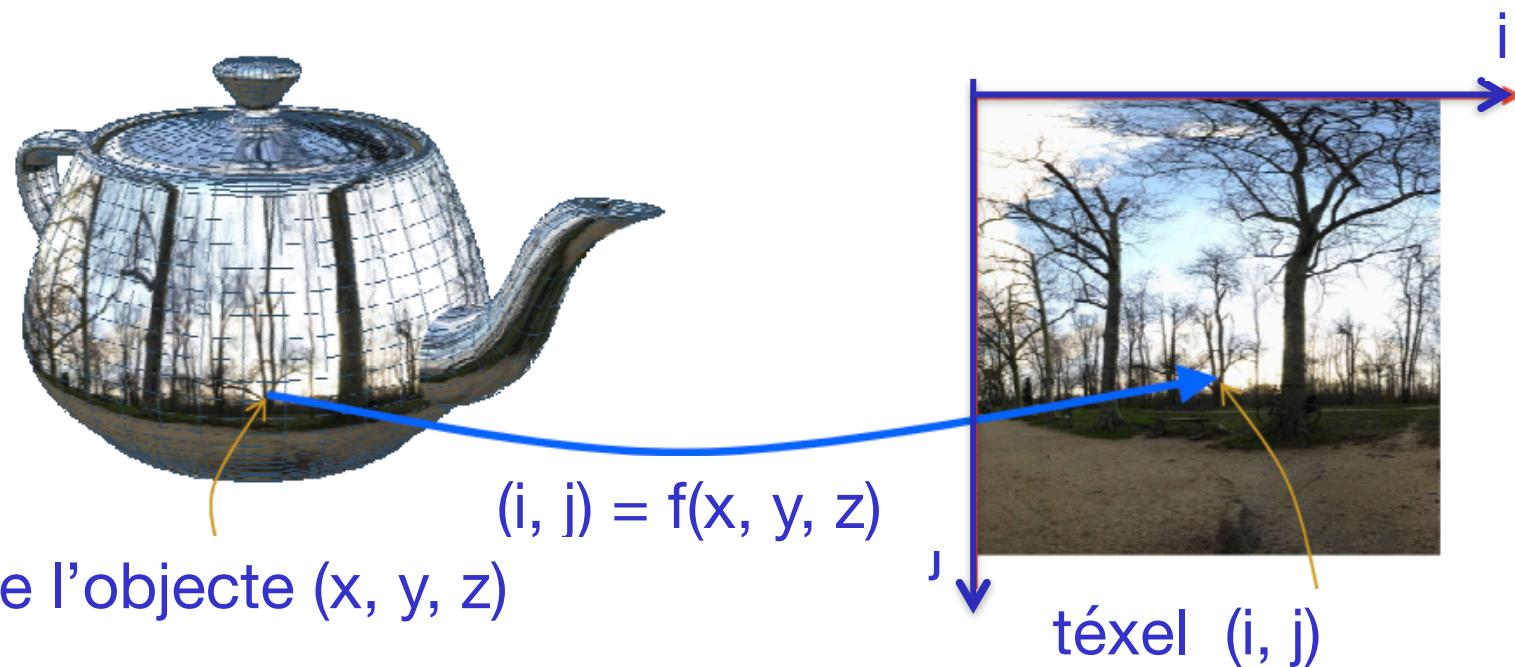
Com s'apliquen les textures? (tema 2b)

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



3.4. Textures: Mapeig

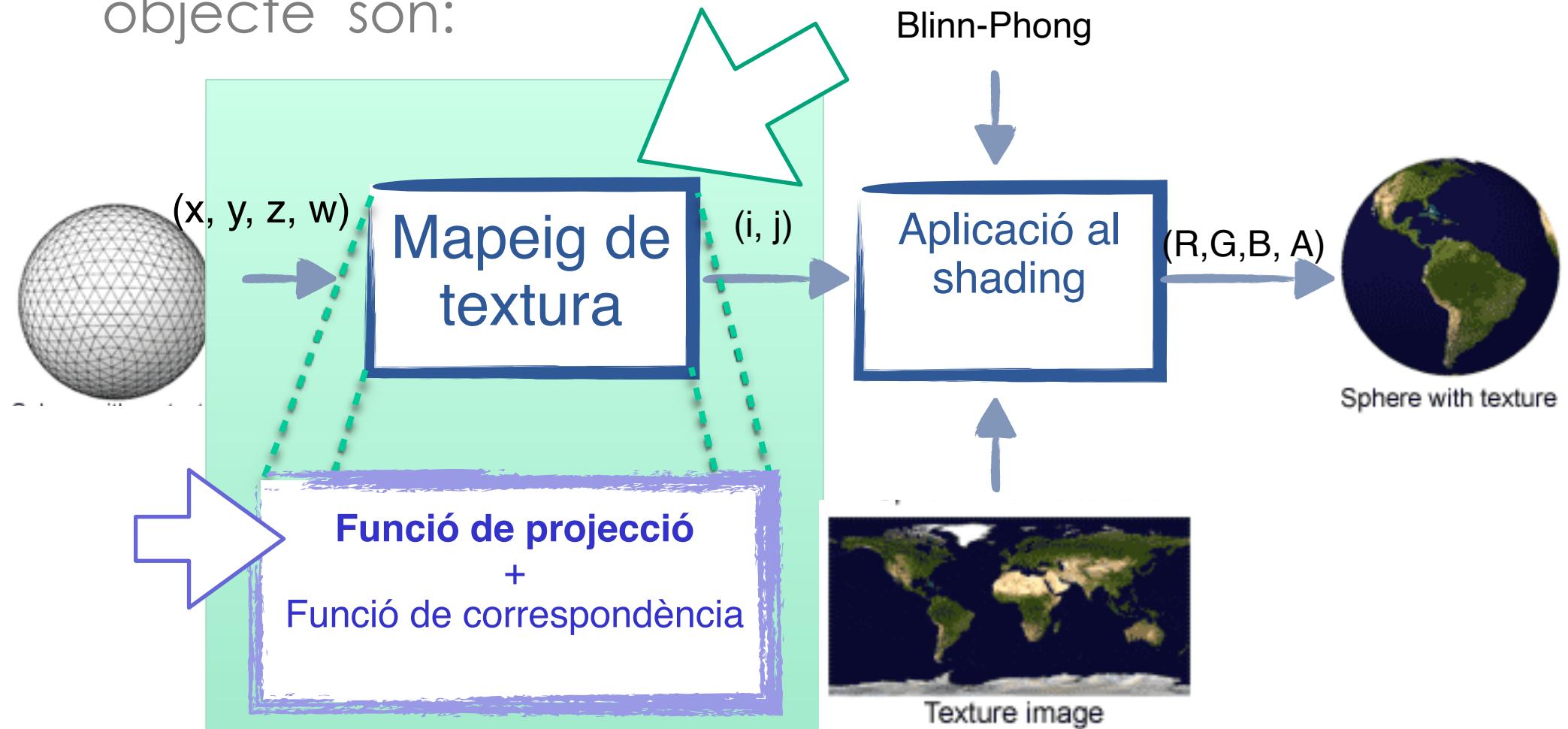
- 'Enganxar' o fer el **mapping** d'una textura sobre un objecte consisteix en definir una funció unívoca (*funció de mapping*) que a cada punt d'un objecte (x, y, z) li fa corresponder un punt d'una imatge (i, j).
- El mapping pot ser **directa** o en **dues fases**



3.4. Textures

Com s'apliquen les textures? (tema 2b)

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

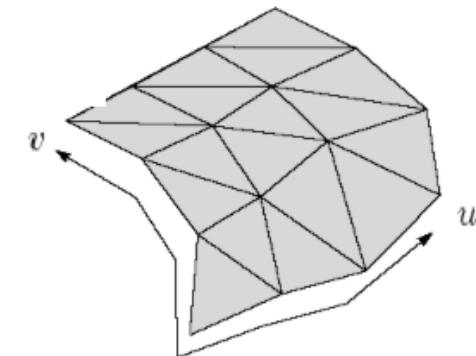
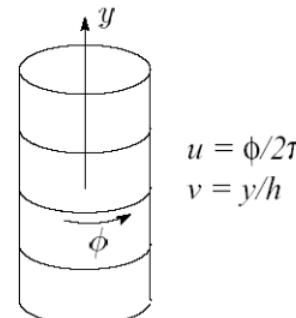
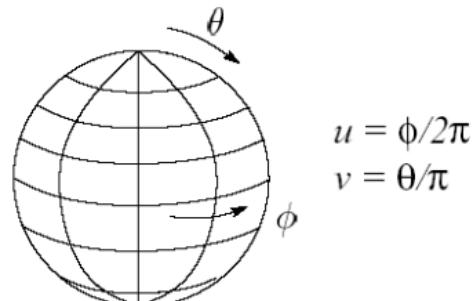
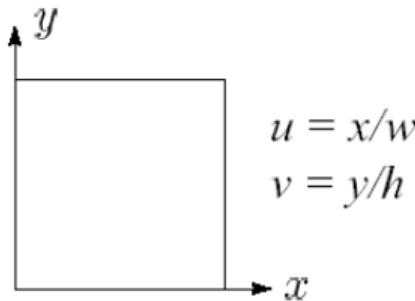


3.4. Textures: funció de projecció

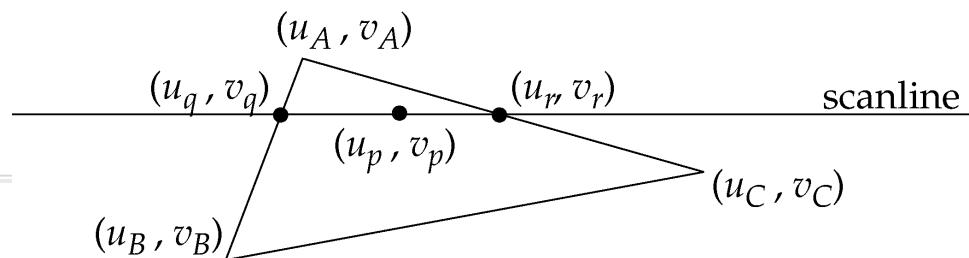
Funció de projecció: $A : (x, y, z) \rightarrow (u, v)$

Les coordenades (x, y, z) d'un objecte es poden escriure en coordenades paramètriques (u, v) mitjançant la funció de projecció A:

1. Per a cada vèrtex de cada polígon de l'objecte, es calculen les coordenades paramètriques (u, v)

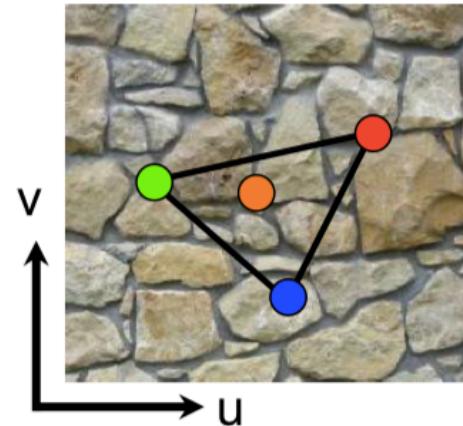
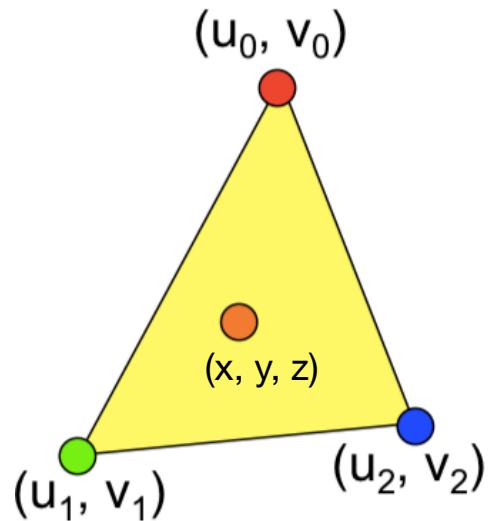


2. Per a cada **punt interior del polígon**, es calculen les coordenades paramètriques a partir de les coordenades paramètriques dels vèrtexs extrems del polígon



3.4. Textures: funció de projecció

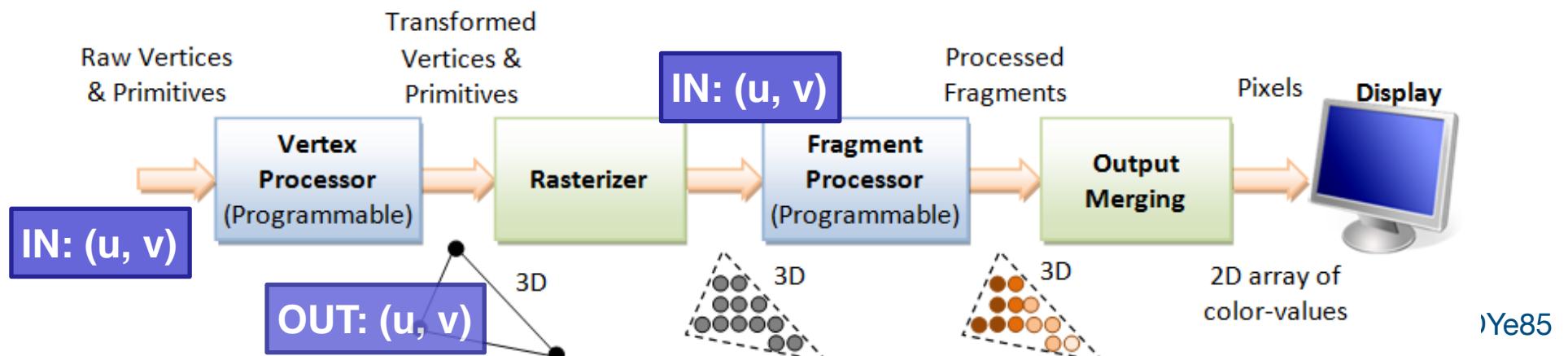
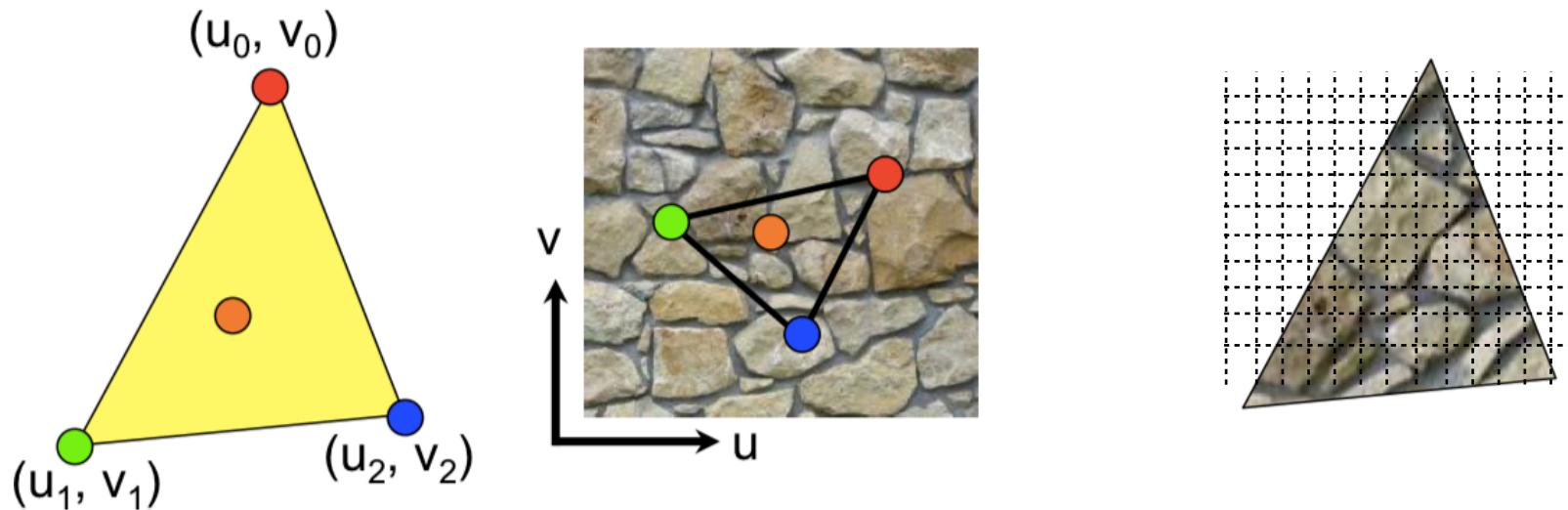
- **Triangle:** $(x, y, z) \rightarrow (u, v)$
- Per a cada vèrtex del triangle es tenen definides les coordenades (u, v)
- A cada punt interior del triangle s'interpolen les seves coordenades (u, v) segons les seves coordenades baricèntriques



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

3.4. Textures: funció de projecció

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



Ye85

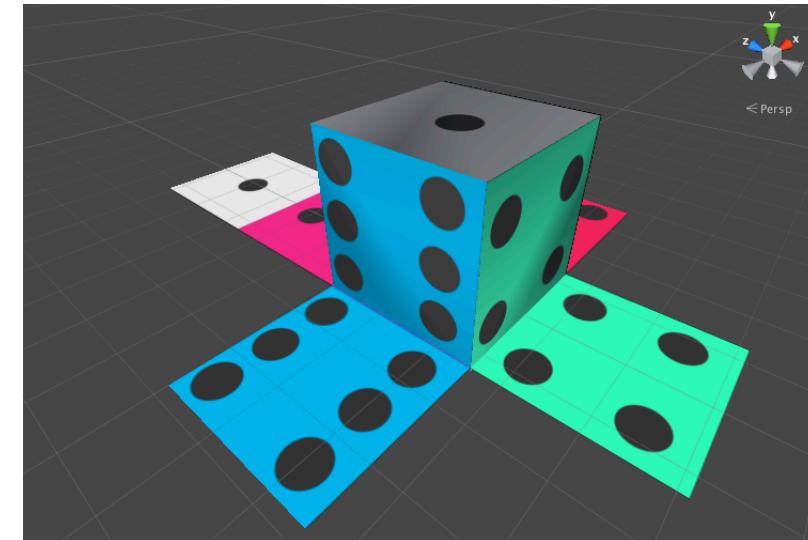
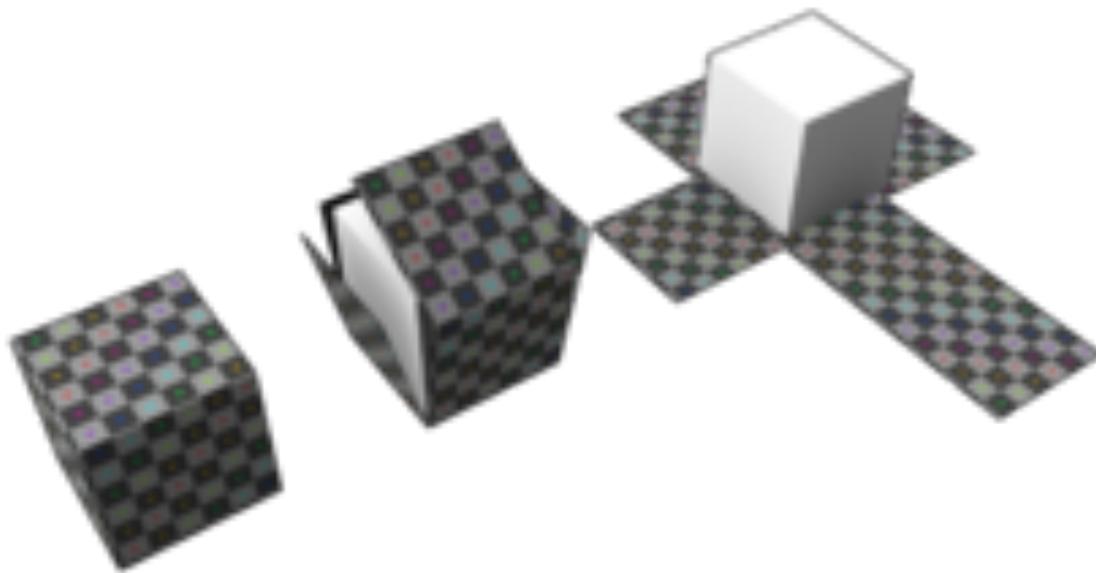
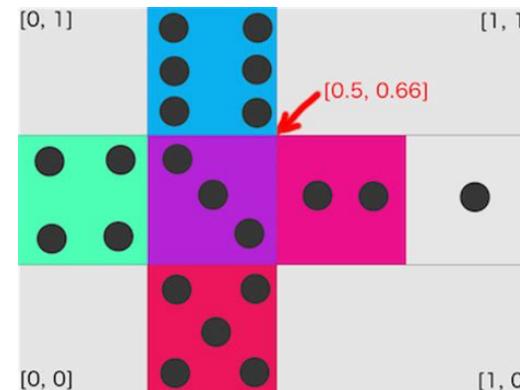
3.4. Textures: funció de projecció

- **Malles poligonals:**

Definir (u, v) a cada vèrtex

UV unwrapping:

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



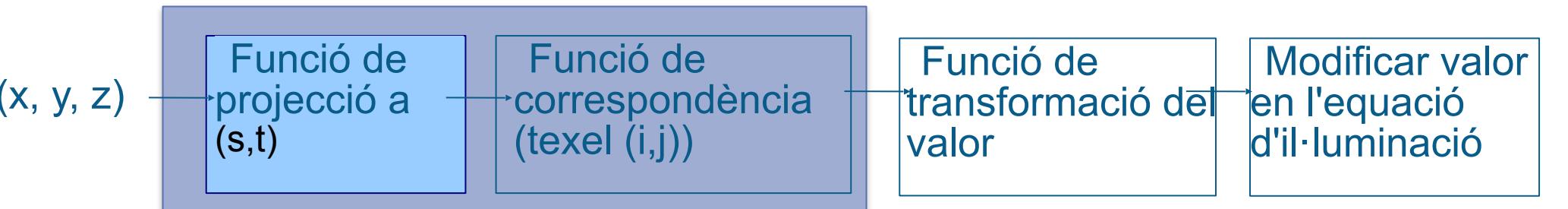
3.4. Textures: funció de projecció

- Procés de **UV-unwrapping**

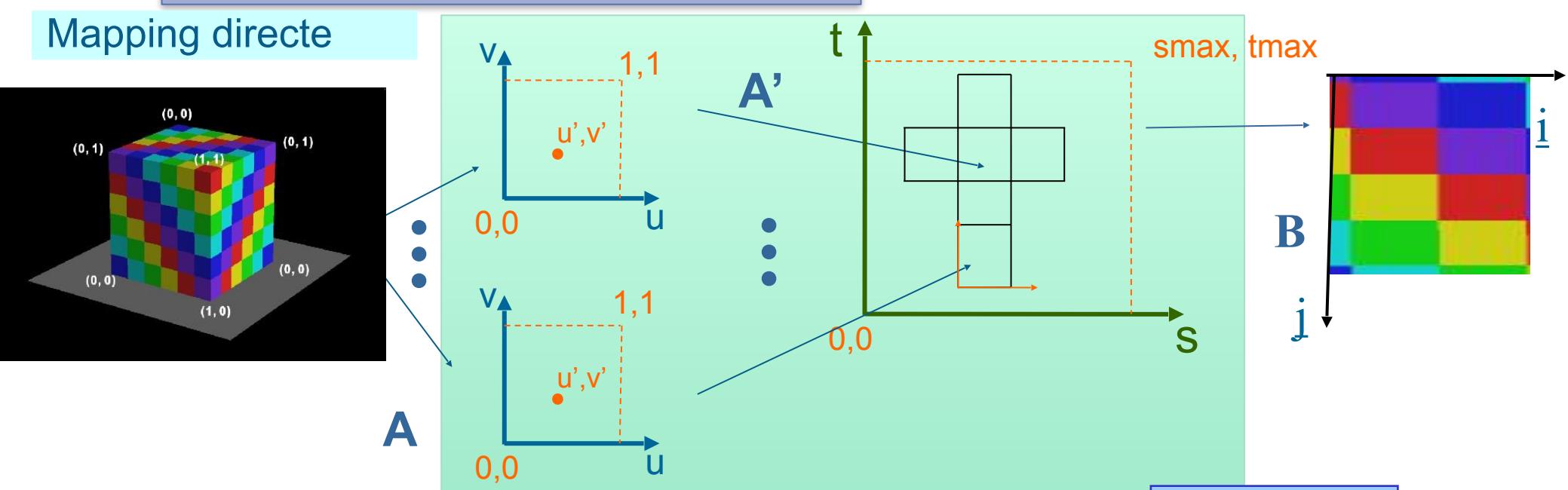


1. Retallat de la superfície en trossos
 2. Desplegament de la superfície en un pla
 3. Mapeig en una imatge
 4. Obtenció dels colors
- Utilitzat en els modeladors com Blender, Maya, etc.
 - Estan codificats en els .obj que es donen

3.4. Textures: funció de projecció



Mapping directe



A : Funció de (x, y, z) a (u, v) .

A' : Funció de (u, v) a l'espai de textures (s, t)

Funció de projecció a (s,t)

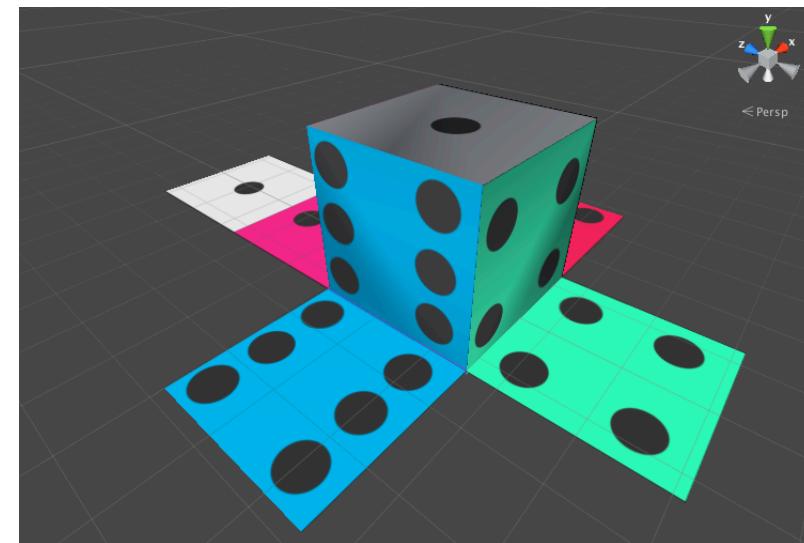
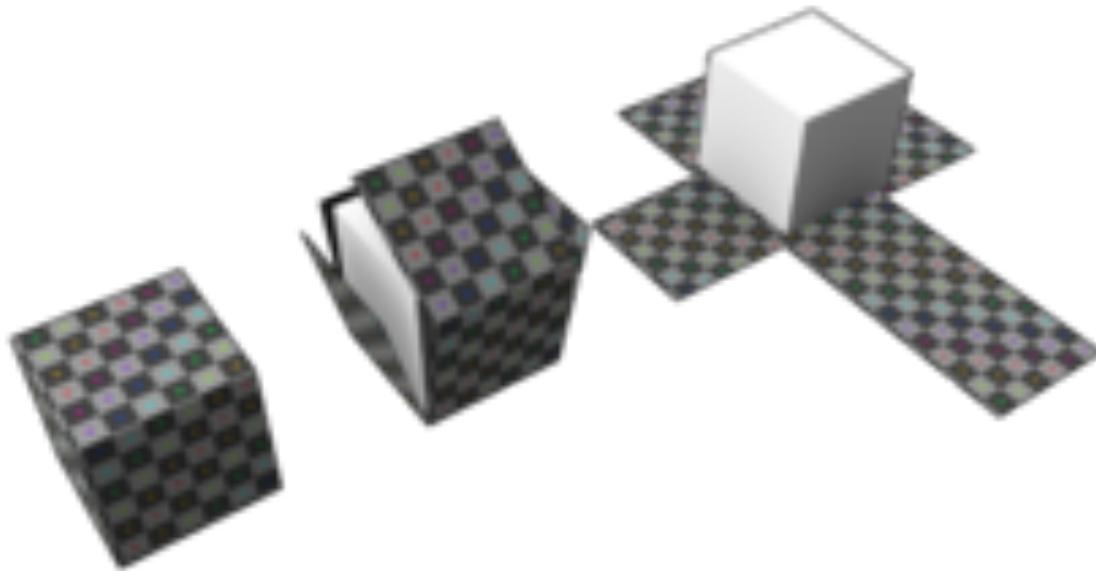
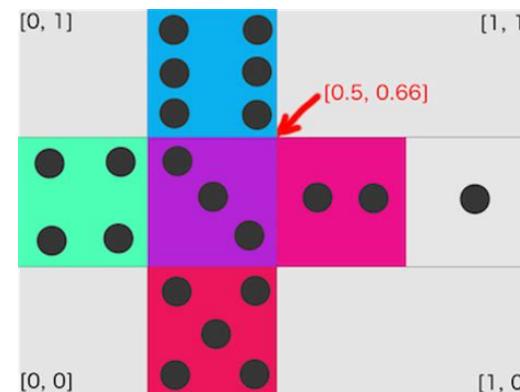
B : Funció de (s, t) al texel (i, j)

Funció de correspondència (texel (i,j))

3.4. Textures: funció de projecció

- **Malles poligonals:**

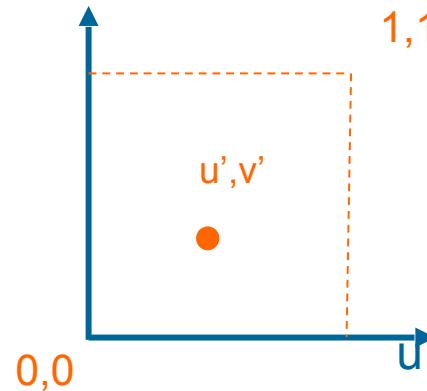
Definir (u, v) a cada vèrtex



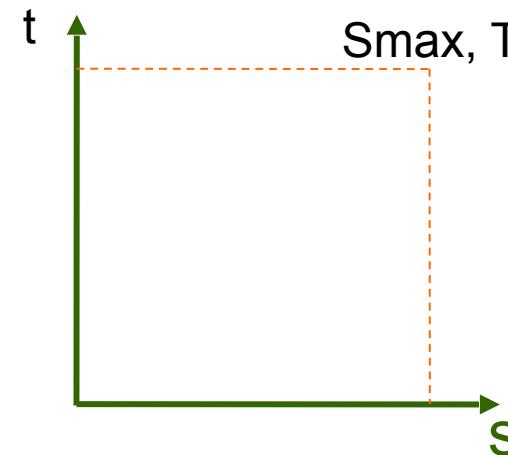
3.4. Textures: funció de projecció

- Quins efectes es poden fer amb la funció de projecció

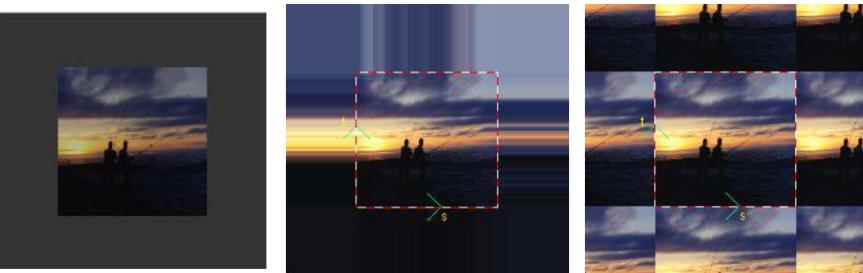
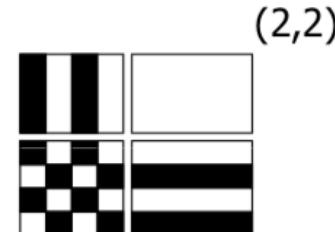
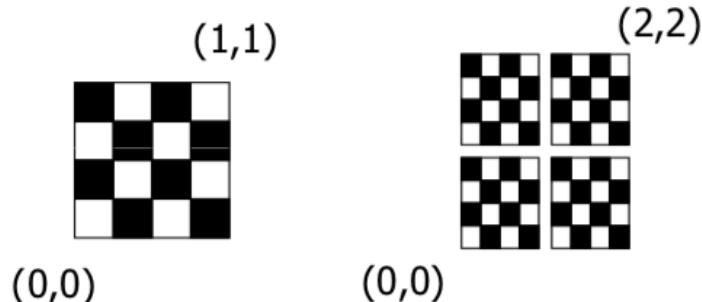
A'?



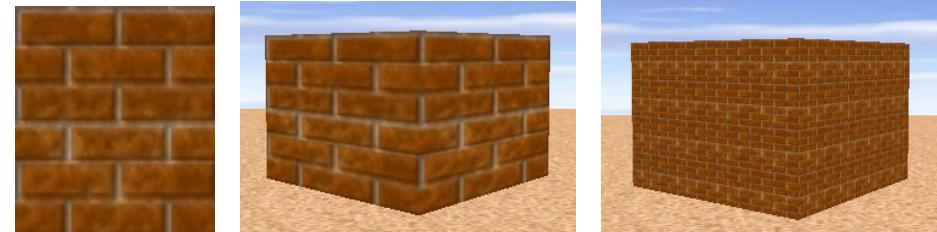
A'



WRAP:
mirall
stretching
repetició o tiling
repetició mirall
clamp
marc

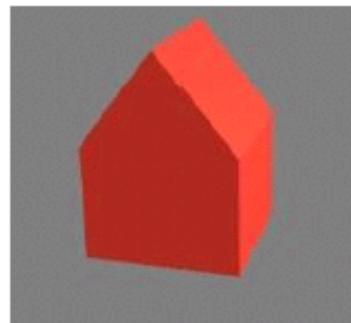


Textura

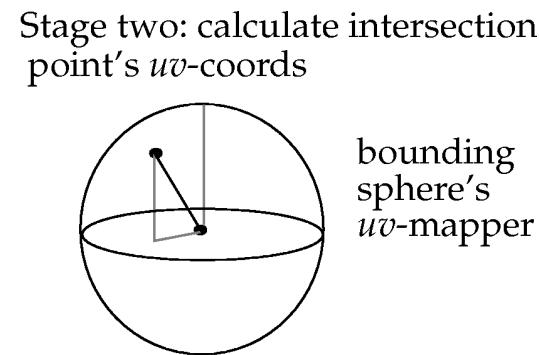
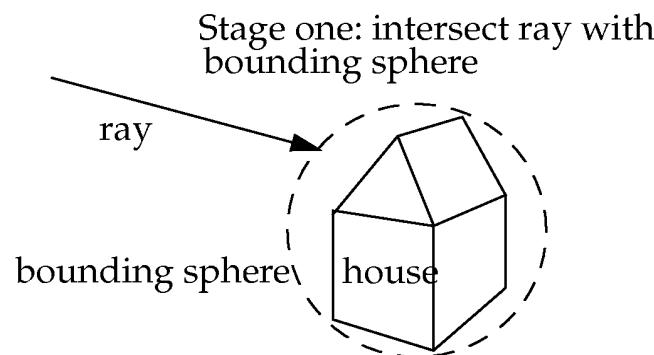


3.4. Textures: funció de projecció

Mapping directe: es el mapping que s'aplica a cada polígon de l'objecte o a cada superfície. Pot donar discontinuïtat a les arestes



Mapping en dues fases: s'utilitza una figura intermitja

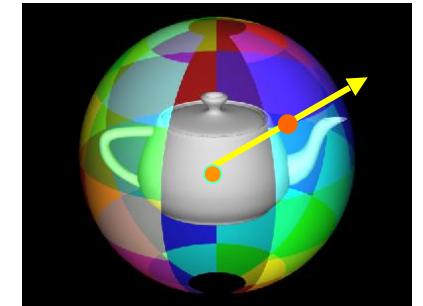


3.4. Textures: funció de projecció

Mapping en dues fases

o-mapping: de (x,y,z) a (x',y',z')

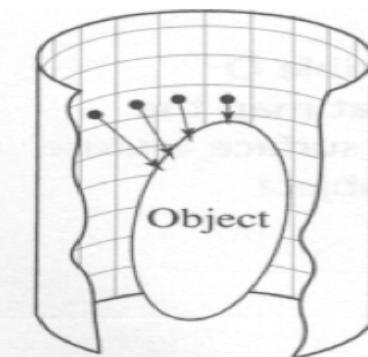
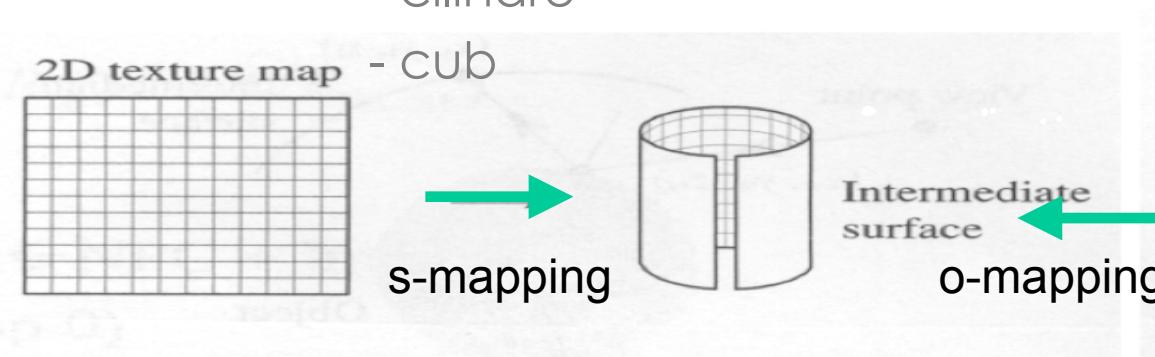
- raig reflectit de visió
- raig normal a x,y,z
- raig x,y,z al centroide de l'objecte
- raig normal a la superfície



s-mapping: de (s,t) a (x',y',z')

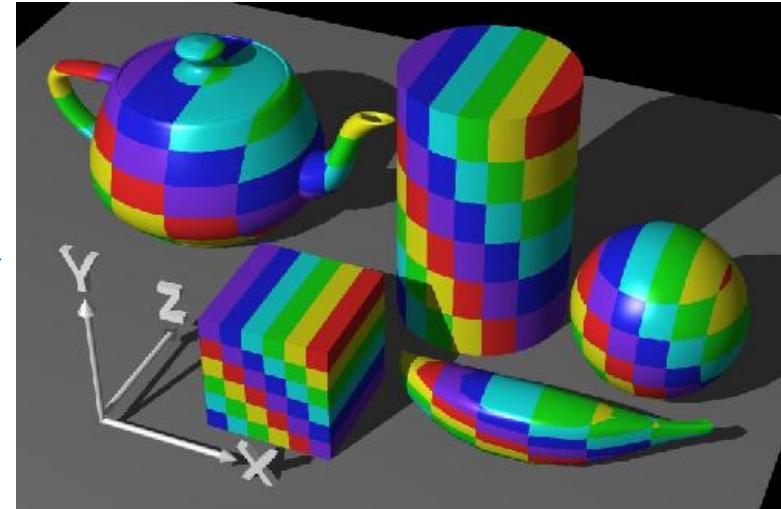
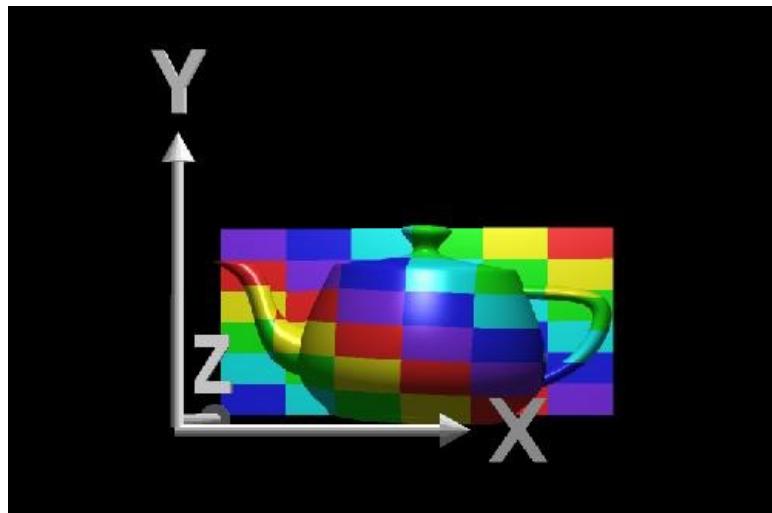
de l'espai de textura a un objecte intermig:

- pla
- esfera
- cilindre
- cub

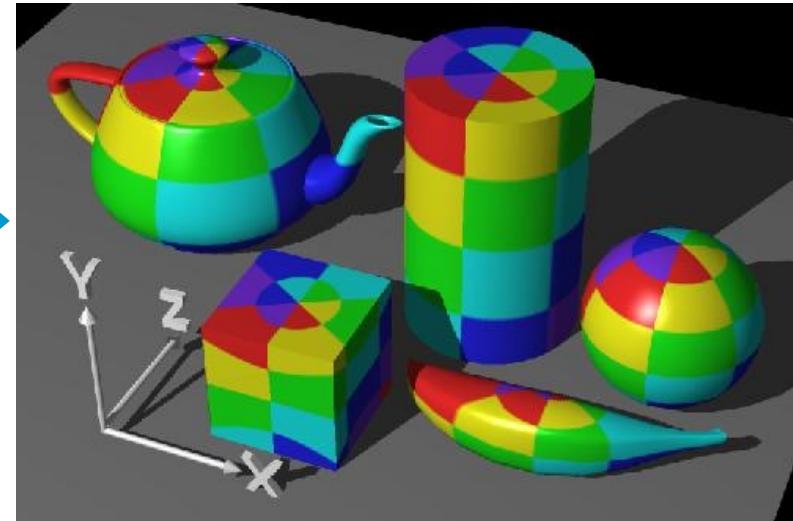
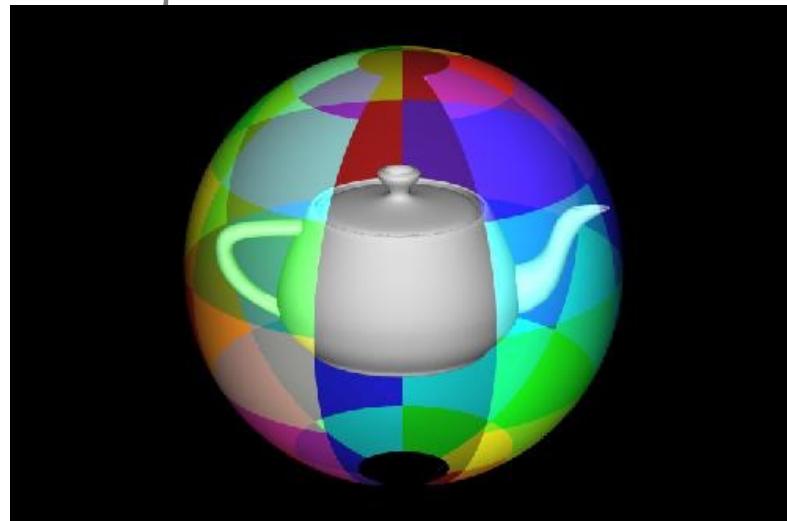


3.4. Textures: funció de projecció

Espai objecte: Pla

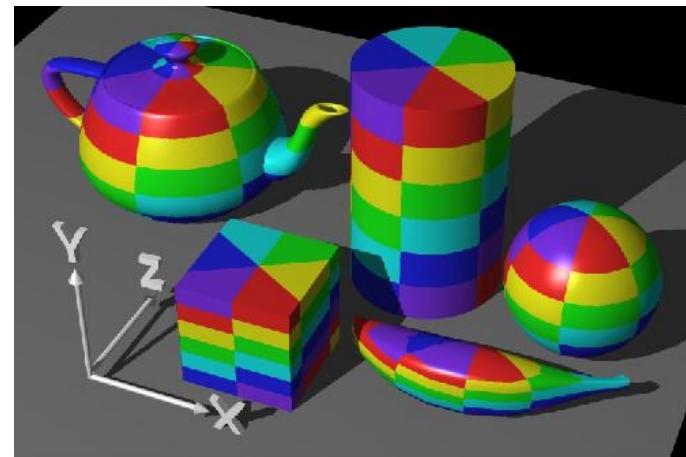


Espai objecte: Esfera

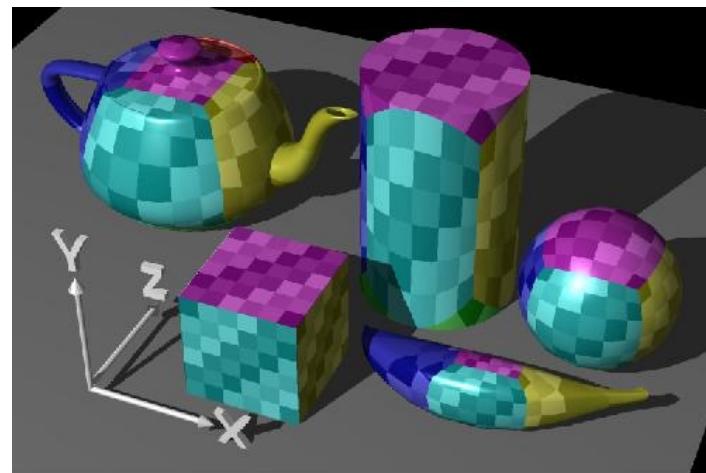
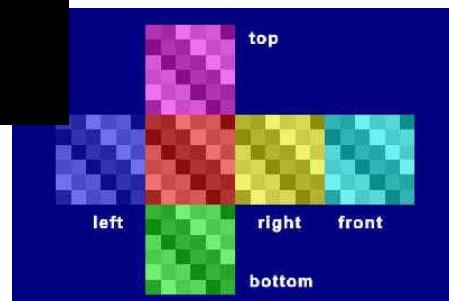
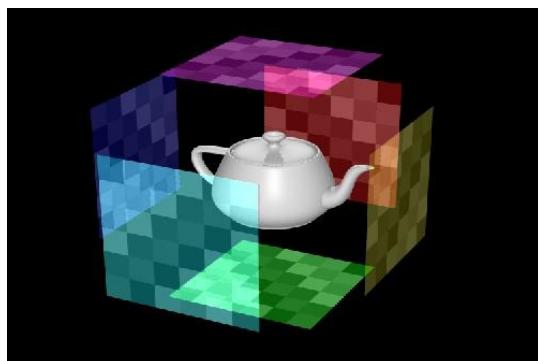


3.4. Textures: funció de projecció

- Espai objecte: Cilindre



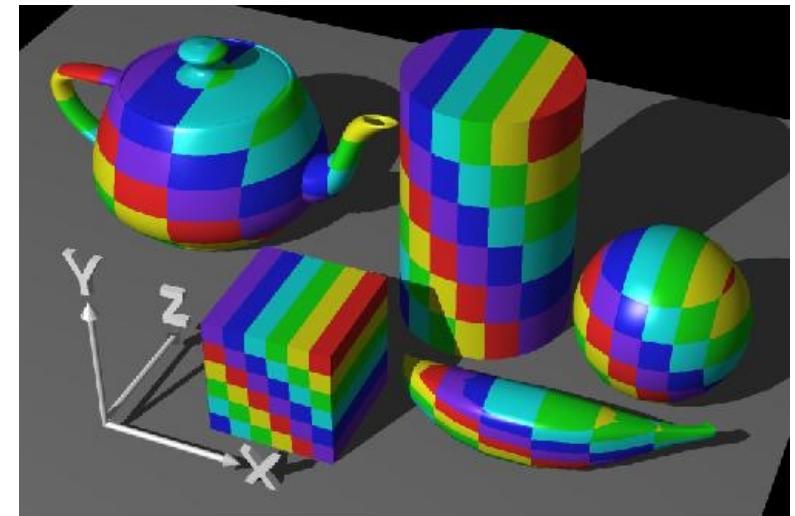
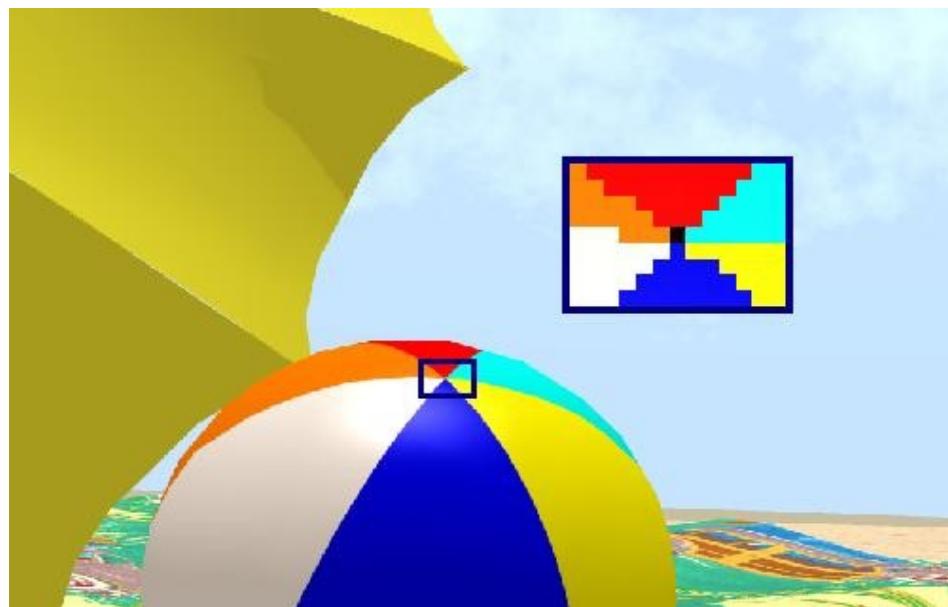
- Espai objecte: Cub



3.4. Textures: mapping en dues fases

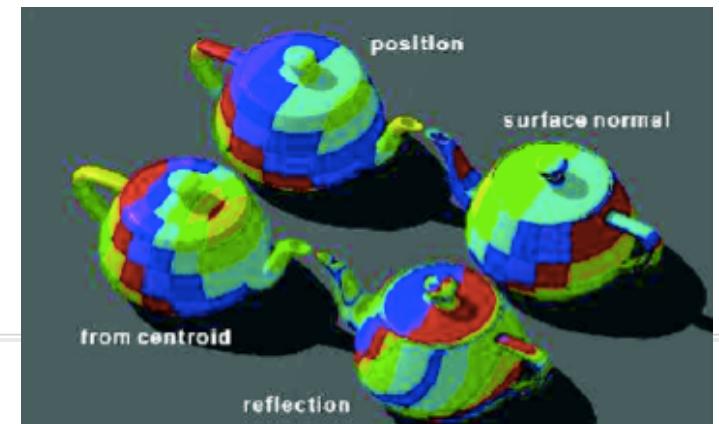
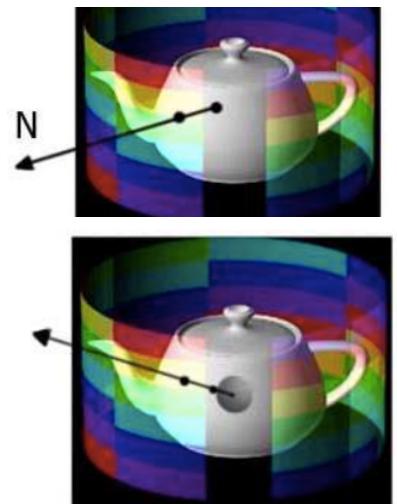
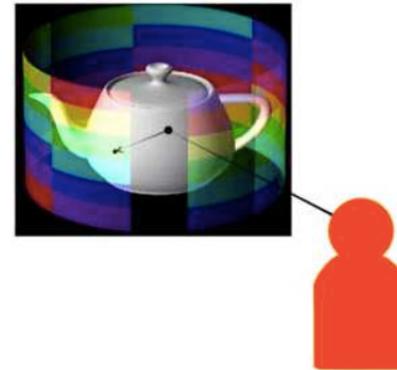
Cada tipus de funció de projecció en dos passos té els seus inconvenients:

- l'esfèrica: warping en els pols de l'esfera
- el cilíndric: discontinuitats en les arestes de les tapes
- la planar: s'ignora una de les components.



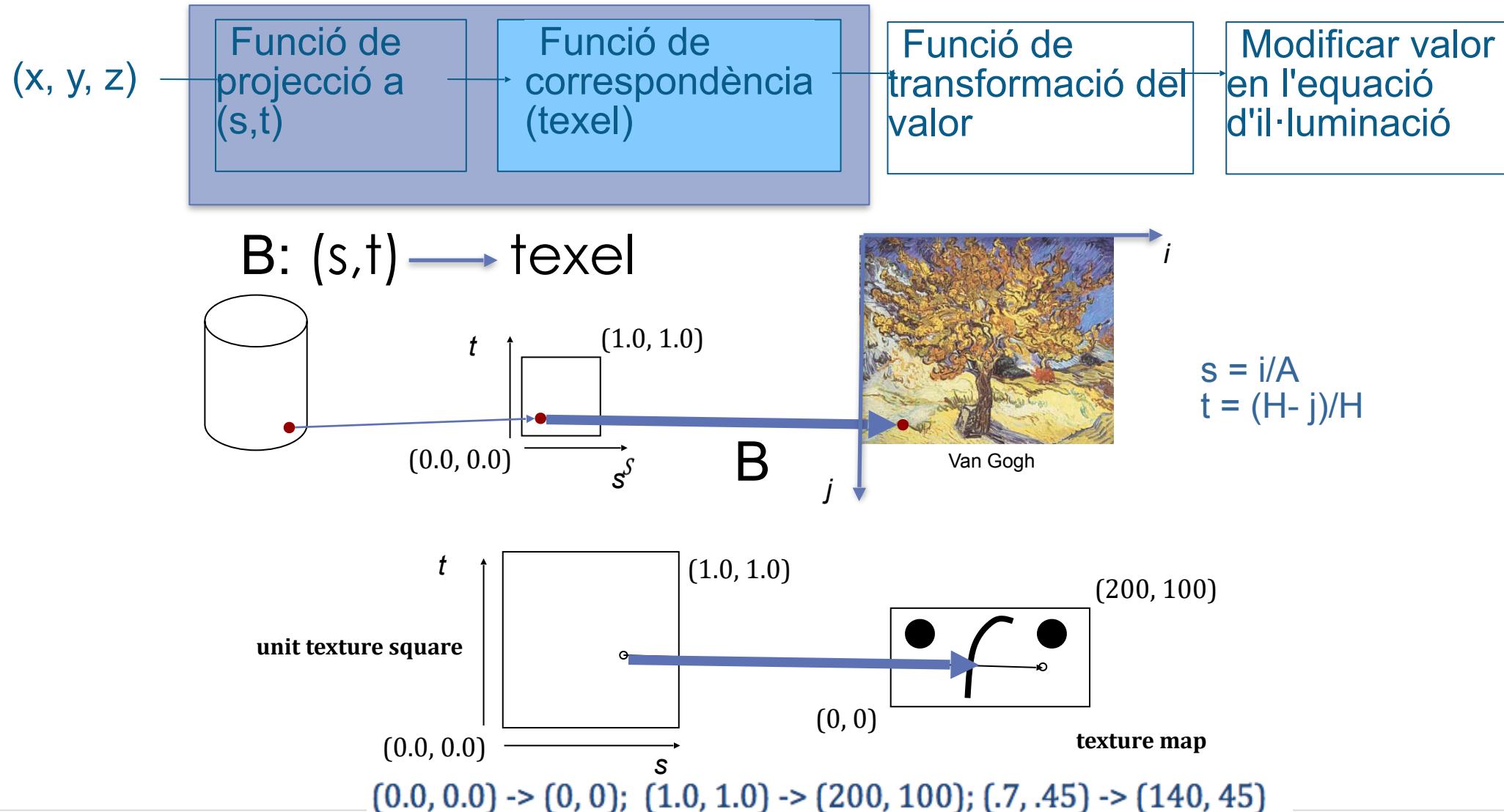
3.4. Textures: mapping en dues fases

- o-mapping:
 - Raig reflectit de visió en (x, y, z)
 - Raig normal a (x, y, z)
 - Raig (x, y, z) al centroide de l'objecte
 - Raig normal a la superfície intermitja que intersecta a (x, y, z)



Simplificació: s'agafa directament el punt de l'esfera que passa per (x, y, z) que té com a centre, el centroid de l'objecte

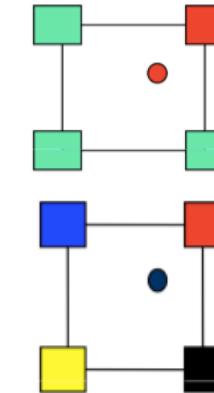
3.4. Textures: funció de correspondència



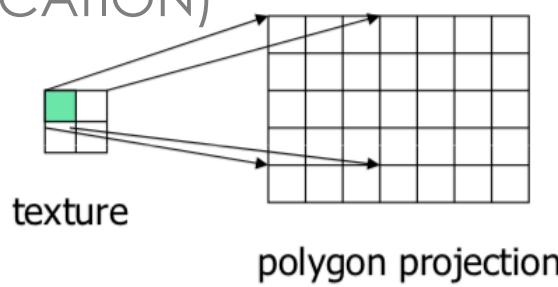
3.4. Textures: funció de correspondència

Es basa en la correspondència entre l'espai continu (s, t) i discret (i, j)

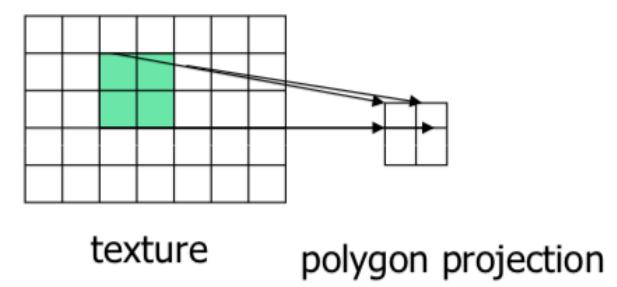
- GL_NEAREST: Selecciona el píxel més proper
- GL_LINEAR: Promig dels 4 píxels veïns al punt
- Aquest fet provoca problemes **d'aliasing**:



- Varis píxels de pantalla corresponen a un mateix téxel (téxels/píxels < 1) (MAGNIFICATION)
- Varis téxels cauen en un píxel de pantalla (téxels/píxels > 1) (MINIFICATION)



Magnification



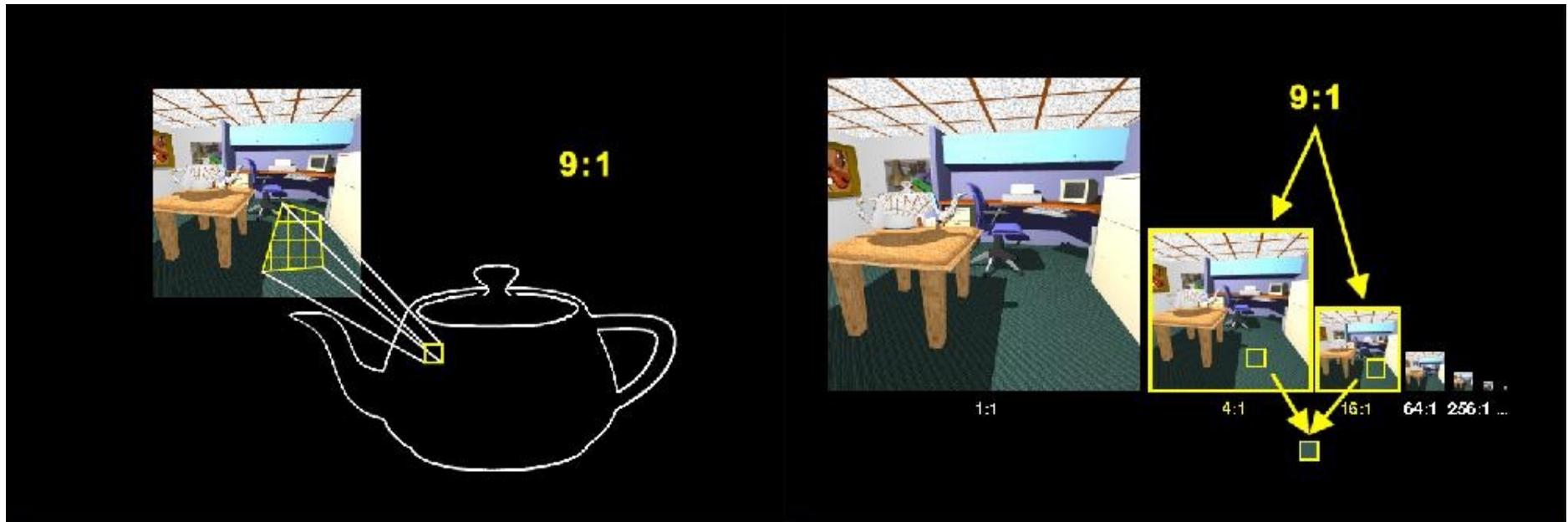
Minification

3.4. Textures: anti-aliasing

MipMapping: Mètode alternatiu d'antialiasing: preprocés

MIP = multium in parco; (molts en un petit lloc)

Idea bàsica: construir una piràmide d'imatges per diferents mides de la mateixa textura i segons la distància a l'observador s'escull una o una altra



3.4. Textures: anti-aliasing

MipMapping i Multi-textures: Per un mateix objecte i coordenades de textures es generen diferents “skins”

