Tema 3 - Elementos (assets) y captura de datos.

3.2 Técnicas de desenrollado y texturas 2D.

Germán Arroyo, Juan Carlos Torres

5 de febrero de 2021

Contenido del tema

Tema 3: Elementos (assets) y captura de datos.

3.1 Nubes de puntos y capturas mediante escáneres 3D.

3.2 Técnicas de desenrollado y texturas 2D.

3.3 Simplificación de modelos 3D y texturización automát

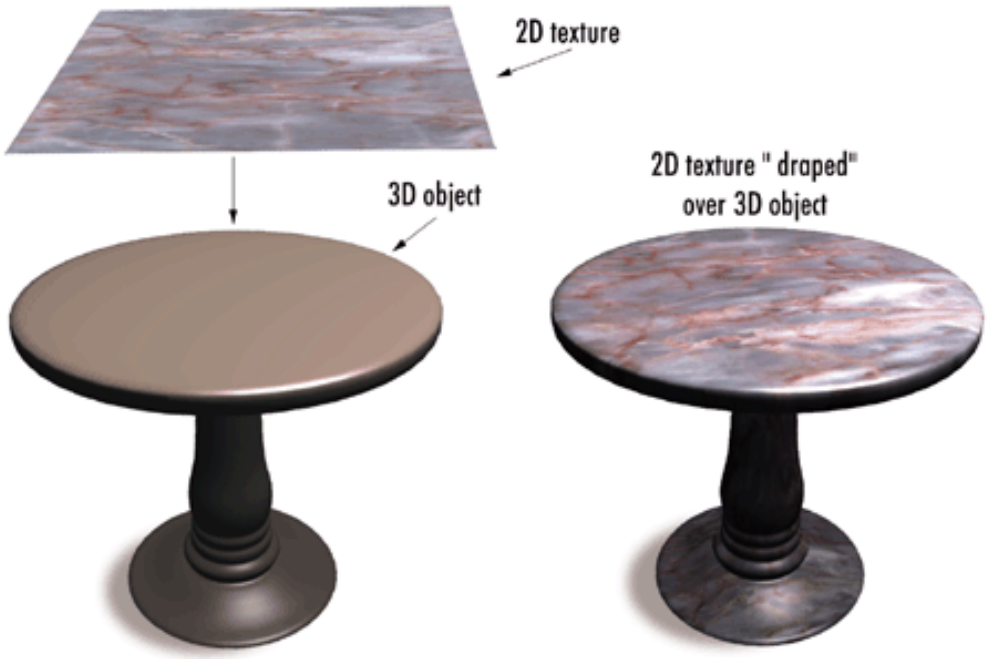
3.4 Materiales y shaders de iluminación.

3.5 Nuevas técnicas software y hardware para la de genera

3.2 Técnicas de desenrollado y texturas 2D.

Una **textura** es una imagen que se proyecta sobre una geometría.

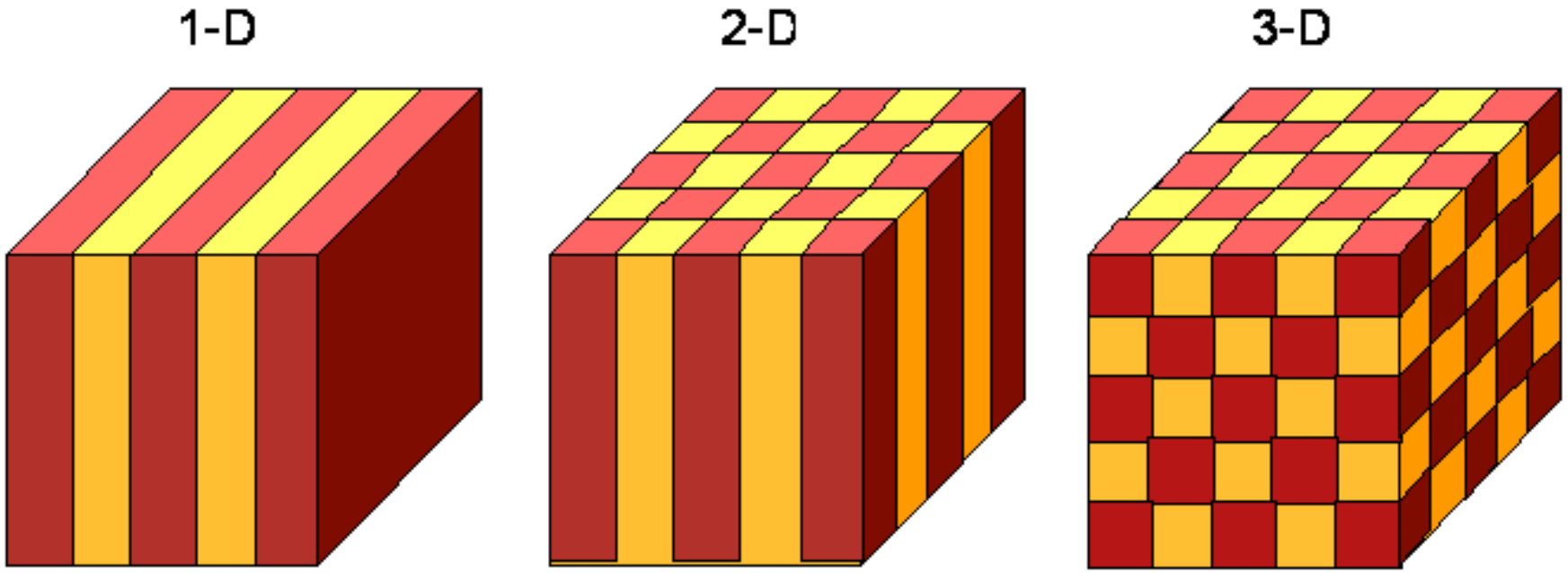
Transferida de disco (SSD)/memoria a GPU.



Tipos de texturas

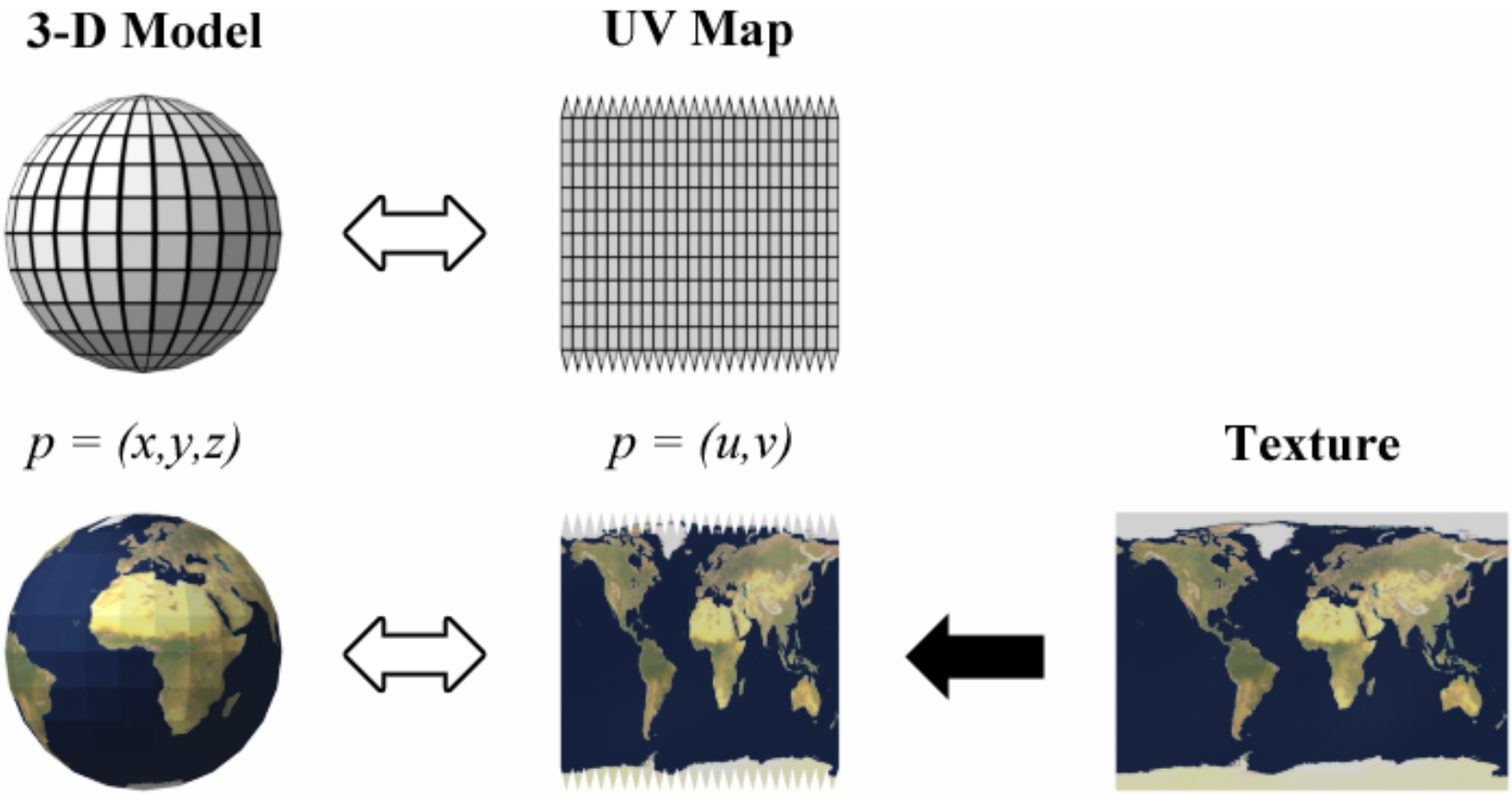
Una textura puede tener múltiples dimensiones: 1D, 2D o 3D.

La unidad básica de una textura es el **texel**.



Texture mappping

La correspondencia se establece asociando al vértice las coordenadas del punto de la textura (coordenadas de textura). Las **coordenadas de textura** se dan **normalizadas**.



Parametrización (I)

Las coordenadas de textura se puede obtener desplegando las caras del objeto.

Problemas de realizar la parametrización:

- Islas (zonas no conexas).
- Aristas abiertas formando costuras (*seams*) en el modelo.

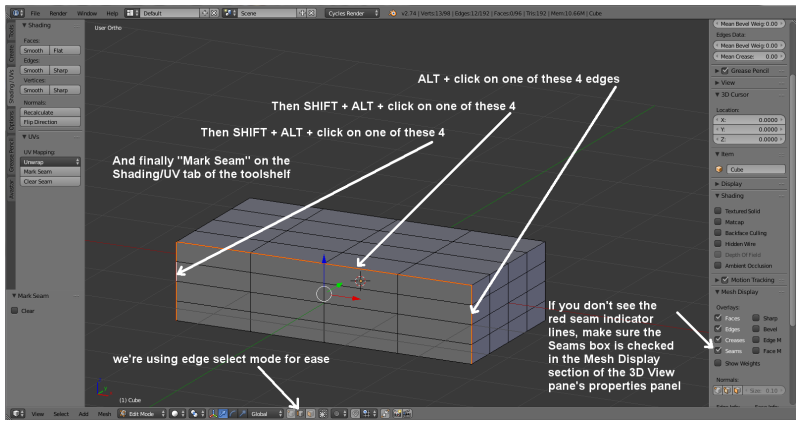


Figura 1: Proceso de marcado de costuras en Blender.

Parametrización (II)

Más problemas de realizar la parametrización:

- Zonas no utilizadas en la textura.

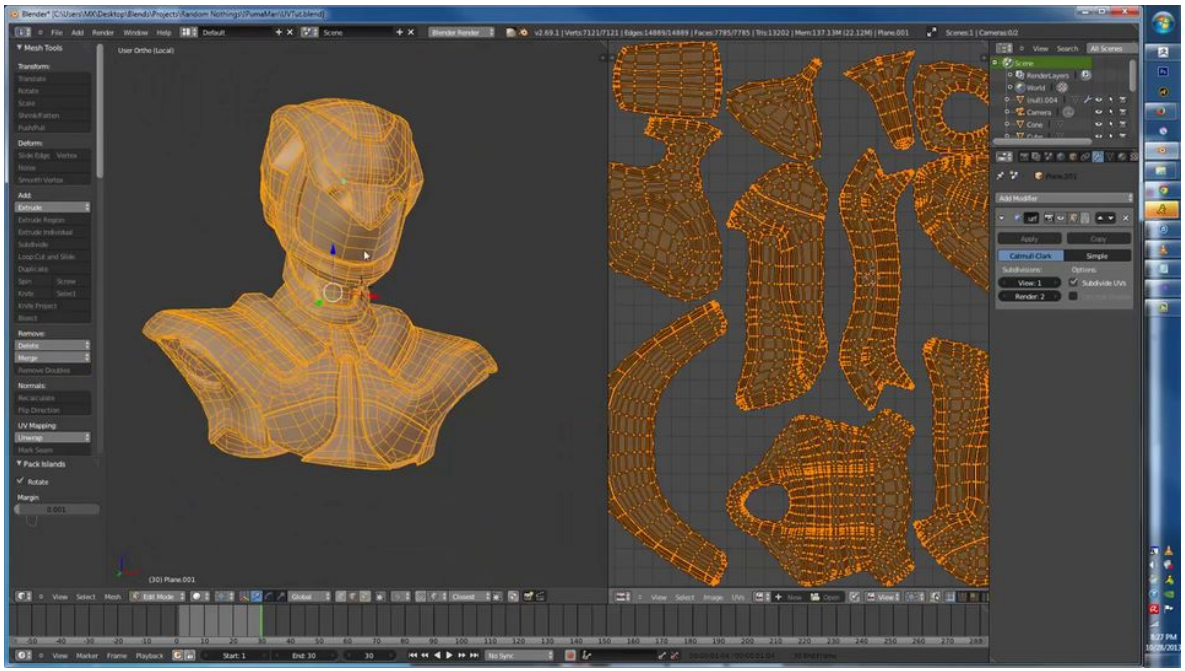


Figura 2: Ejemplo de textura no aprovechada al 100%.

Parametrización (III)

Tipos de parametrización:

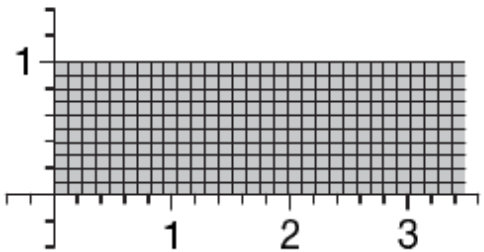
- Completamente manual (triángulo a triángulo o por áreas).
- Completamente automática (aparecen problemas).
- Semi-automática (zonas desplegadas mediante algoritmos, marco de costuras, etc.)

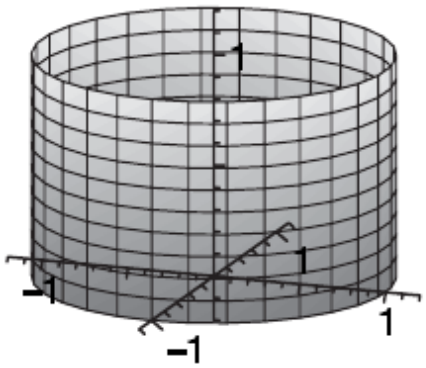
Parametrización automática (I)

Parametrización automática al crear un objeto:

- Envolver el objeto con una superficie paramétrica.
- Asignar coordenadas de textura en a partir de puntos correspondientes en la superficie.
- La parametrización es la inversa de la función paramétrica que define la superficie.

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (\cos u, \sin u, v) \\ \textit{inverse:} \quad & f^{-1}(x,y,z) = (\arccos x, z) \end{aligned}$$





Parametrización automática (II)

Para cada vértice tendremos unas coordenadas UV asignadas.

Se pueden tener varios canales UV.

Las islas se consiguen duplicando los vértices.

```
var uvs = PoolVector2Array()
...
arr[Mesh.ARRAY_TEX_UV] = uvs
```

Parametrización automática (III)

La proyección genera distorsiones y no es unívoca.

Se puede obtener desplegando cintas de triángulos.

- Genera muchas costuras.
- Genera muchos espacios perdidos.
- Genera muchas islas.

Kai Hormann, Bruno Lévy, Alla Sheffer. **Mesh Parametrization: Theory and Practice.** Siggraph Course Notes, 2007.

Olga Sorkine and Daniel Cohen-Or. **Warped textures for UV mapping encoding.** SHORT PAPER EUROGRAPHICS 2001

Aplicaciones de la parametrización

Se utiliza entre otras operaciones para:

- *Morphing*.
- Reparar mallas.
- Materiales.
- Creación de terrenos.
- etc.

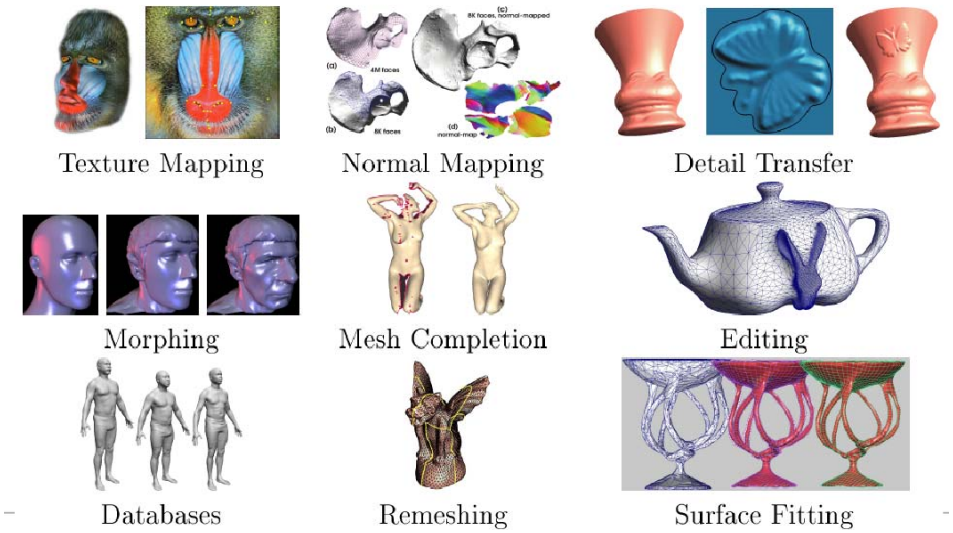


Figura 3: Usos comunes de las texturas.

Bump mapping (I)

Las texturas no solamente se utilizan para el color:

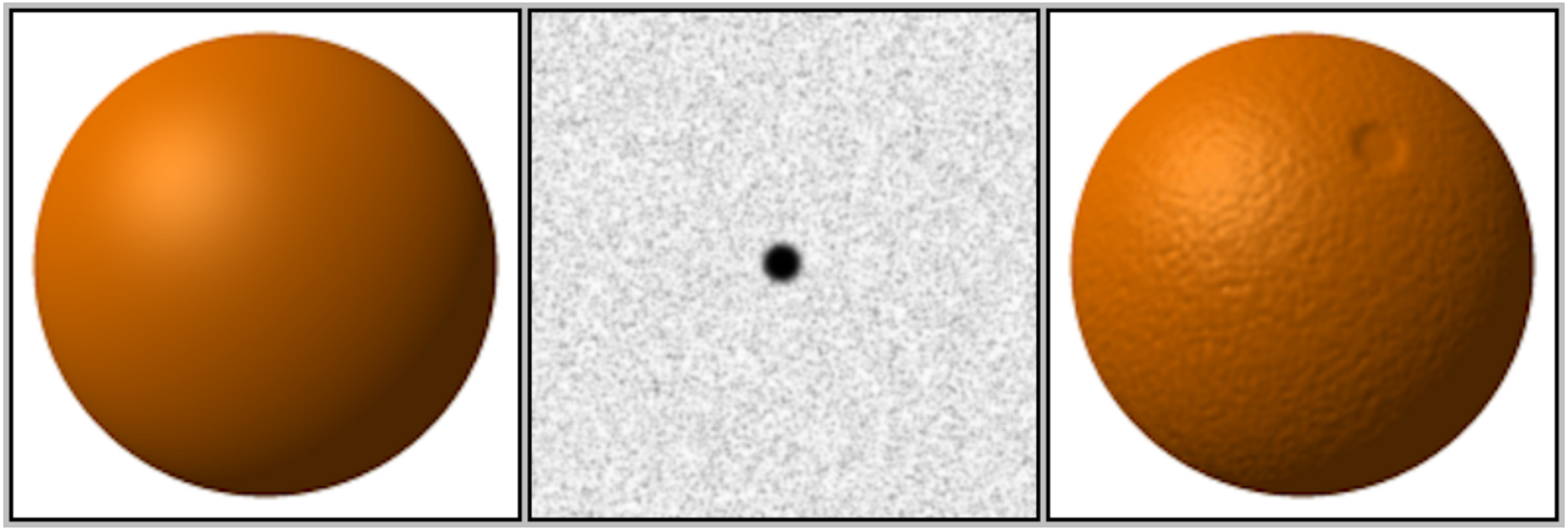


Figura 4: Ejemplo de bump mapping.

https://upload.wikimedia.org/wikipedia/commons/9/93/FakeBump2D-animation.gif

Bump mapping (II)

1. Comprobar la altura del mapa que corresponde a la superficie.
2. Calcular la normal a la superficie en el mapa de altura, típicamente mediante derivadas.
3. Combinar la normal de la superficie con la normal real (*geométrica*), combinándolas en una nueva dirección.
4. Calcular la interacción de la luz con la superficie con algún modelo de iluminación (ej. Lambert, Phong, etc.).

Derivada de una imagen (I)

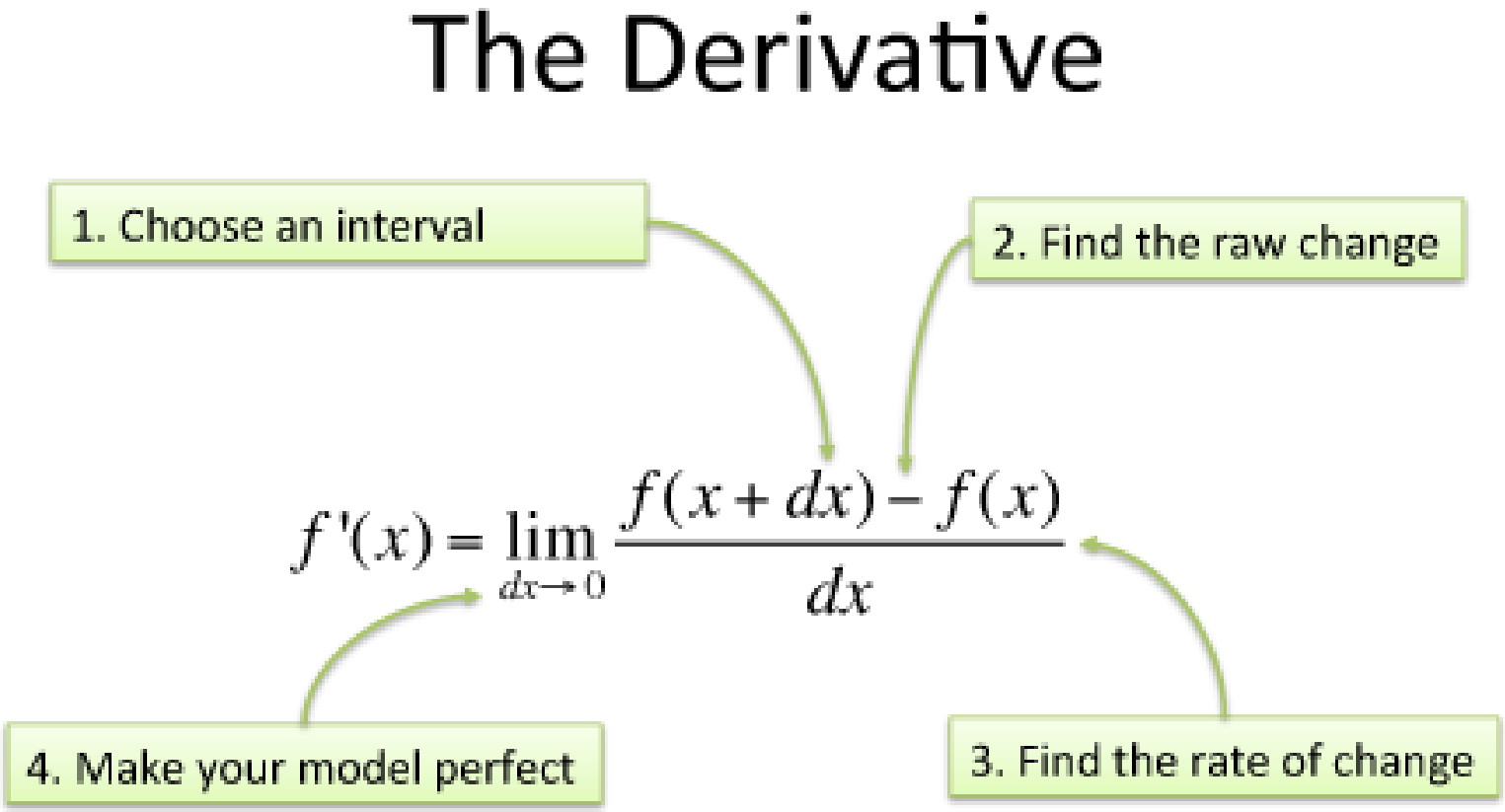


Figura 5: Definición de derivada.

Derivada de una imagen (II)

Convolución y *kernel*:

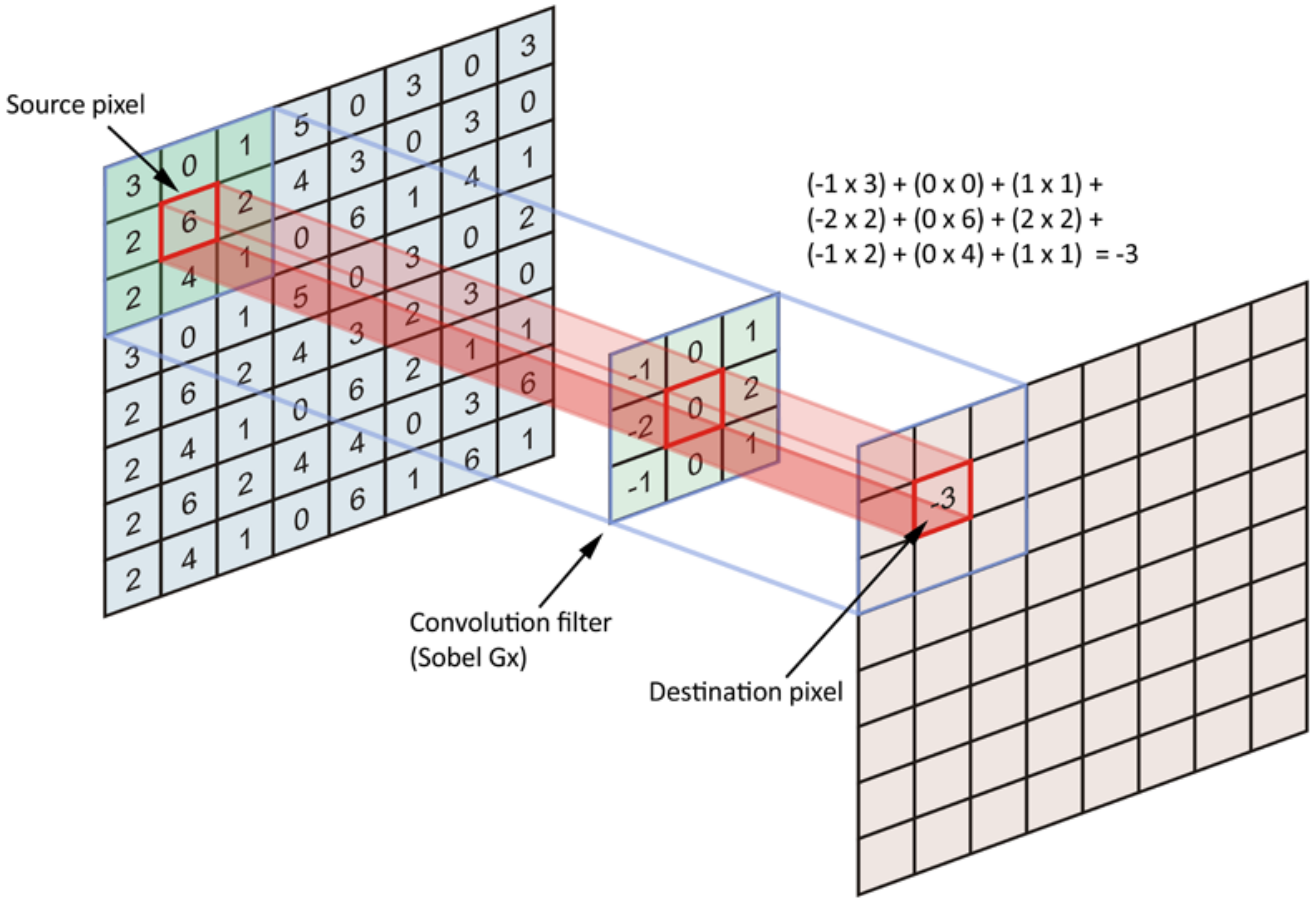


Figura 6: Operación de convolución.

Derivada de una imagen (III)

Kernel DoG:

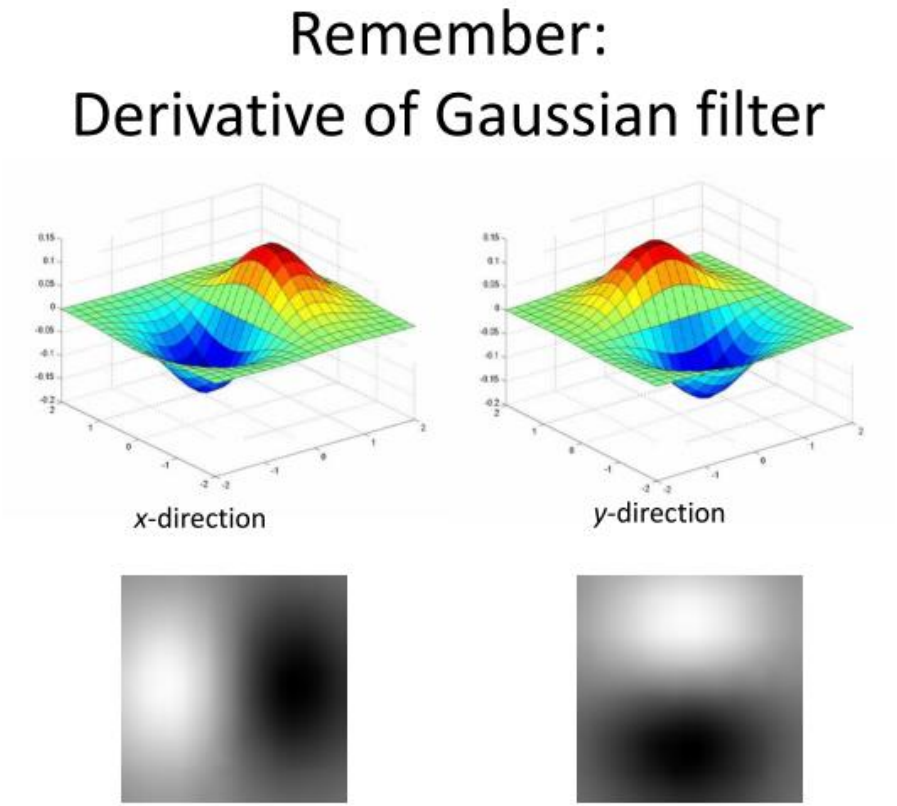


Figura 7: 1ª derivada de la Gaussiana.

Derivada de una imagen (IV)

Magnitud y dirección:

Figura 8: Direcciones (izq. y centro) y magnitud (derecha) de los vectores.

Normal mapping

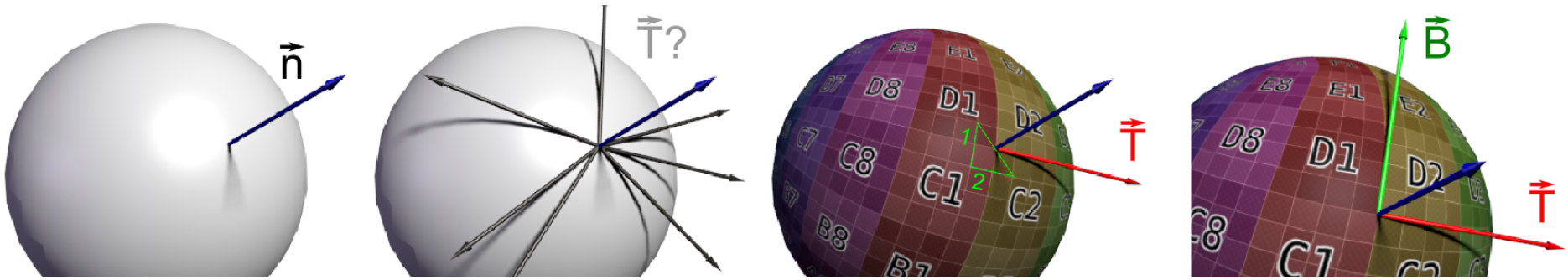
Parecido al Bump Mapping, pero las normales del mapa ya vienen dadas en RGB: $\vec{n} = (n_x = \mathsf{R}, n_y = \mathsf{G}, n_z = \mathsf{B})$.

Figura 9: Ejemplo de imagen de normales.

Tangente y bitangente (I)

Calcular la tangente ($\vec{T}$) y la bitangente ($\vec{B}$) a partir de los lados del triángulo ($\Delta_1 pos$, $\Delta_2 pos$) usando sus coordenadas de textura ($\Delta_1 UV$, $\Delta_2 UV$):

- $\Delta_1 pos_x = \Delta_2 UV_x \cdot \vec{T} + \Delta_1 UV_y \cdot \vec{B}$
- $\Delta_2 pos_x = \Delta_2 UV_x \cdot \vec{T} + \Delta_2 UV_y \cdot \vec{B}$



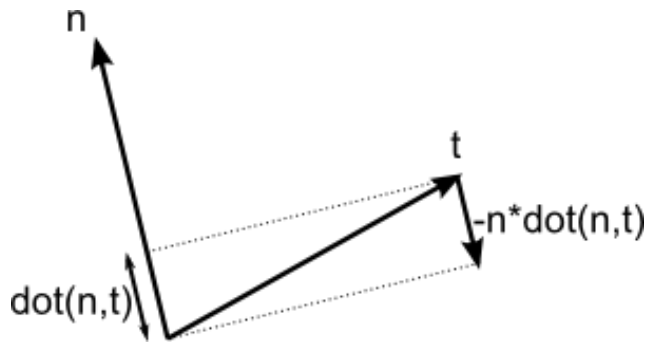
Tangente y bitangente (II)

Teniendo $\vec{T}$, $\vec{N}$ y $\vec{B}$ pasamos todo al espacio de la tangente, usamos la inversa de la matriz TBN (descompuesta en filas), o (por eficiencia) su traspuesta:

$$\text{TBN}^T = \begin{pmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{pmatrix}$$

Solamente si primero la hacemos ortogonal. Para ello, hacemos la tangente perpendicular a la normal:

$$\vec{T} = \text{norm}([\vec{T} - \vec{N}] \cdot [\vec{N} \cdot \vec{T}])$$



Cocinado (bake) de texturas

Necesario que ambos modelos compartan el mismo espacio UV.

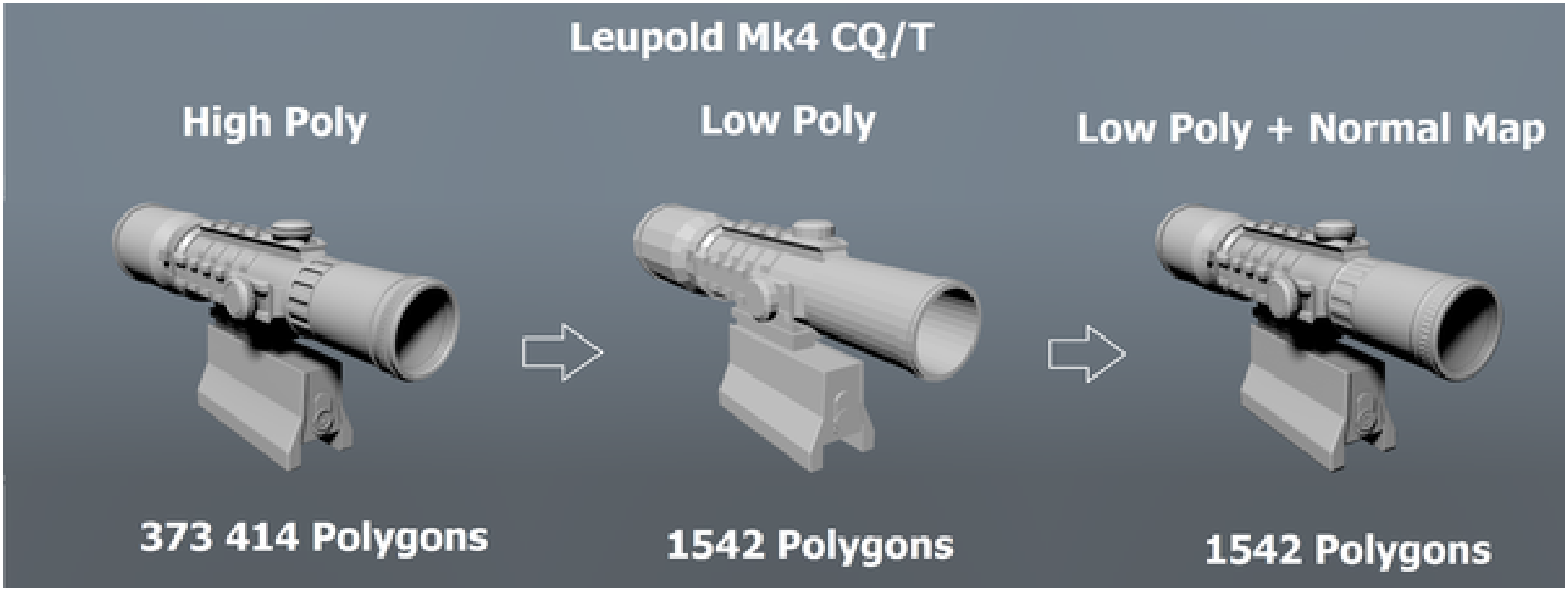


Figura 10: Modelo a alta resolución (izq.), modelo a baja (dcha.) y mismo modelo con normales precocinadas.