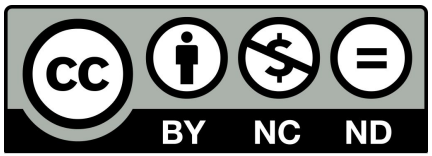
Computação Gráfica I (MAB122)

João Vitor de Oliveira Silva [1]

Textura

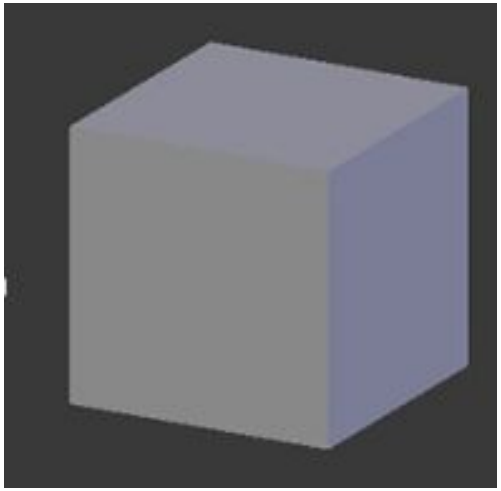
6 de Novembro



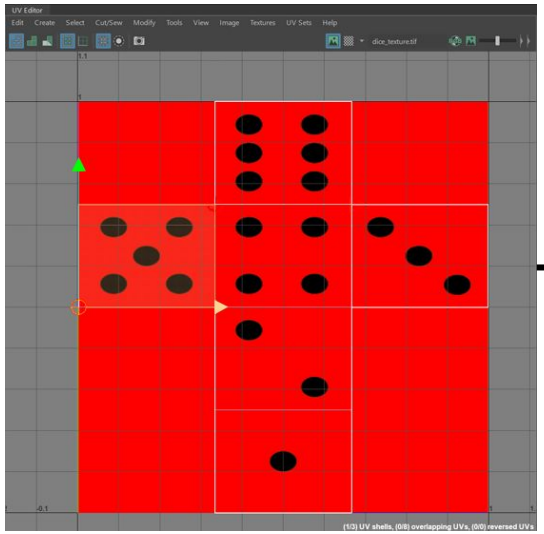
[1] jvos@dcc.ufrj.br, jvos@lncc.br

Introdução

https://commons.wikimedia.org/wiki/File:Texture_mapping_demonstration_animation.gif



https://i.stack.imgur.com/wSWRe.png

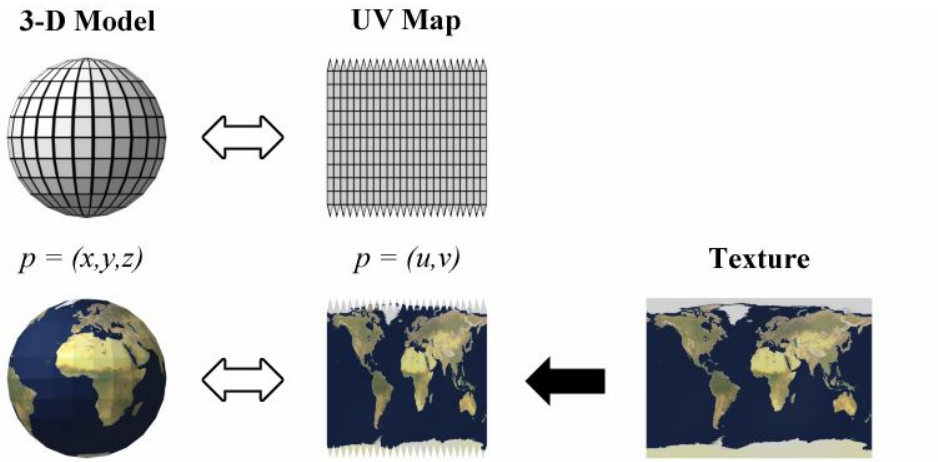


http://users.design.ucla.edu/~cariesta/3DCourseNotes/assets/images/auto_generated_images/dice_done.png

Textura

- Uma imagem em raster com canais $[r, g, b, a]$
- Mapeamento de textura: cobrir objeto 3D com a imagem de textura, como se fosse um adesivo
- Um elemento de uma textura chama-se de *texel*

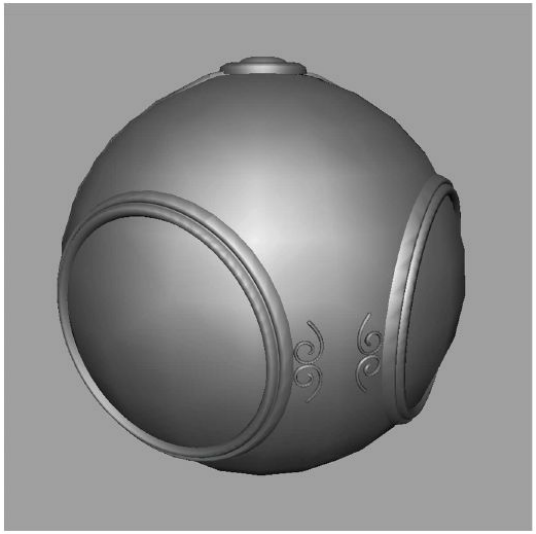
https://pixar-community-production.s3.us-west-1.amazonaws.com/resources/texture_library_128/220_contactsheet_002.jpg

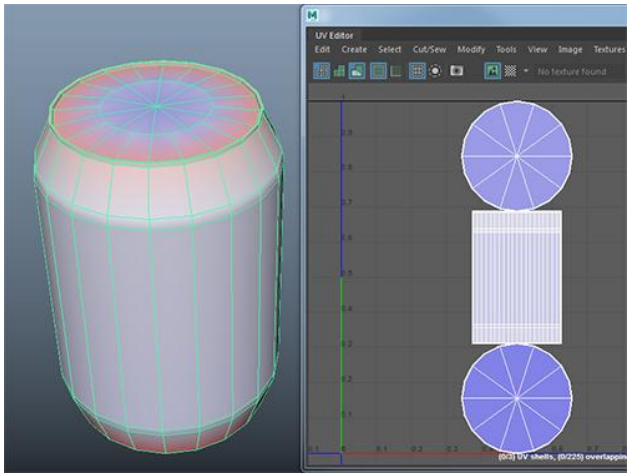


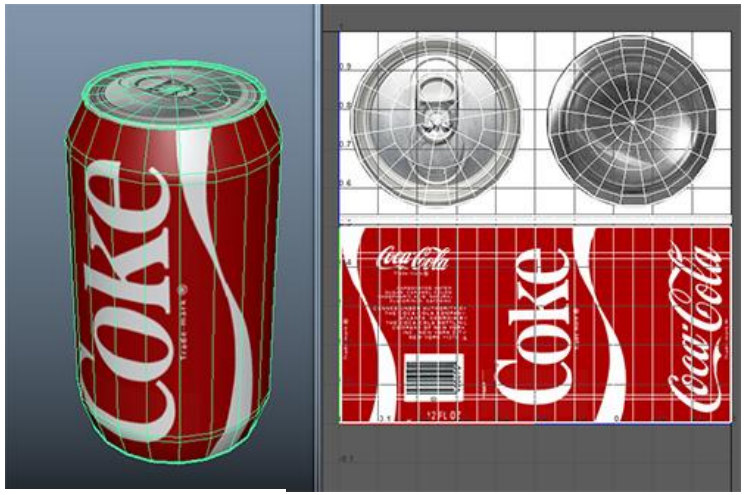
https://upload.wikimedia.org/wikipedia/commons/0/04/UVMapping.png

Textura: motivação

Possibilidade de descrever objetos geométricos realistas, sem a necessidade de modelar um objeto complexo todo vetorialmente



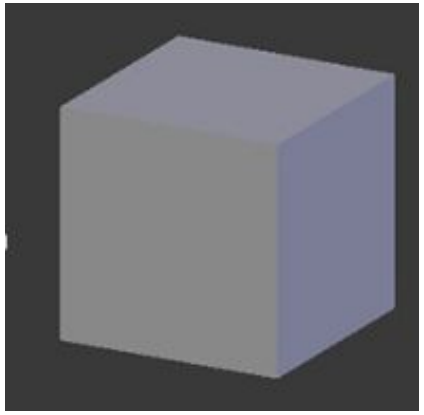


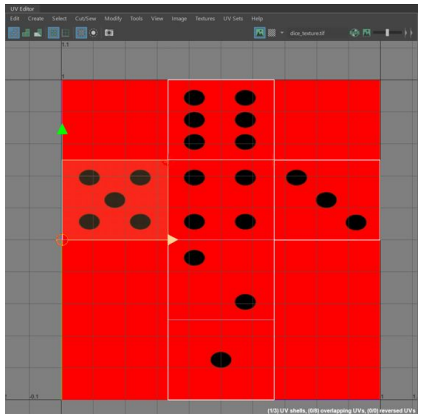


https://documents.sessions.edu/eforms/courseware/coursedocuments/maya_i/lesson5.html

Textura: aplicações

- ⋆ **Texture Mapping**: aplica um certo padrão sobre o seu objeto a partir de uma imagem





- ⋆ *Bump Mapping*: (ou mapa de altura) simula diferenças na altura ao longo da superfície, destacando imperfeições

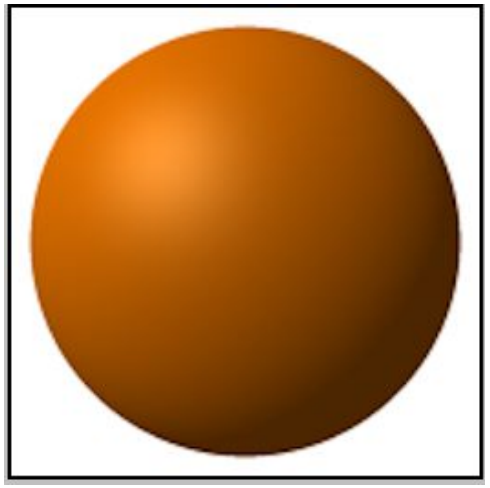
- ⋆ *Environmental mapping*: também chamado de *reflection mapping*. Usa textura para representar reflexões na cena

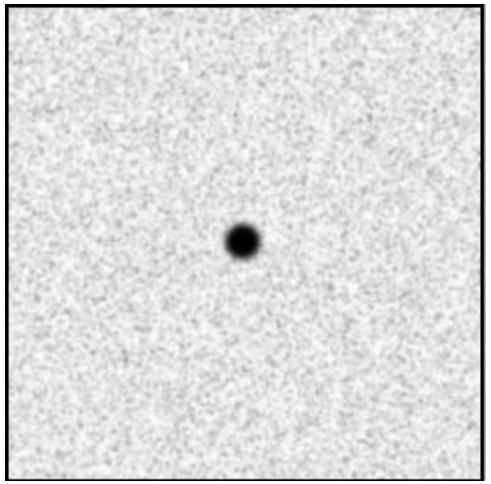
- ⋆ *Shadow mapping*: semelhante o anterior, mas para sombras

http://users.design.ucla.edu/~cariesta/3DCourseNotes/assets/images/auto_generated_images/dice_done.png

Textura: aplicações

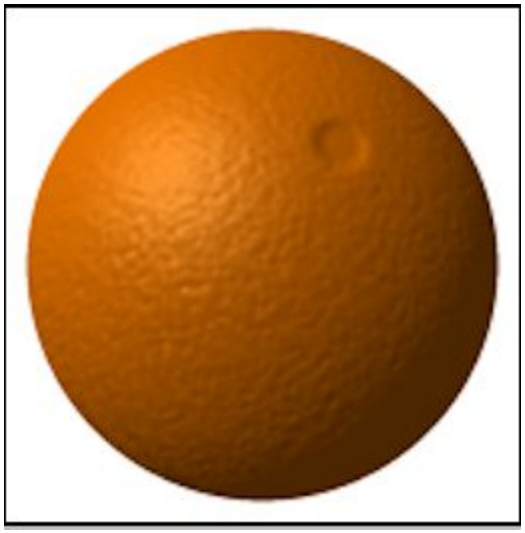
- ⋆ *Texture Mapping*: aplica um certo padrão sobre o seu objeto a partir de uma imagem





- ⋆ **Bump Mapping**: (ou mapa de altura) simula diferenças na altura ao longo da superfície, destacando imperfeições

- ⋆ *Environmental mapping*: também chamado de *reflection mapping*. Usa textura para representar reflexões na cena



- ⋆ *Shadow mapping*: semelhante o anterior, mas para sombras

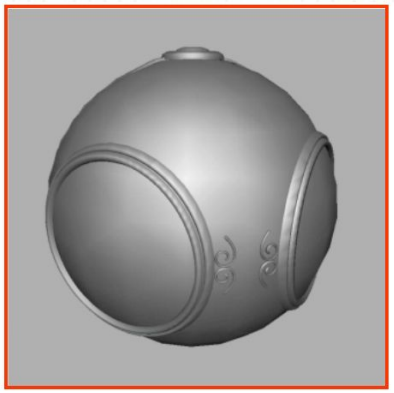
https://pt.m.wikipedia.org/wiki/Ficheiro:Bump-map-demo-full.png

Textura: aplicações

- ⋆ *Texture Mapping*: aplica um certo padrão sobre o seu objeto a partir de uma imagem

- ⋆ *Bump Mapping*: (ou mapa de altura) simula diferenças na altura ao longo da superfície, destacando imperfeições

- ⋆ **Environmental mapping**: também chamado de *reflection mapping*. Usa textura para representar reflexões na cena

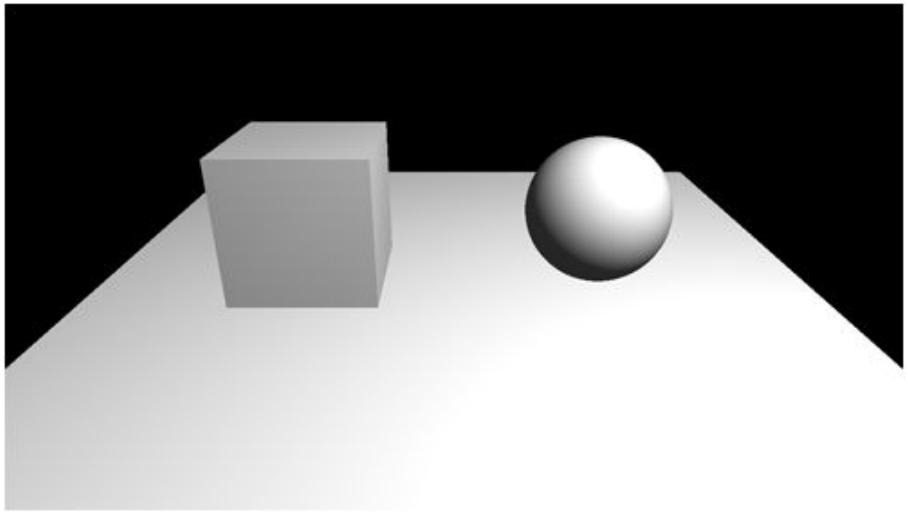


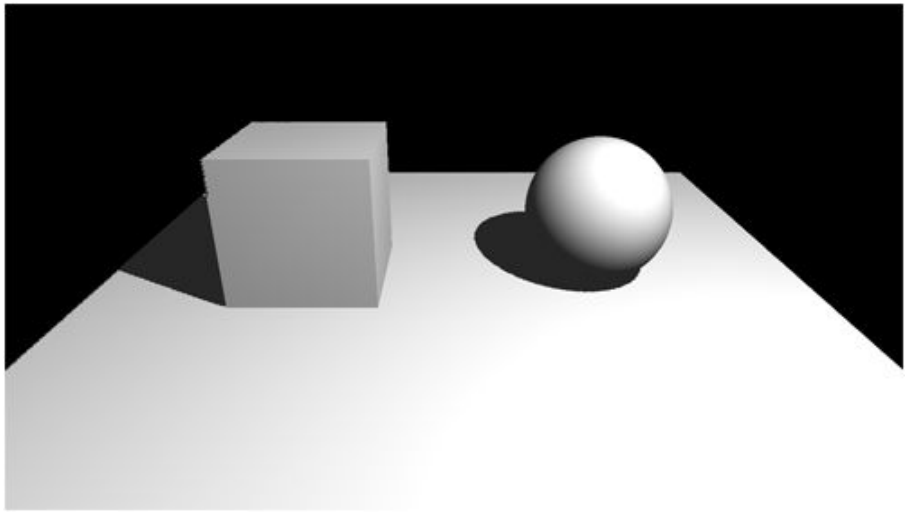
reflection map →

- ⋆ *Shadow mapping*: semelhante o anterior, mas para sombras

Textura: aplicações

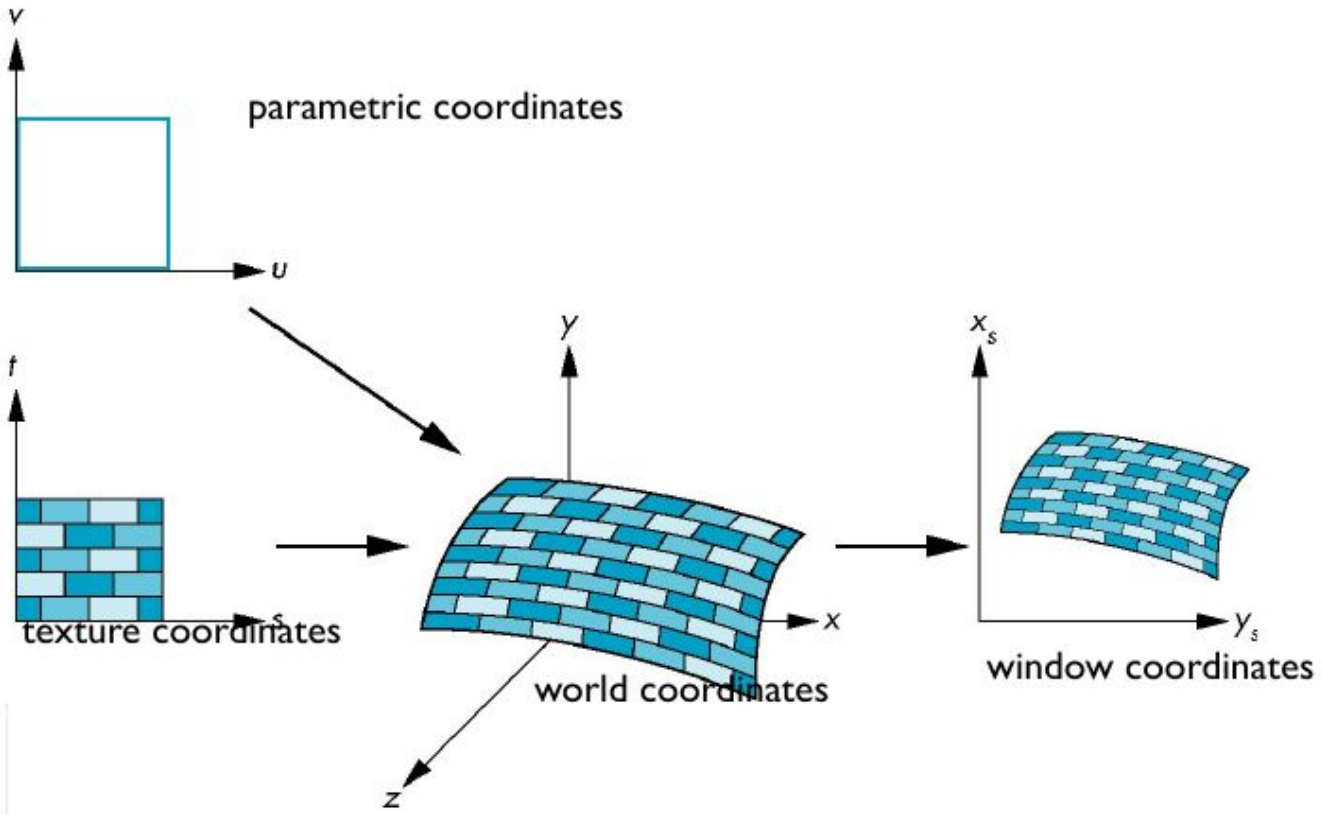
- ⋆ *Texture Mapping*: aplica um certo padrão sobre o seu objeto a partir de uma imagem
- ⋆ *Bump Mapping*: (ou mapa de altura) simula diferenças na altura ao longo da superfície, destacando imperfeições
- ⋆ *Environmental mapping*: também chamado de *reflection mapping*. Usa textura para representar reflexões na cena
- ⋆ **Shadow mapping**: semelhante o anterior, mas para sombras





http://www.rastertek.com/dx11tut40.html

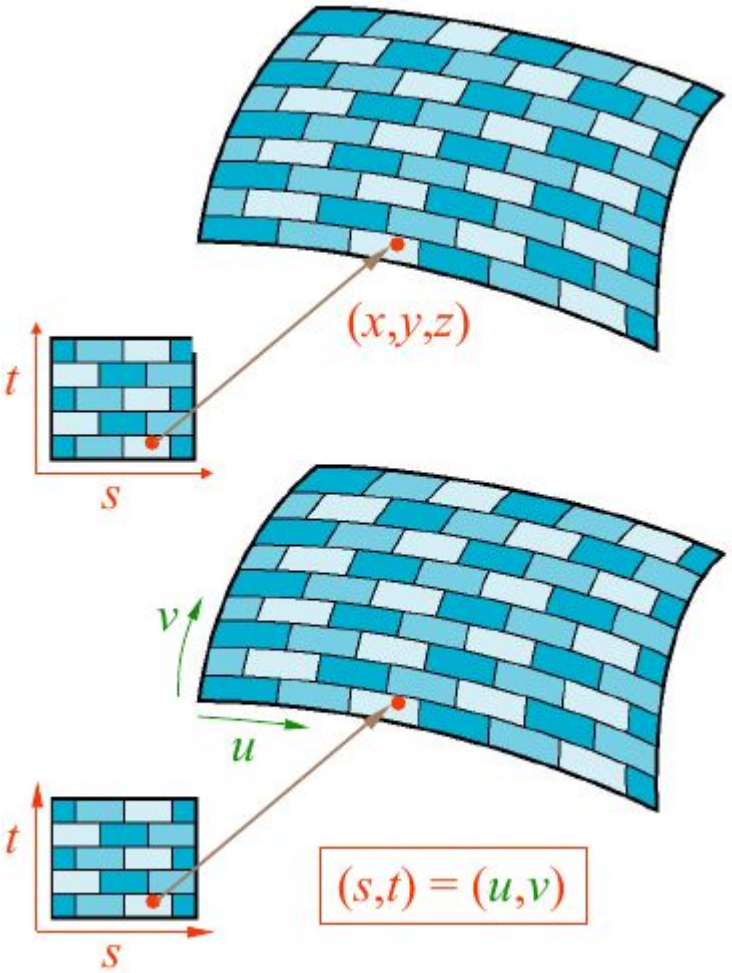
Mapeamento de textura: espaços de coordenadas



Coordenadas paramétricas: (u, v)
Coordenadas de textura: (s, t)
Coordenadas do objeto: (x, y, z)
Coordenadas de tela: (x_s, y_s)

Mapeamento de textura: considerações

Dada uma coordenada de textura (s, t), qual a coordenada no objeto (x, y, z) correspondente?

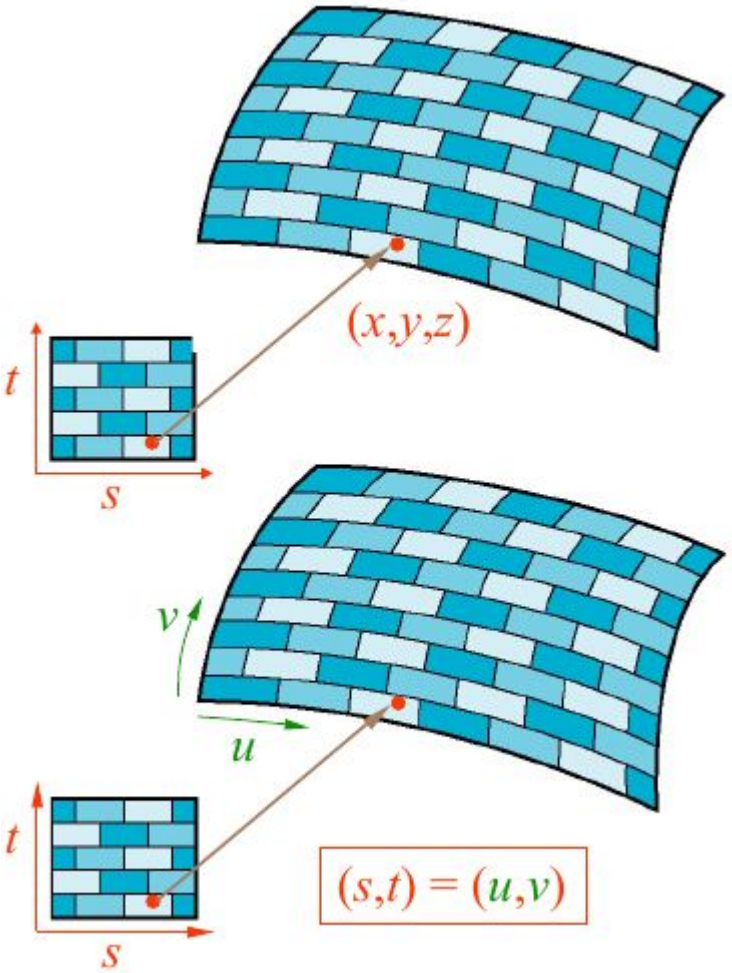


Mapeamento de textura: considerações

Dada uma coordenada de textura (s, t), qual a coordenada no objeto (x, y, z) correspondente?

Superfície paramétrica:

$$\begin{cases} x = p_x(u, v) \\ y = p_y(u, v) \\ z = p_z(u, v) \\ u, v \in [0, 1] \end{cases}$$



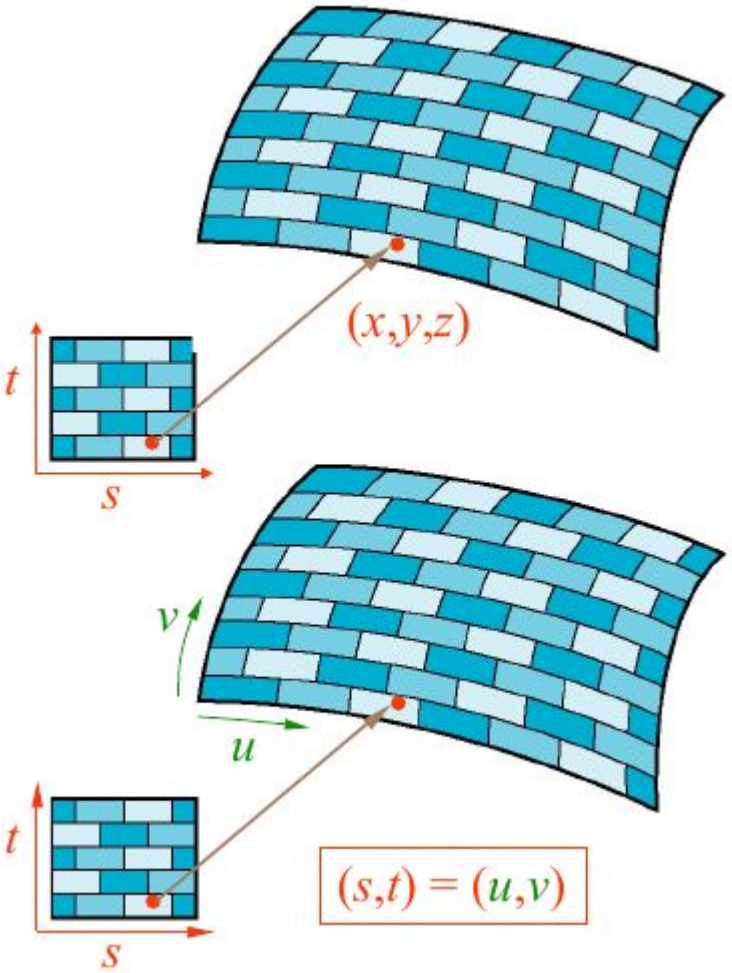
Mapeamento de textura: considerações

Dada uma coordenada de textura (s, t), qual a coordenada no objeto (x, y, z) correspondente?

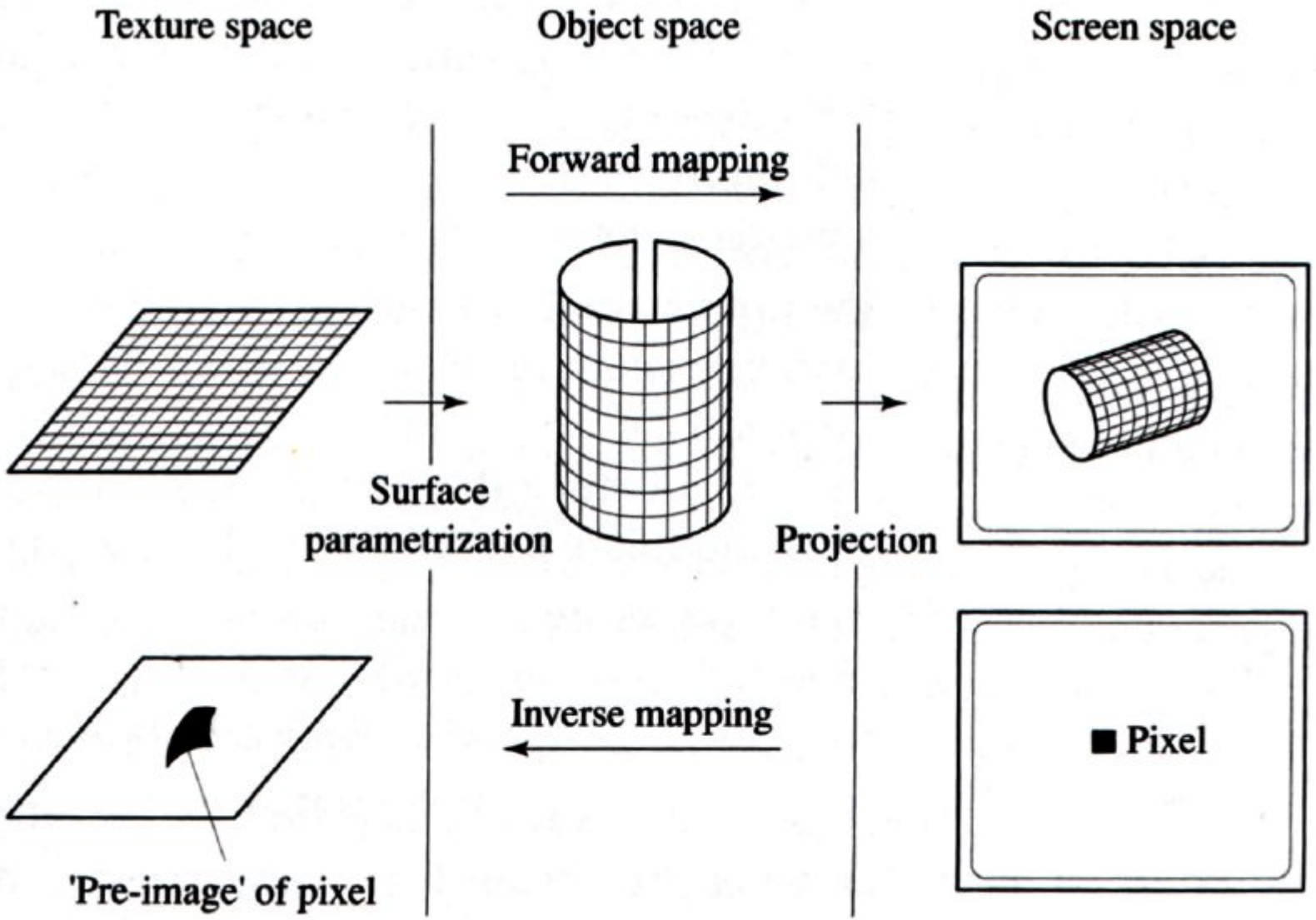
Superfície paramétrica:

$$\begin{cases} x = p_x(u, v) \\ y = p_y(u, v) \\ z = p_z(u, v) \\ u, v \in [0, 1] \end{cases}$$

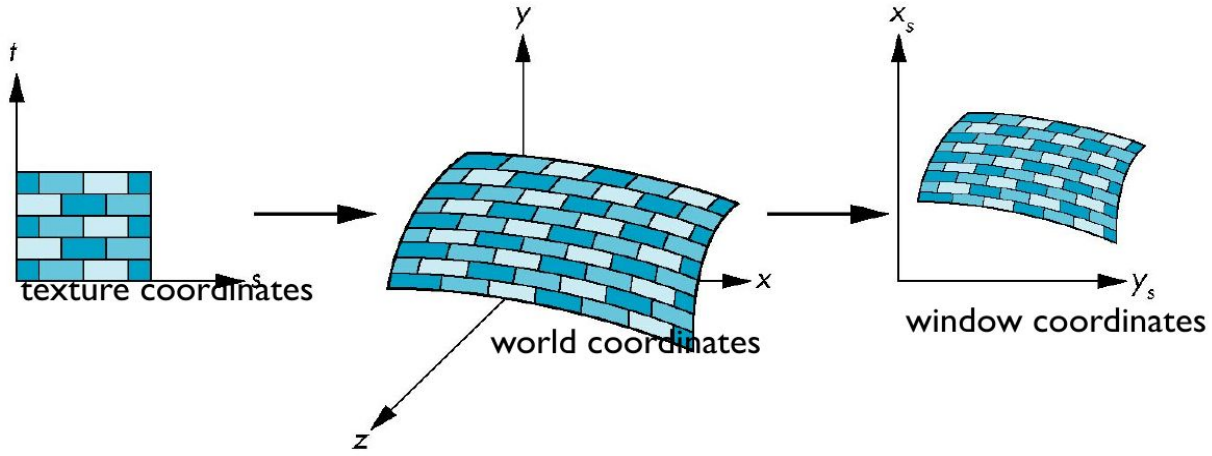
Uma possibilidade é usar $(s, t) = (u, v)$



Mapeamento de textura: duas estratégias

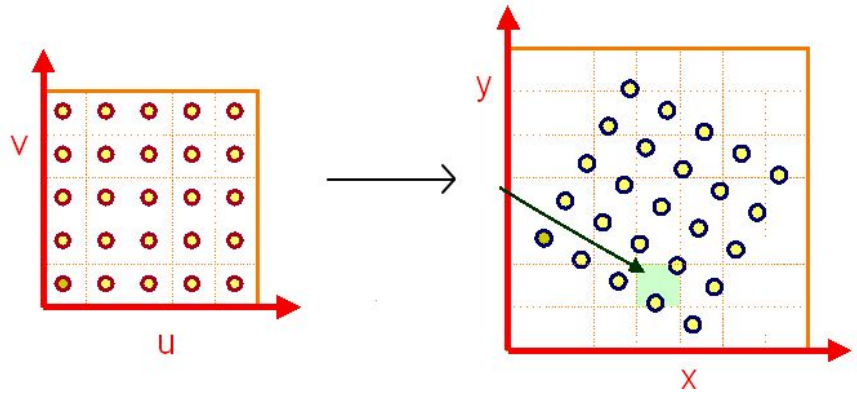


Forward mapping (mapeamento direto): $(s, t) \rightarrow (x_s, y_s)$

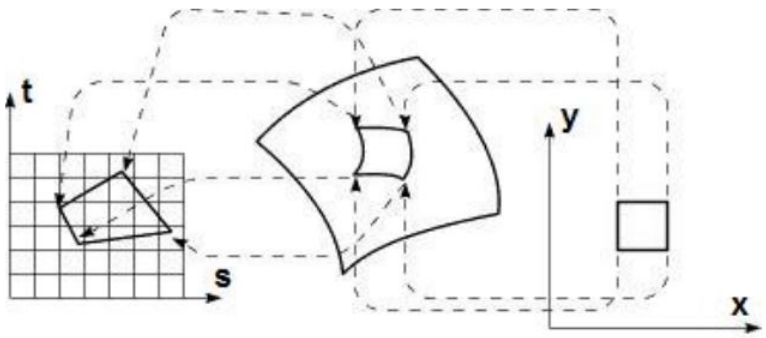


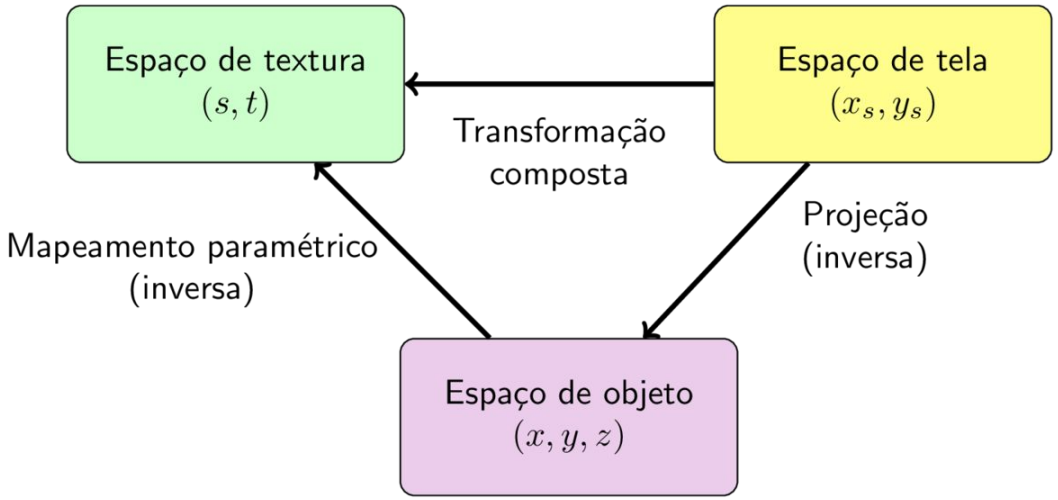
Problemas

- Texturas desnecessárias são calculadas (nem toda cena é visível)
- Pode gerar buracos e sobreposições na imagem final renderizada, semelhante ao problema de *forward warping* em processamento de imagens



Inverse mapping (mapeamento inverso): $(x_s, y_s) \rightarrow (s, t)$





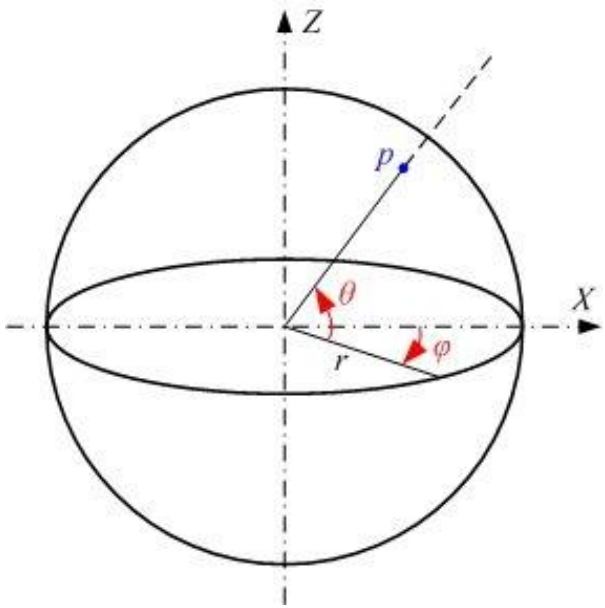
Vantagens

- Apenas texturas associadas aos fragmentos visíveis são calculadas (mais eficiente)
- Para cada coordenada (s, t) obtida, é possível obter o *texel* correspondente por *nearest neighboor* ou interpolação

Exemplo: esfera

Forma paramétrica:

$$x = \rho \cos(u) \sin(v)$$
$$y = \rho \sin(u) \sin(v)$$
$$z = \rho \cos(v)$$
$$u \in [0, 2\pi],\ v \in [0, \pi]$$

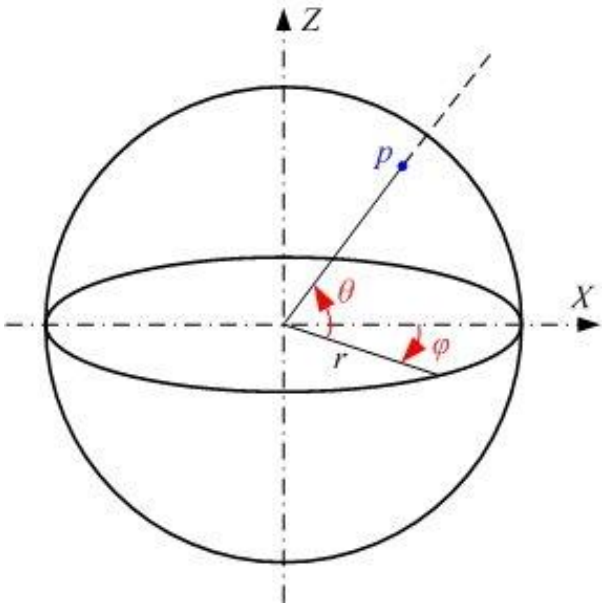


https://www.researchgate.net/profile/Pantelis_Poulakis/publication/228859507/figure/fig3/AS:669488408182790@1536629883097/FigB2-Surface-parametrization-of-a-sphere_W640.jpg

Exemplo: esfera

Forma paramétrica:

$$x = \rho \cos(u) \sin(v)$$
$$y = \rho \sin(u) \sin(v)$$
$$z = \rho \cos(v)$$
$$u \in [0, 2\pi],\ v \in [0, \pi]$$



https://www.researchgate.net/profile/Pantelis_Poulakis/publication/228859507/figure/fig3/AS:669488408182790@1536629883097/FigB2-Surface-parametrization-of-a-sphere_W640.jpg

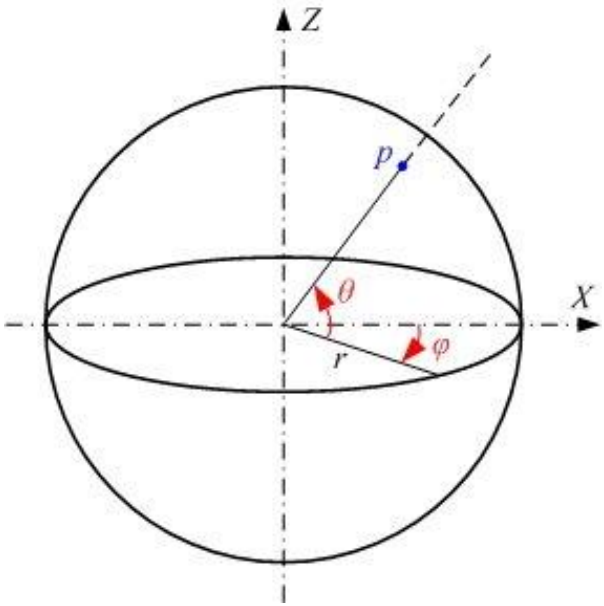
Mapeamento paramétrico inverso:

$$v = \arccos\left(\frac{z}{\rho}\right)$$
$$u = \text{atan2}(y, x)$$

Exemplo: esfera

Forma paramétrica:

$$x = \rho \cos(u) \sin(v)$$
$$y = \rho \sin(u) \sin(v)$$
$$z = \rho \cos(v)$$
$$u \in [0, 2\pi],\ v \in [0, \pi]$$



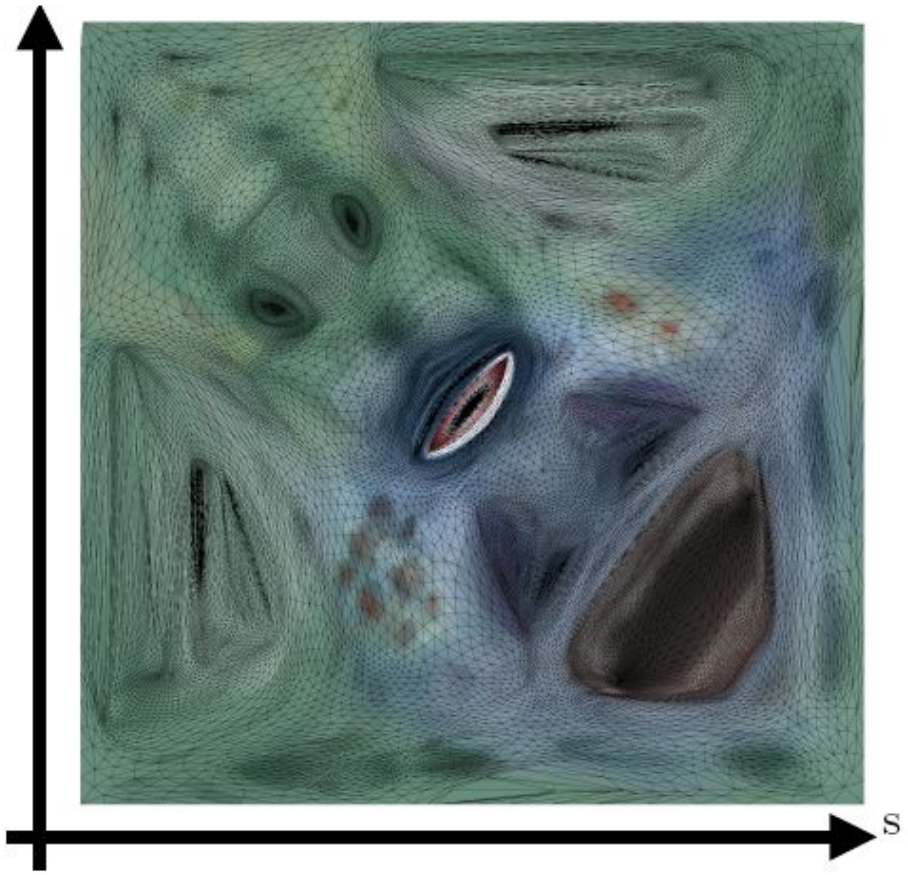
https://www.researchgate.net/profile/Pantelis_Poulakis/publication/228859507/figure/fig3/AS:669488408182790@1536629883097/FigB2-Surface-parametrization-of-a-sphere_W640.jpg

Mapeamento paramétrico inverso:

$$v = \arccos\left(\frac{z}{\rho}\right)$$
$$u = \text{atan2}(y, x)$$

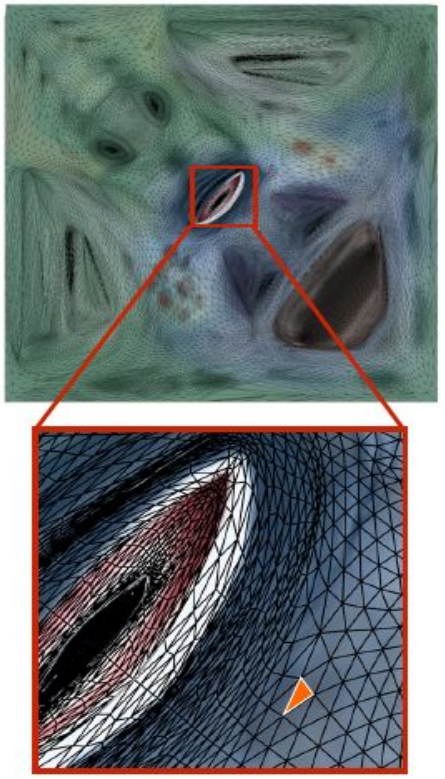
Por fim, fazemos $(s, t) = \left(\frac{1}{2\pi}u, \frac{1}{\pi}v\right)$

Mapeamento de textura em malha de triângulos



Mapeamento de textura em malha de triângulos

- Para cada vértice $V_i = (x_i, y_i, z_i)$, tem-se um coordenada de textura (s_i, t_i) associada



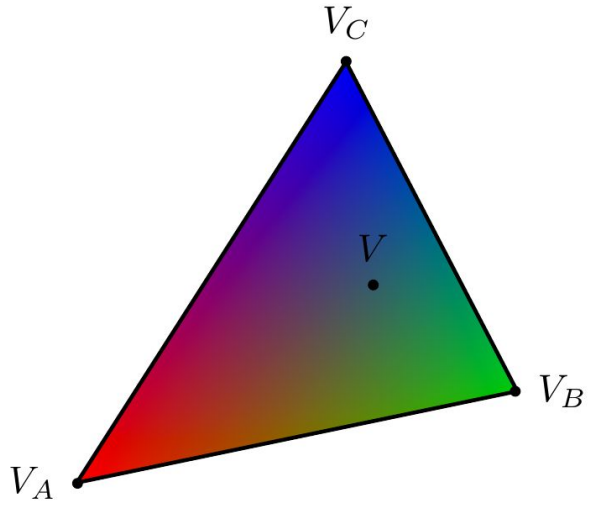
- No processo de rasterização, São usadas coordenadas baricêntricas em cada face para calcular (s, t) (com correção perspectiva)

$V = \lambda_A \mathcal{P}(V_A) + \lambda_B \mathcal{P}(V_B) + \lambda_C \mathcal{P}(V_C) \rightarrow$ calcular λ's

$\gamma = \frac{\lambda_A}{z_A} + \frac{\lambda_B}{z_B} + \frac{\lambda_C}{z_C} \rightarrow$ calcular fator de correção

$$s = \gamma \left(s_A \frac{\lambda_A}{z_A} + s_B \frac{\lambda_B}{z_B} + s_C \frac{\lambda_C}{z_C} \right)$$

$$t = \gamma \left(t_A \frac{\lambda_A}{z_A} + t_B \frac{\lambda_B}{z_B} + t_C \frac{\lambda_C}{z_C} \right)$$



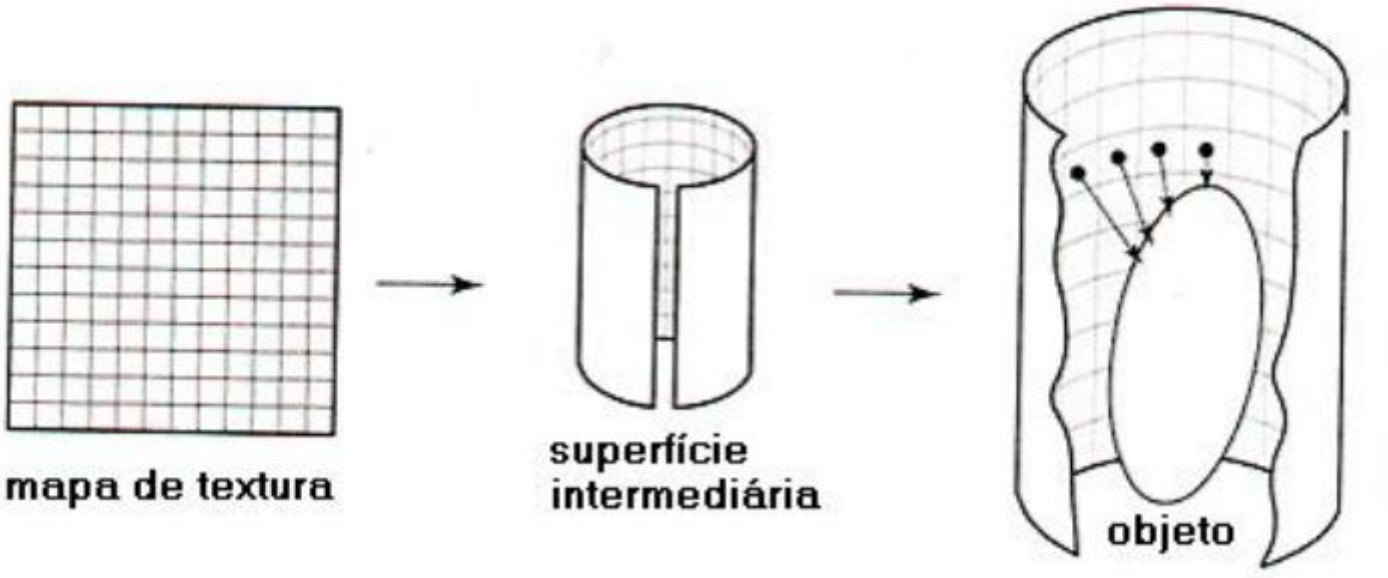
Mapeamento de textura em malha de triângulos

Para objetos muito complexos, talvez seja muito complicado fazer o mapeamento de textura sobre a malha de triângulos diretamente

Mapeamento de textura em malha de triângulos

Para objetos muito complexos, talvez seja muito complicado fazer o mapeamento de textura sobre a malha de triângulos diretamente

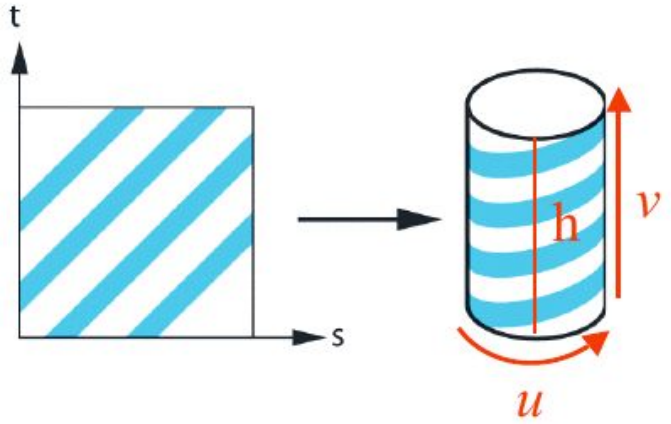
Possível fazer mapeamento em dois estágios:

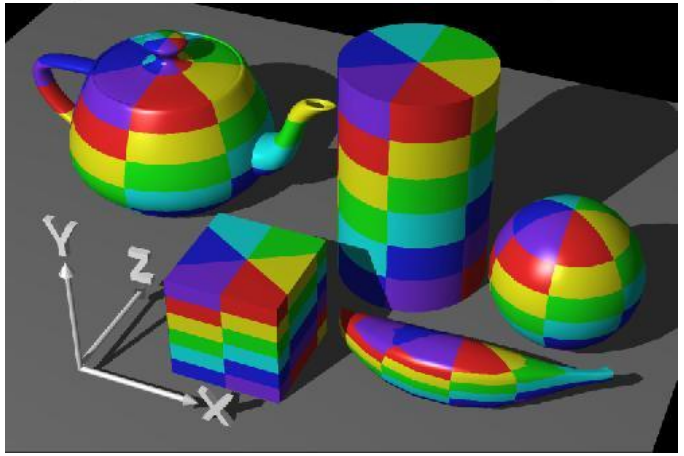


Mapeamento em dois estágios (cilindro)

Forma paramétrica do cilindro:

$$\begin{cases} x = r\cos(u) \\ y = v \\ z = r\sin(u) \\ u \in [0, 2\pi],\ v = [0, h] \end{cases}$$



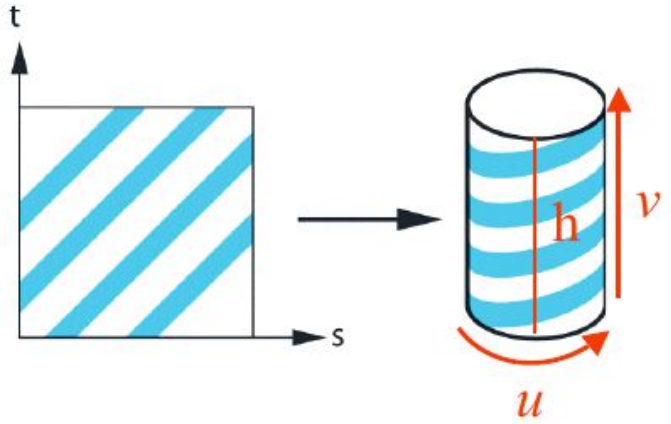


https://education.siggraph.org/static/HyperGraph/mapping/surface0.htm

Mapeamento em dois estágios (cilindro)

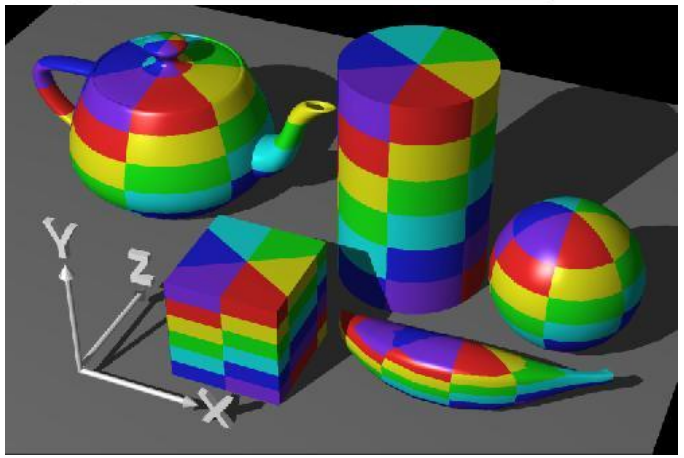
Forma paramétrica do cilindro:

$$\begin{cases} x = r\cos(u) \\ y = v \\ z = r\sin(u) \\ u \in [0, 2\pi],\ v = [0, h] \end{cases}$$



Mapeamento paramétrico inverso:

$$\begin{cases} u = \text{atan2}(z, x) \\ v = y \end{cases}$$

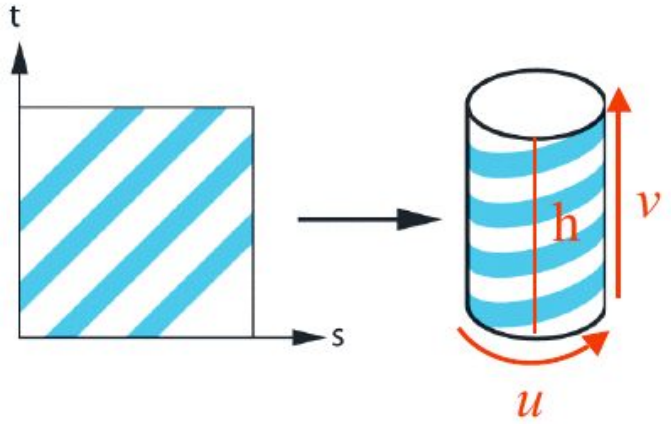


https://education.siggraph.org/static/HyperGraph/mapping/surface0.htm

Mapeamento em dois estágios (cilindro)

Forma paramétrica do cilindro:

$$\begin{cases} x = r\cos(u) \\ y = v \\ z = r\sin(u) \\ u \in [0, 2\pi],\ v = [0, h] \end{cases}$$



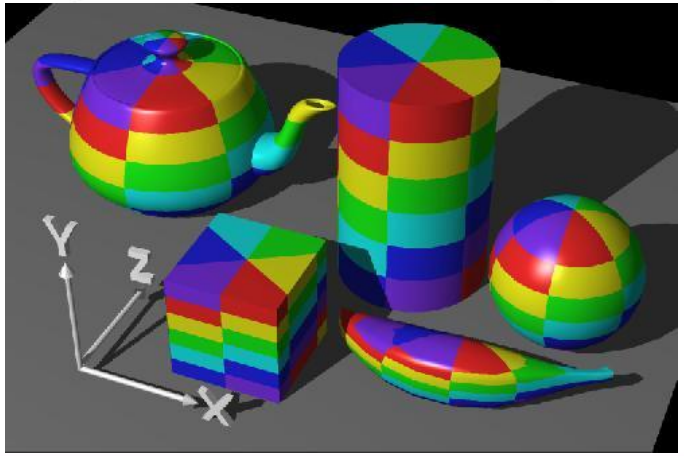
Mapeamento paramétrico inverso:

$$\begin{cases} u = \text{atan2}(z, x) \\ v = y \end{cases}$$

Finalmente, obtemos a coordenada de textura:

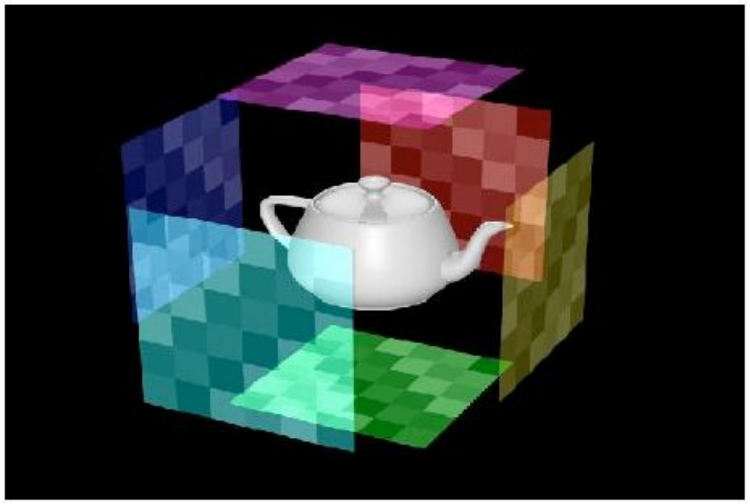
$$\begin{cases} s = \frac{1}{2\pi}u \\ t = \frac{1}{h}v \end{cases}$$

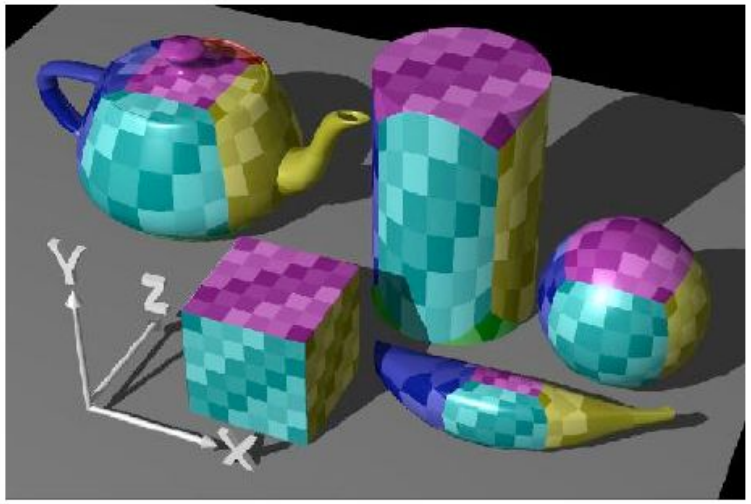
https://education.siggraph.org/static/HyperGraph/mapping/surface0.htm



Mapeamento em dois estágios: outros tipos

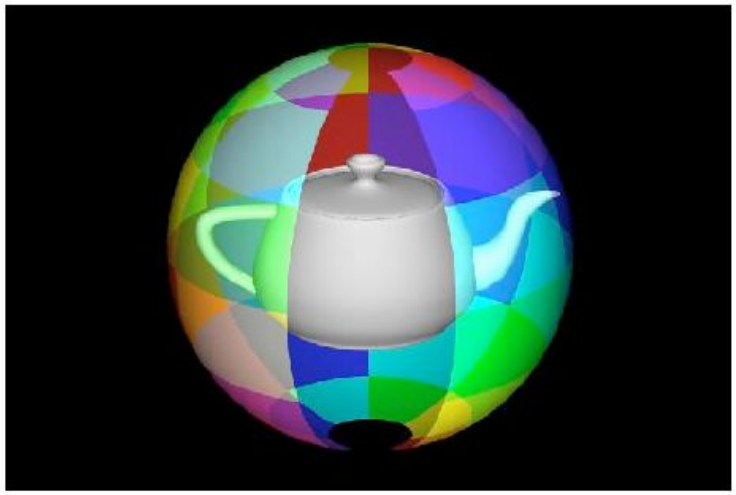
Cube map projection

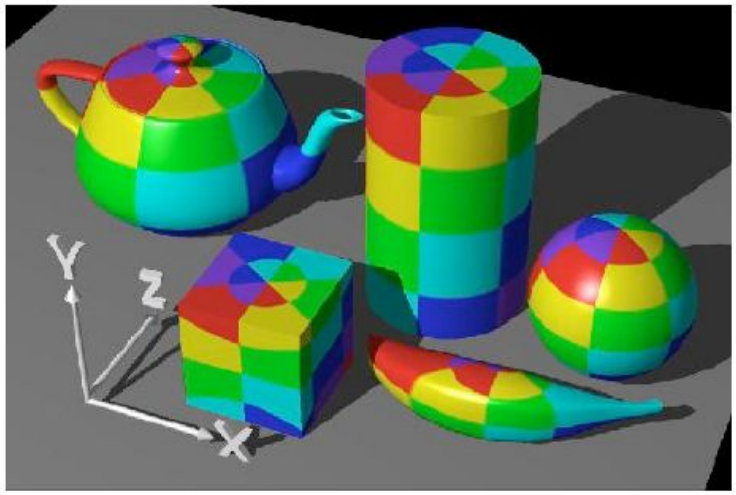




Spherical projection

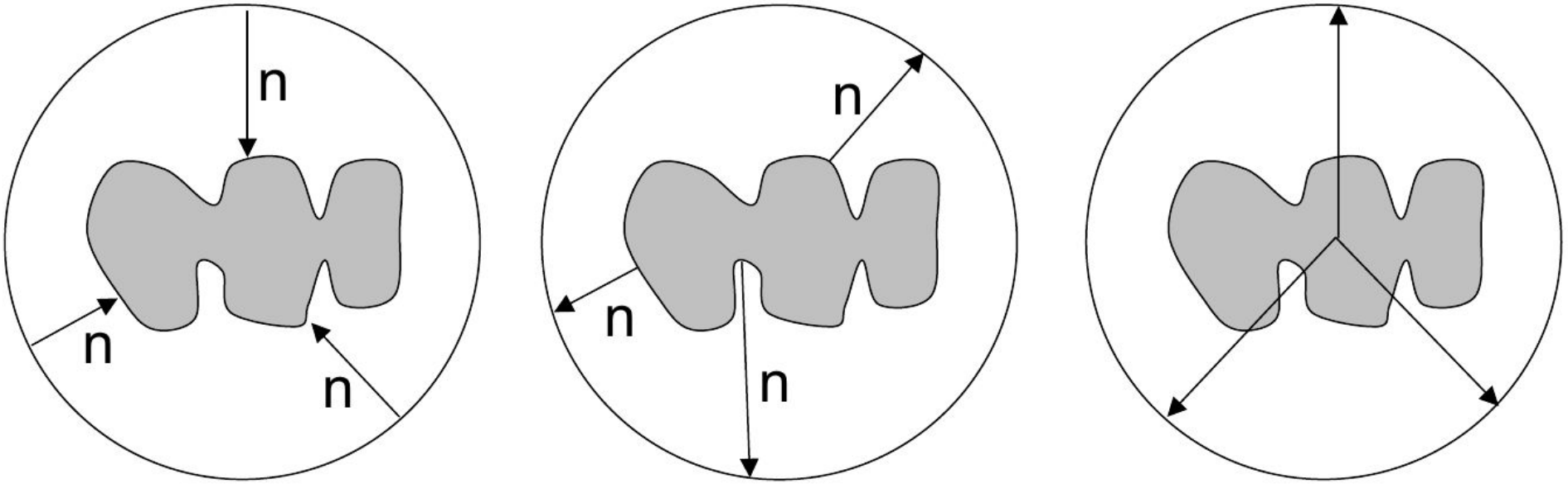
https://education.siggraph.org/static/HyperGraph/mapping/surface0.htm





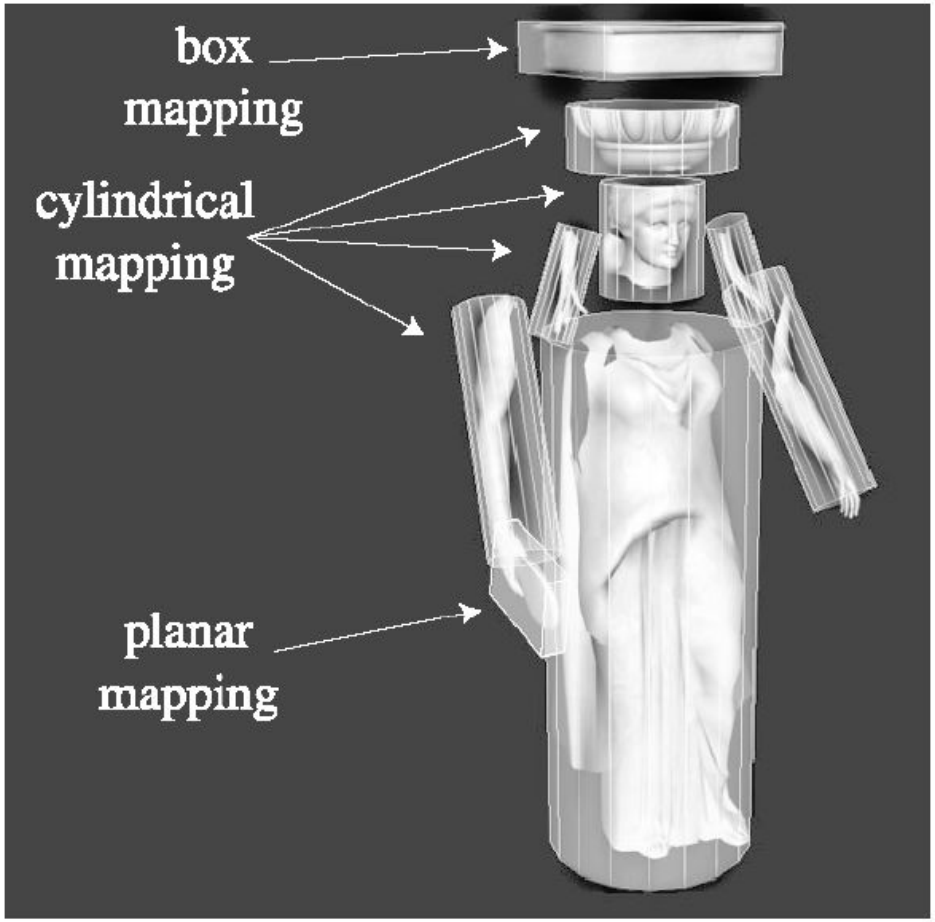
Mapeamento em dois estágios

Superfície intermediária → superfície



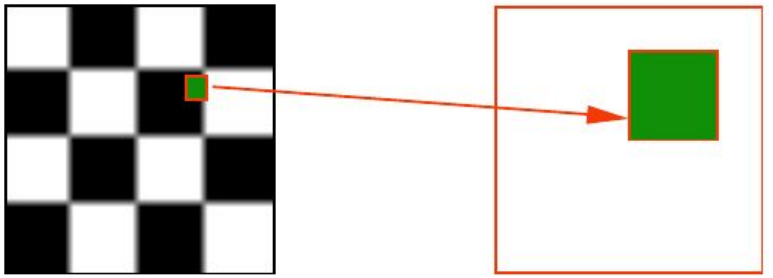
Mapeamento em dois estágios

Exemplo



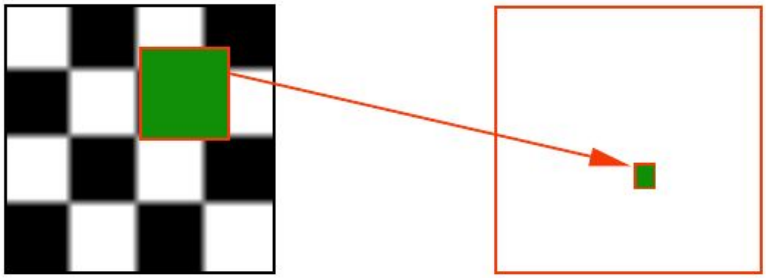
[Tito Pagan]

Filtragem de textura



Magnification

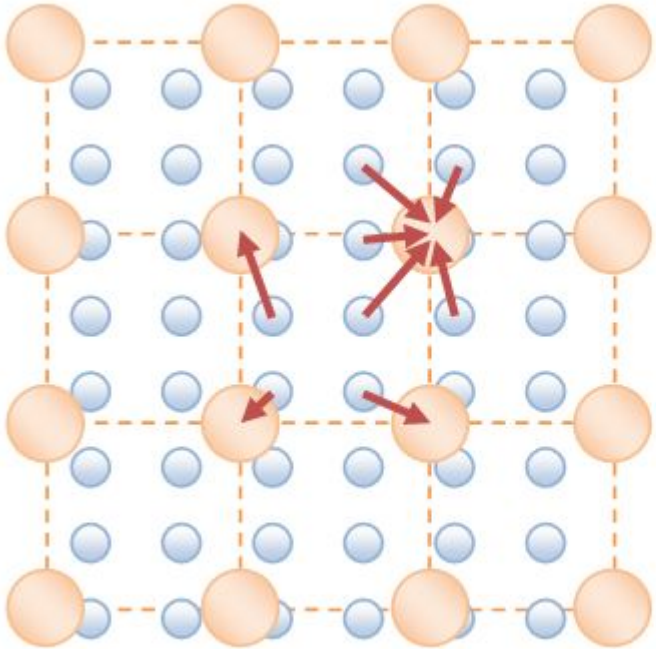
- Cada *pixel* é associado com uma região pequena da superfície e da textura



Minification

- Há 3 possibilidades:
  1. Um *texel* para um *pixel* (raro)
  2. *Magnification:* um *texel* para muitos *pixels*
  3. *Minification:* muitos *texel* para um *pixel*

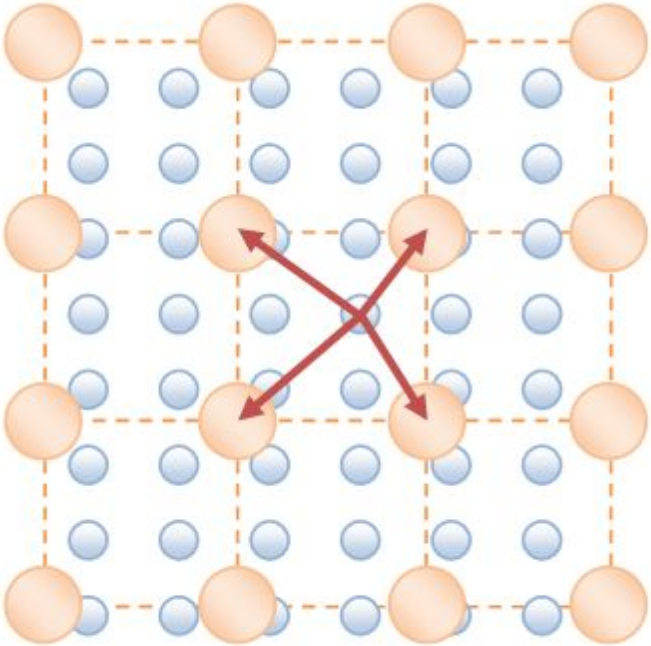
Filtragem de textura: *Magnification*



texel

fragment

Magnification – Nearest Point Sampling

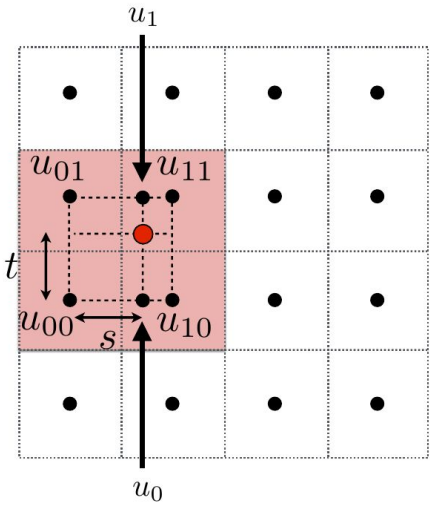


texel

fragment

Magnification – Bilinear Interpolation

https://www3.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html



$$\mathbf{u} = (\text{round}(s), \text{round}(t))$$

$$\mathbf{u}_0 = \text{Lerp}(s, \mathbf{u}_{00}, \mathbf{u}_{10})$$
$$\mathbf{u}_1 = \text{Lerp}(s, \mathbf{u}_{01}, \mathbf{u}_{11})$$
$$\mathbf{u} = \text{Lerp}(t, \mathbf{u}_0, \mathbf{u}_1)$$

Filtragem de textura: *Minification*

https://mlhr8q6s8c91.i.optimole.com/VVLFqVU-fxiu5NwV/w:768/h:512/q:90/https://www.imaginationtech.com/wp-content/uploads/2019/08/example_mipmap_chain-1.png

Texture *wrapping*

O que fazer se não tivermos $0 \leq s, t \leq 1$?

GL_REPEAT

GL_MIRRORED_REPEAT

GL_CLAMP_TO_EDGE

GL_CLAMP_TO_BORDER

Dúvidas?

- João Vitor de Oliveira Silva
- jvos@dcc.ufrj.br ou jvos@lncc.br