

# Mapeamento de Ambiente

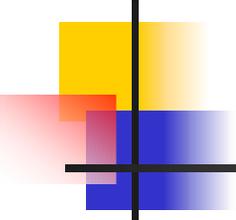
---

IA725 – Computação Gráfica I

Prof<sup>a</sup>. Wu Shin Ting

Carlos Henrique da Silva Santos

Milena Alexandre dos Santos



# Tópicos

---

Introdução

Texturas

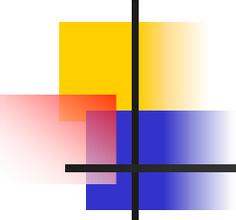
Mapeamento de Ambiente

Mapa do Ambiente

Aplicações - Evolução

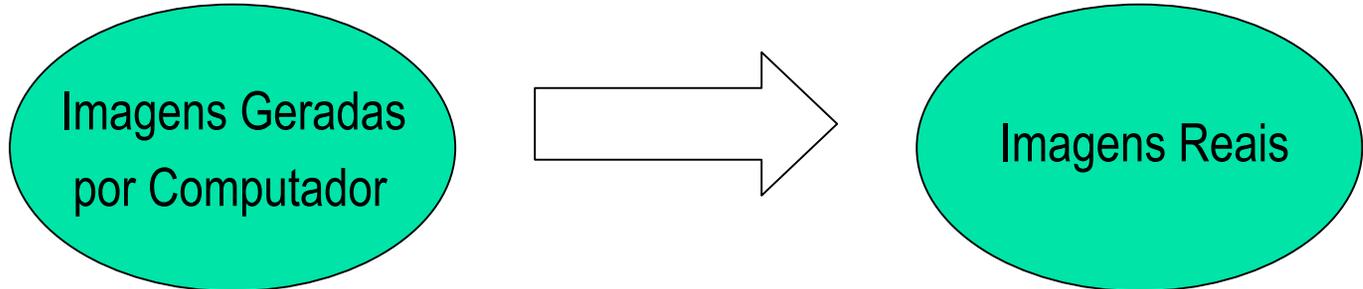
OpenGL

Conclusão



# Introdução

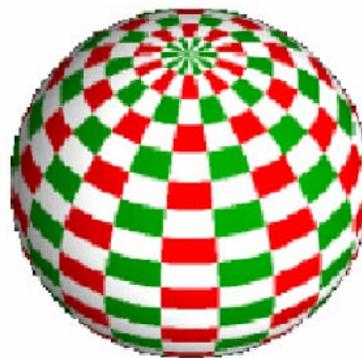
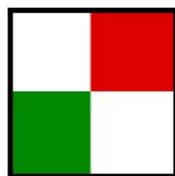
---

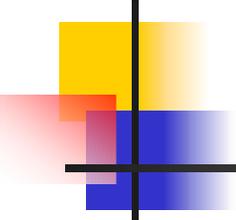


- Técnicas de texturização permitem criar imagens visualmente interessantes e ricas, sem a necessidade de produzir descrições complexas das superfícies dos objetos.

# Textura

- A idéia é reproduzir sobre a superfície de algum objeto da cena as propriedades de alguma função ou mapa – bidimensional.
- Texturas 2D são funções  $F(s,t)$  cujo domínio é um espaço bidimensional, e o contradomínio pode ser cor, opacidade, entre outros [1].





# Textura

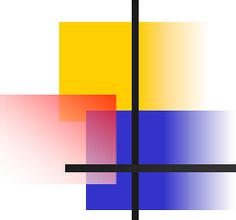
---

- Não é possível mapear uma textura 2D num objeto poligonal de forma arbitrária sem se “recortar” a textura, pois existem as seguintes dificuldades:

- Encontrar a função de mapeamento, ou modelação:

$$(u,v) = F(x, y, z)$$

- Derivar um método através do qual o *render* pode associar valores de textura ao pontos internos do polígono quando este é recortado.



# Textura

---

- Cilíndrico:
- Representação de pontos da superfície cilíndrica de raio  $r$  e altura  $h$ :

$$(r \cos \theta, r \sin \theta, hz)$$

$$0 < \theta < 2\pi$$

$$0 < z < 1$$

- Associar os valores da textura com um ponto do cilindro:

$$(u, v) = \left(\frac{\theta}{2\pi}, z\right)$$

$$u, v \in [0, 1]$$

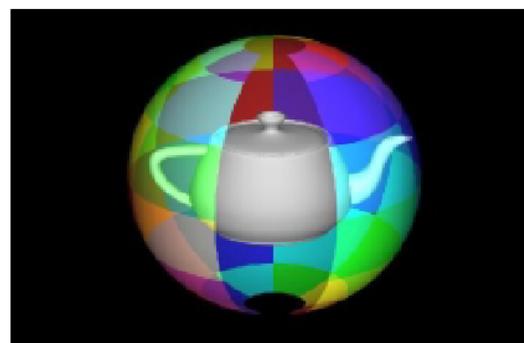
# Textura

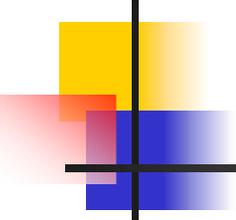
1º-Mapeamento-S, do espaço 2D da textura num espaço 3D (por exemplo cilíndrico):

$$T(u, v) \rightarrow T'(x_i, y_i, z_i)$$

2º-Mapeamento-O do espaço 3D da textura no espaço 3D da superfície:

$$T'(x_i, y_i, z_i) \rightarrow O(x, y, z)$$





# Textura

---

- Mapeamento S:

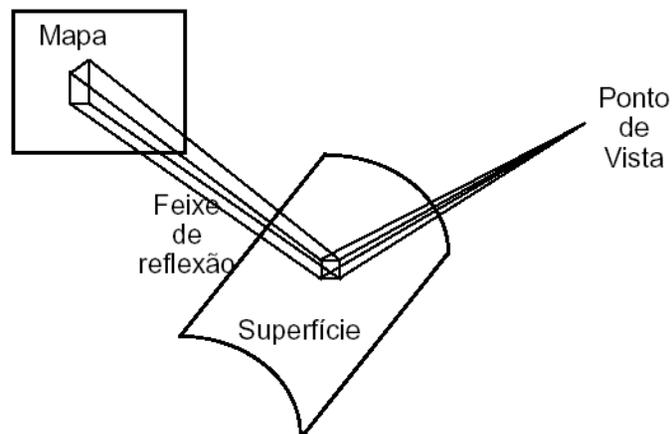
- Escolher entre, a superfície de um cilindro, as faces de um cubo ou a face de uma esfera.

- Mapeamento O:

- Intersecção do raio refletido do ponto do observador com a superfície intermediária  $T'$ ;

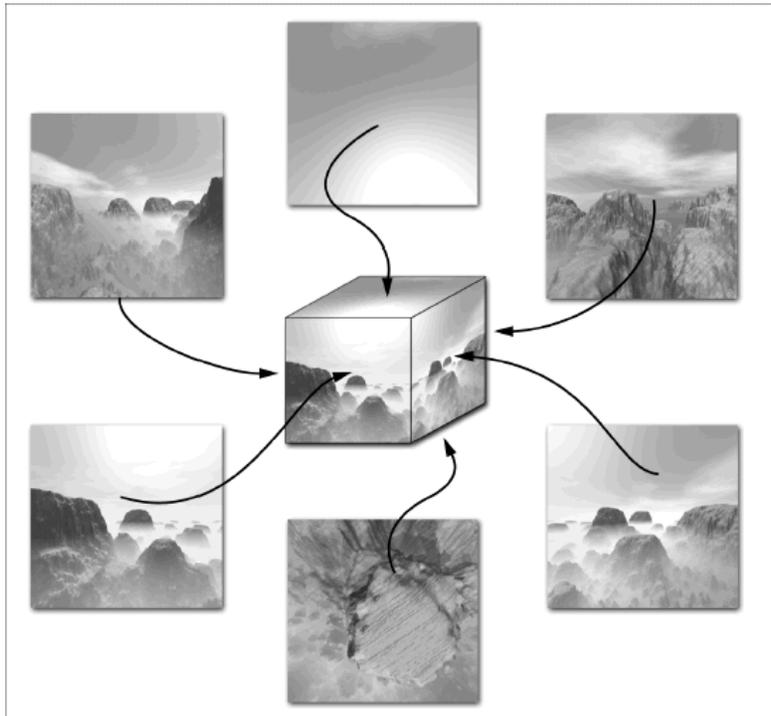
# Mapeamento de Ambiente

- Definição: onde ocorre a simulação de um objeto refletindo o seu arredor e supõe-se que o ambiente de um objeto (isto é, tudo que o cerca) está infinitamente distante dele e, pode conseqüentemente ser codificado em uma imagem conhecida, como mapa do ambiente [19].



# Mapa do Ambiente

- Formas de adquirir o mapa do ambiente:



A primeira maneira requer seis imagens da textura, cada uma corresponderá a uma faceta de um cubo, que representa o ambiente circunvizinho.



# Mapa do Ambiente

$$\mathbf{r} \rightarrow (s, t)$$

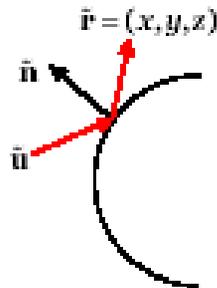
Reflection vector:

$$\tilde{\mathbf{r}} = \tilde{\mathbf{u}} - 2(\tilde{\mathbf{n}} \cdot \tilde{\mathbf{u}})\tilde{\mathbf{n}}$$

Resulting texture coordinates:

$$s = \frac{x}{2m} + \frac{1}{2} \quad t = \frac{y}{2m} + \frac{1}{2}$$

$$m = \sqrt{x^2 + y^2 + (z+1)^2} \\ = \sqrt{2(z+1)}$$



$\tilde{\mathbf{n}}$  : unit normal

$\tilde{\mathbf{u}}$  : unit viewing direction

$\tilde{\mathbf{r}}$  : unit reflected direction

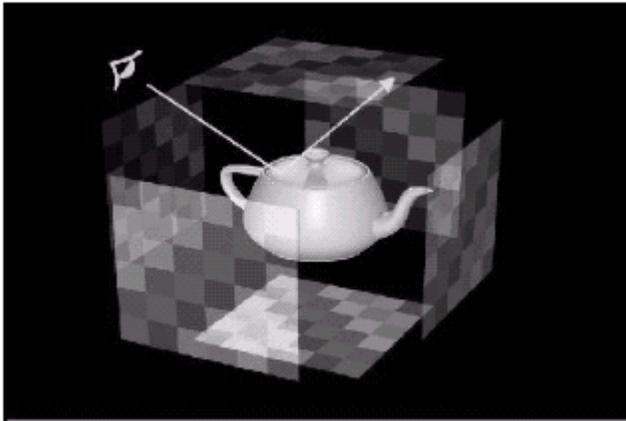
All in eye space!



O segundo método deve gerar uma única imagem da textura que corresponderá a uma esfera que simula o ambiente refletido. A imagem é aquela de uma esfera perfeita refletindo o ambiente no qual ela está situada.

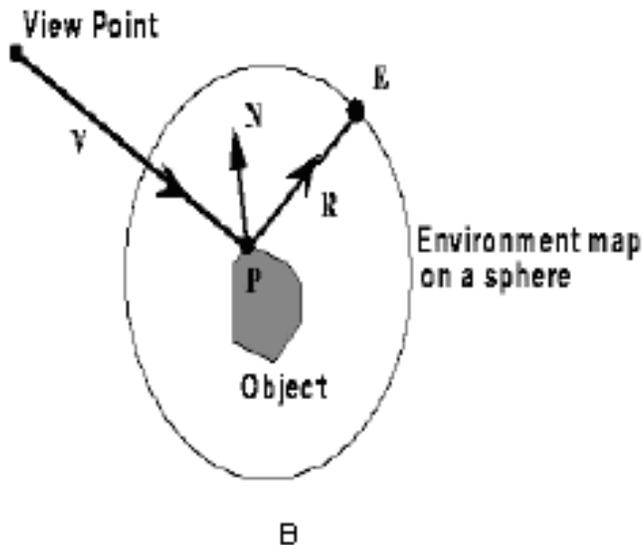
# Mapa do Ambiente

- Aplicando o mapa do ambiente:



Em cada vértice do polígono que representará o ambiente mapeado, é calculado um vetor da reflexão do observador da superfície. Este vetor da reflexão posiciona uma das seis imagens da textura. Quando todos os vértices do polígono gerarem reflexões na mesma imagem, ela será mapeada no polígono através da projeção da textura. Se um polígono tiver reflexões em mais de uma faceta do cubo, o polígono será subdividido em partes, onde cada uma das partes gera reflexões em somente uma faceta.

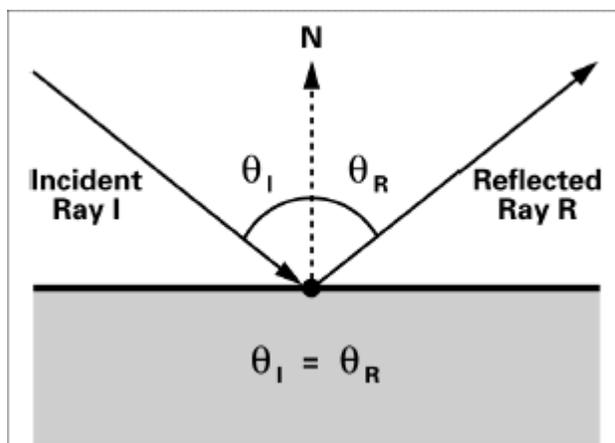
# Mapa do Ambiente



Para aplicar a textura deve-se olhar cada texel por vez e para cada um determinar o sentido do raio de reflexão correspondente. Utilizar a cor do ambiente visto, nesta direção, para a cor do texel.

# Mapa do Ambiente

- Calculando Vetores de Reflexão:

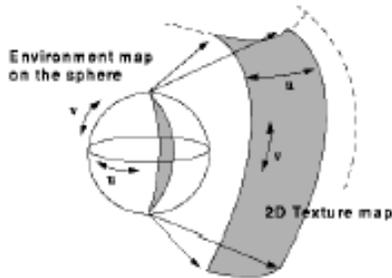


$$R = I - 2N(N \cdot I)$$

Imagina-se um objeto, um observador, e um mapa cúbico do ambiente que cerca o objeto. O vetor  $I$  é chamado raio incidente do observador à superfície do objeto. Quando  $I$  alcança a superfície, reflete-se no sentido  $R$  baseado na normal da superfície  $N$ . Este segundo raio é o raio refletido.

# Mapeamento de Ambiente

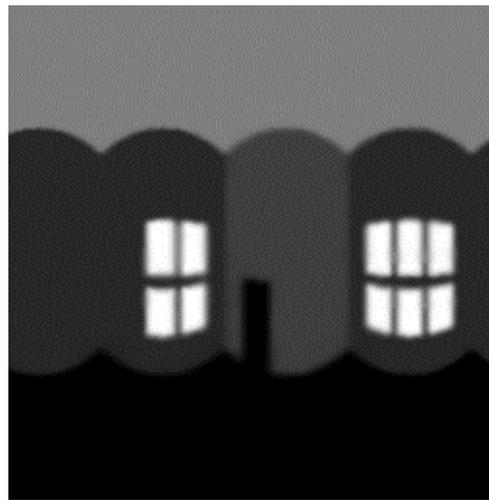
- Aplicações – Evolução:
- 1976 – Jim Blinn: Mapa do ambiente.



$$u = \frac{\text{atan}(R_x/R_z)}{2\pi}$$

$$v = \frac{\text{asin } R_y + \pi/2}{\pi}$$

Mapeamento de Latitudes



Mapa do Ambiente



Objeto

# Mapeamento de Ambiente

- 1982 /1983 – Michael Chou: Fotografias reais como mapa de ambiente.



Mapeamento de Ambiente de Reflexão usado para colocar objetos numa cena.

# Mapeamento de Ambiente

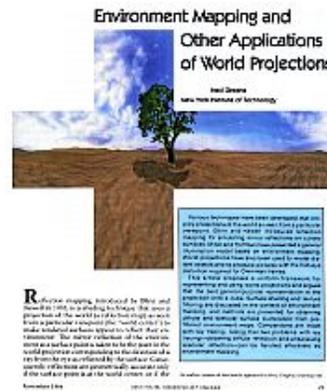
- 1985 – Lance Williams: mapeamento de ambiente de reflexão numa cena em movimento com um elemento animado por Computação Gráfica.
- 1986 - Flight of the Navigator de Randal Kleiser (1º filme a utilizar a técnica).



Nave do Filme

# Mapeamento de Ambiente

- 1986 – Ned Greene: combinou uma imagem real com uma imagem gerada no computador para criar um mapa cúbico do ambiente.



Mapa Cúbico do Ambiente

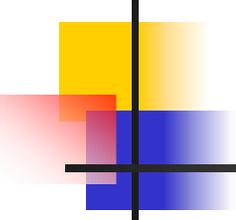
# Mapeamento de Ambiente

- 1991 – Filme Exterminador do Futuro II de James Cameron.



Robô do filme

- 1993 - Paul Haeberli: imagem real, mapa do ambiente.



# OpenGL

---

- Comandos de Textura:

- Especificar a textura:

*glTexImage()* ;

- Indicar como a textura será aplicada para cada pixel:

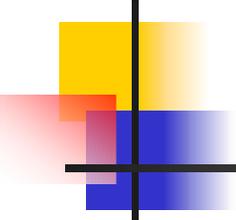
*glTexEnv()* ;      *glTexParameter()*;

- Habilitar o mapeamento de textura:

*glEnable()*;

- Desenhar a cena, fornecendo as coordenadas geométricas e as coordenadas de textura:

*glTexCoord()*.



# OpenGL

---

- Mapeamento de Ambiente:

- Definir, automaticamente, coordenadas do Mapa Esférico:

```
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_SPHERE_MAP);
```

```
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_SPHERE_MAP);
```

- Habilitar Mapa Esférico:

```
glEnable(GL_TEXTURE_GEN_S);
```

```
glEnable(GL_TEXTURE_GEN_T);
```

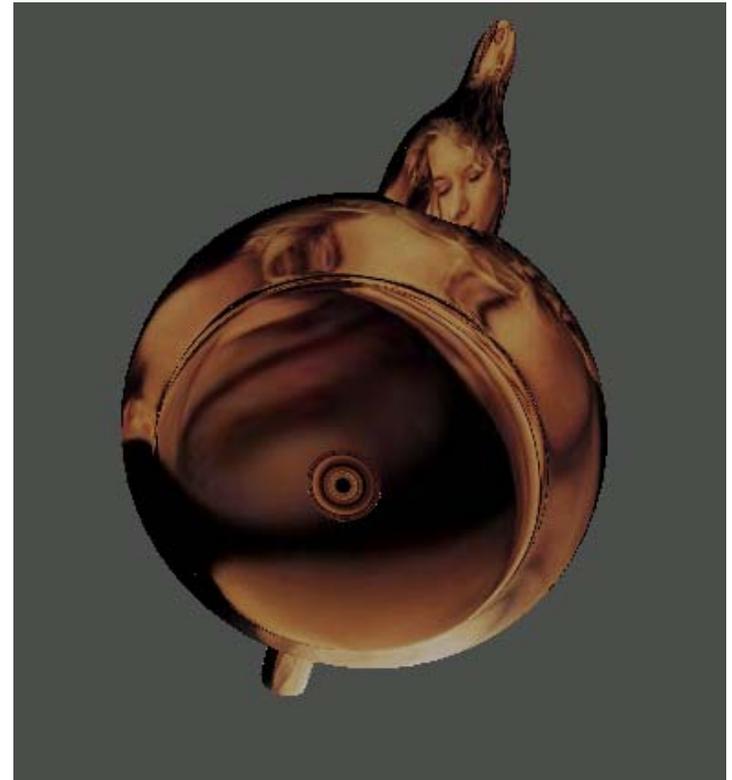
# OpenGL

- Mapeamento de Textura x Mapeamento de Ambiente:



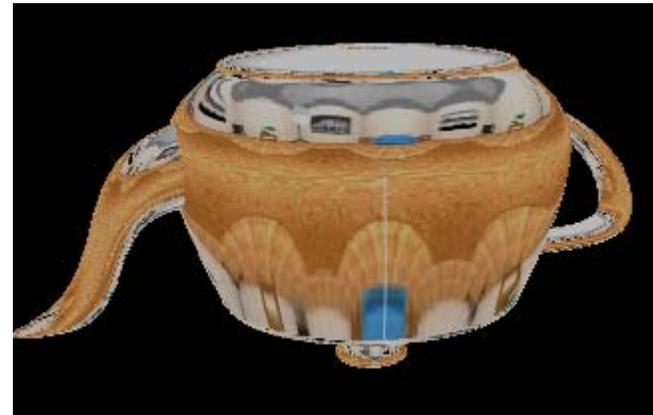
# OpenGL

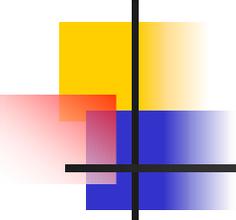
- Mapeamento de Ambiente - Exemplos:



# OpenGL

- Mapeamento de Ambiente - Exemplos:

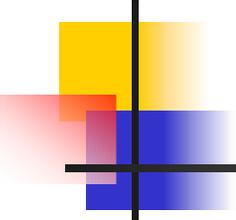




# Conclusão

---

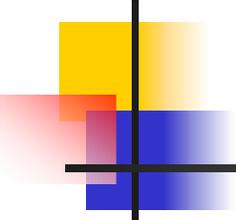
- Dificuldade na formação do mapa do ambiente;
- Reflexão apenas do ambiente;
- Melhor desempenho em superfícies curvas.



# Referências

---

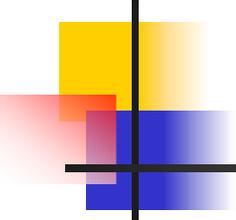
- [1] CAVALCANTI, P.R.; ESPERANÇA, C. Apostila de Introdução à Computação Gráfica. UFRJ – 2000.
- [2] TING, W.S. Notas de Aula da disciplina Computação Gráfica I. FEEC – Unicamp – 2004.
- [3] WILSON, R. SPANN, M. Image segmentation and uncertainty. In: Kittler, J. ed. Pattern Recognition and Image Processing Series, Herts: RSP, 1988. v. 9, p. 354-385.
- [4] JAIN, A.K. Fundamentals of digital image processing. Englewood Cliffs: Prentice Hall, 1988, 592 p.
- [5] Institute of Electrical and Electronics Engineers (IEEE). IEEE standard glossary of image processing and pattern recognition terminology. New York, 1990. 16 p. (IEEE Standard 610.4-1990).
- [6] GONZALEZ, R.C.; WOODS, R.E. Digital image processing. Reading: Addison-Wesley, 1992, 716 p.
- [7] FAUGERAS, O.D.; PRATT, W.K. Decorrelation methods of texture feature extraction. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 2, n. 4, p. 323-332, 1980.
- [8] CROSS, G.R.; JAIN, A.K. Markov random fields texture models. In: PRIP' 81 IEEE Computer Society Conference on Pattern Recognition and Image Processing, Dallas, 1981. Proceedings. Piscataway: IEEE, 1981, p. 597-602.
- [10] MENDONÇA, M.B. Dissertação de Mestrado: Aplicação de Texturas em Visualização Científica. USP – 2001.



# Referências

---

- [11] BLINN, J.; NEWELL, M. Texture and Reflection in Computer Generated Images. Communications of the ACM 19:10 (1976), 542—547.
- [12] CROW, F.C. “Texture” in ROGERS, D.F.; EARNSHAW, R.A. (eds), Computer Graphics Techniques – Theory and Practice pp. 159-187, Springer-Verlag, 1990.
- [13] SEGAL, M.; AKELEY, K. The Design of the OpenGL Graphics Interface. Technical report, Silicon Graphics Inc, [on-line] [http://www.opengl.org/developers/documentation/white\\_papers/opengl/index.html](http://www.opengl.org/developers/documentation/white_papers/opengl/index.html), 1996.
- [14] NEIDER, J.; DAVIS, T.; MASON, W. OpenGL Programming Guide (AddisonWesley, 1993).
- [15] KILGARD, M.J. OpenGL and X, Part 1: An Introduction. Technical report, SGI, <http://www.sgi.com/software/opengl/glandx/intro/intro.html>, 1994.
- [16] KILGARD, M.J. The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3. Technical report, SGI, <Http://www.opengl.org/developers/documentation/glut/spec3/spec3.html>, 1996.
- [17] GTK+ The GIMP Toolkit. <http://www.gtk.org/>, 2002.
- [18] SEGAL, M.; AKEEY, K. The OpenGL Graphics Interface. Technical report, Silicon Graphics Inc, [http://www.opengl.org/developers/documentation/white\\_papers/oglGraphSys/opengl.html](http://www.opengl.org/developers/documentation/white_papers/oglGraphSys/opengl.html), 1994.



# Referências

---

- [19] RANDIMA, F.; KILGARD, M.J. The Cg Tutorial: The Definitive Guide to Programmable Real-time Graphics Cap. Sete (Addison Wesley)
  - [20] AZEVEDO, E.; CONCI, A. Computação Gráfica: Teoria e Prática p.314-323, Campus – 2003.
  - [21] WILLIAMS, L. "Pyramidal Parametrics," Computer Graphics (SIGGRAPH), vol. 17, No. 3, Jul. 1983 pp. 1-11.
  - [22] GREENE, N. Environment Mapping and Other Applications of World Projections. IEEE Computer Graphics and Applications, Vol 6. No. 11. Nov. 1986.
  - [23] HAERBERLI, P.; SEGAL, M. Texture Mapping as a Fundamental Drawing Primitive. Fourth Eurographics Workshop on Rendering. June 1993, pp. 259-266.
- 

## Dúvidas?