aula 22
Texturas
IC/UFF – 2017

Aura

Texturas: Texture mapping – permite dar a uma face plana um aparência bem complexa!

Edwin Catmull em 1974, em sua tese de doutorado, foi o primeiro a adicionar detalhes de textura na superfície de modelos 3D por mapeamentos

Fazer um **texture map** em uma superfície é como aplicar uma **folha de papel auto adesivo nela "contact" para lhe dar um aspecto** semelhante **a textura do** desenho deste papel !

Edwin Earl "Ed" Catmull (1945, ...)

Americano formado em **ciência da computação e atualmente presidente da Pixar e Disney Animation.**

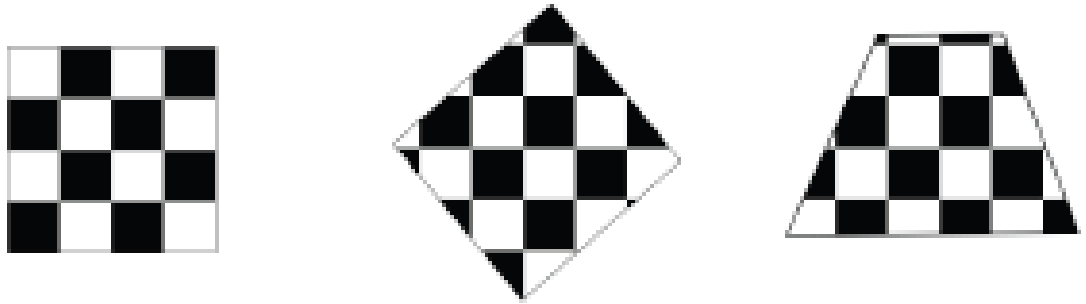
Tem feito diversas contribuições a CG.

Em 2001, recebeu um **Oscar** "for significant advancements to the field of motion picture rendering".

Em 2006, foi premiado com a IEEE von Neumann medal pelas suas contribuições na ***modelagem em CG, animação e rendering***.

Texture mapping pode ser:
Paramétrico ou não-Paramétrico:

Pode ser fixo (em tamanho ou orientação) ou se deformar com o objeto

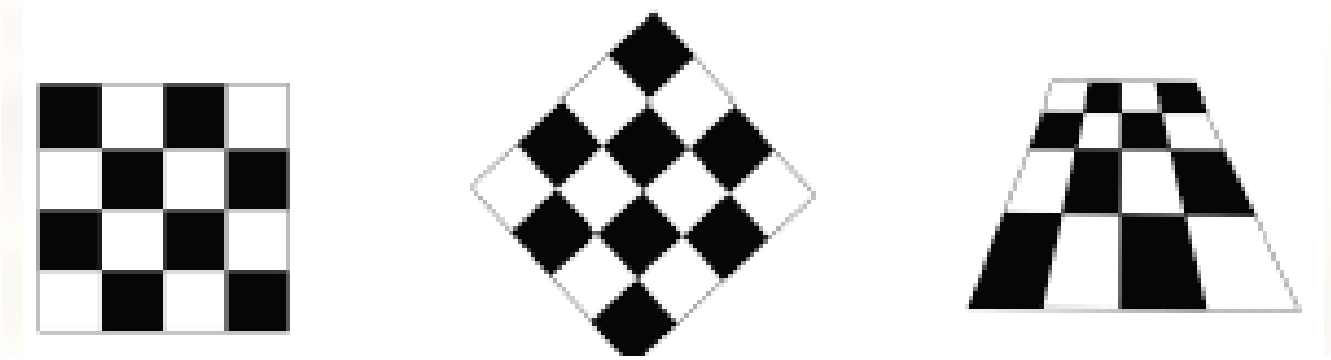


Não-Paramétrico: parece que se usou FORMA COMO CORTADOR SOBRE UM TECIDO OU MATERIAL

Paramétrico

se deforma com o objeto

Parece que A TEXTURA É do material DO OBJETO!
TEM A MESMA deformação QUE ELE NA CENA OU NA ANIMAÇÃO!

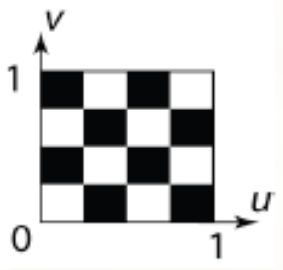


Aplicar um **texture map** é

Associar às coordenadas da face , as coordenadas do mapa de textura 2D: chamadas coordenadas UV.

ESPAÇO DE TEXTURA (u,v)

As texturas são definidas em um sistema cartesiano normalizado [0,1]x[0,1]



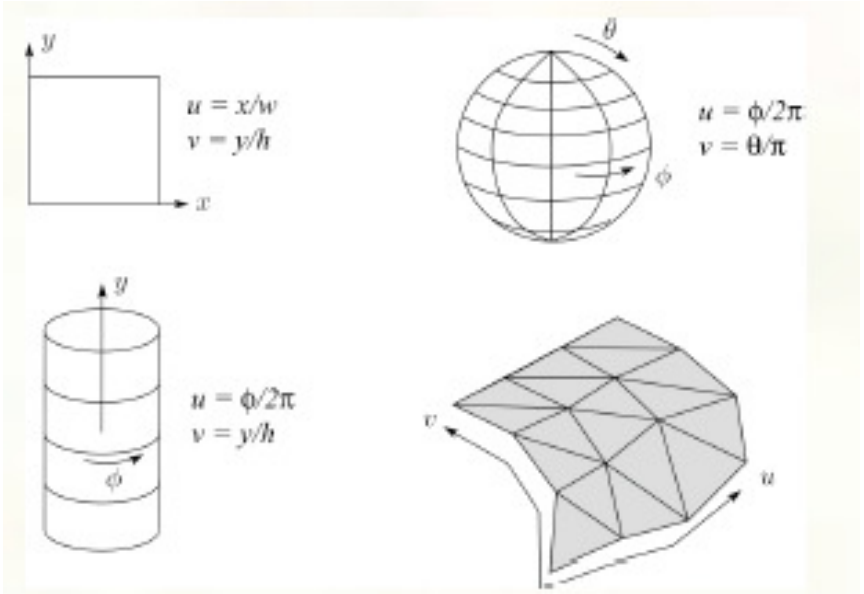
E depois usadas para

"encapar" nossos objetos,

De modo que quando

A superfície de mover

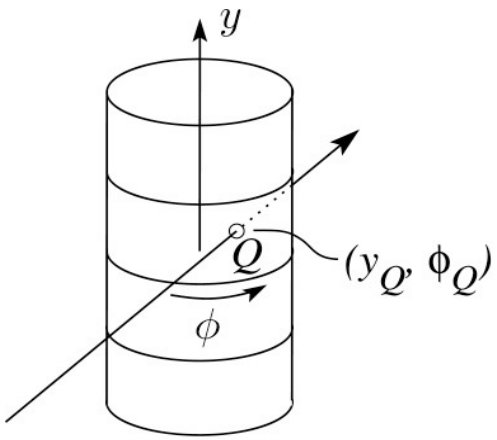
ela vá junto!



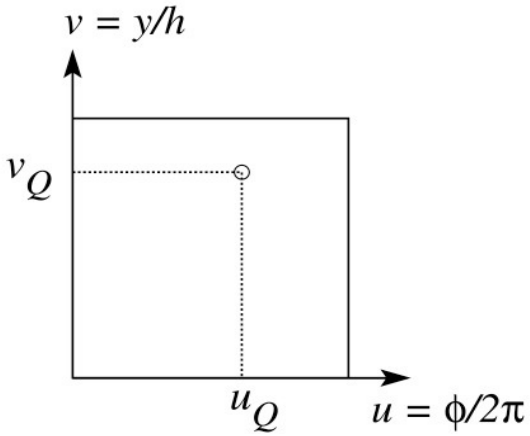
O mapa de textura é uma imagem

Essa imagem deve ser convertida para [0,1]x[0,1] e depois para as coordenadas onde será mapeada

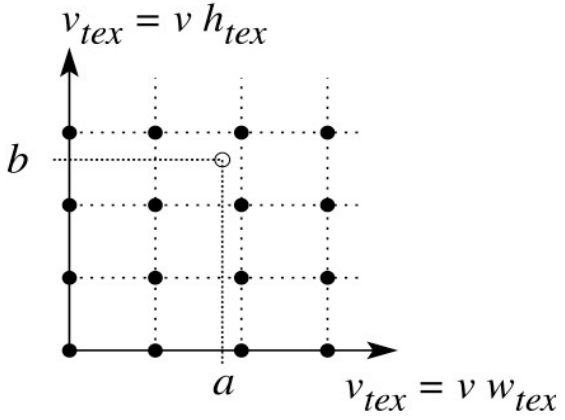
(u_{tex},v_{tex}) entre os valores: ([0.. w_{tex}], [0.. ht_{ex}])



Ray intersection

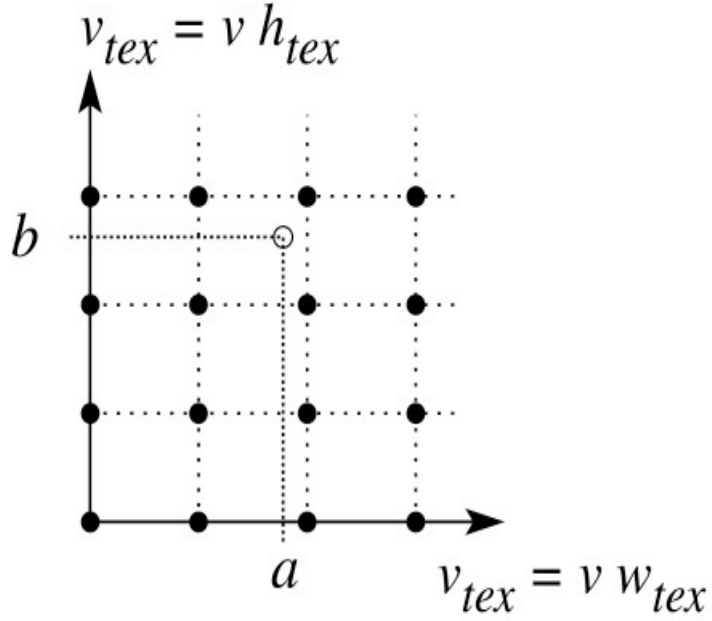


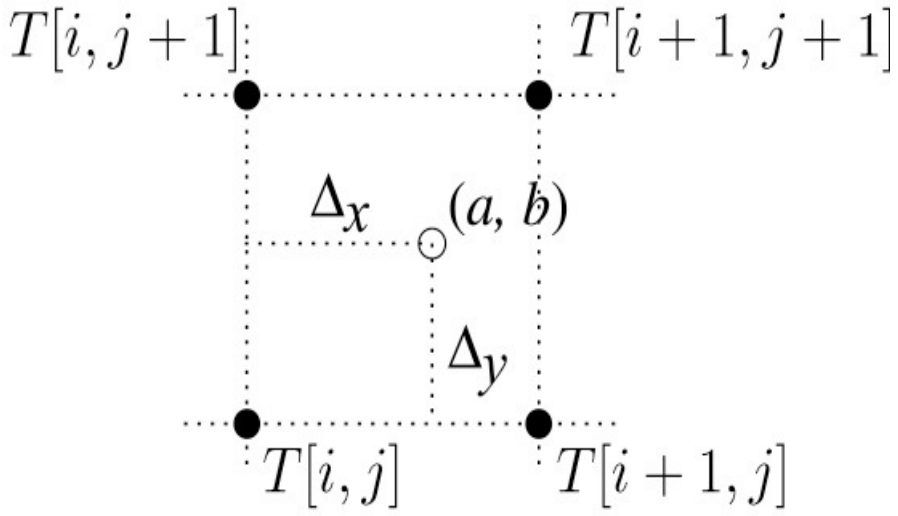
Mapping to
abstract texture coords



Mapping to
texture pixel coords

Reamostrando pelo uso de interpolação bilinear quando necessário





$T(a,b) = T[i + \Delta x, j + \Delta y]$
$= (1- \Delta x)(1- \Delta y)\ T[i, j]+ \Delta x\ (1- \Delta y)\ T[i +1, j]$
$+(1- \Delta x)\ \Delta y\ T[i, j +1]+ \Delta x\ \Delta y\ T[i +1, j +1]$

Os mapas podem fazer mais que apenas mudar os tons

Podem mudar a **geometria da superfície** em que serão mapeados, por exemplo:

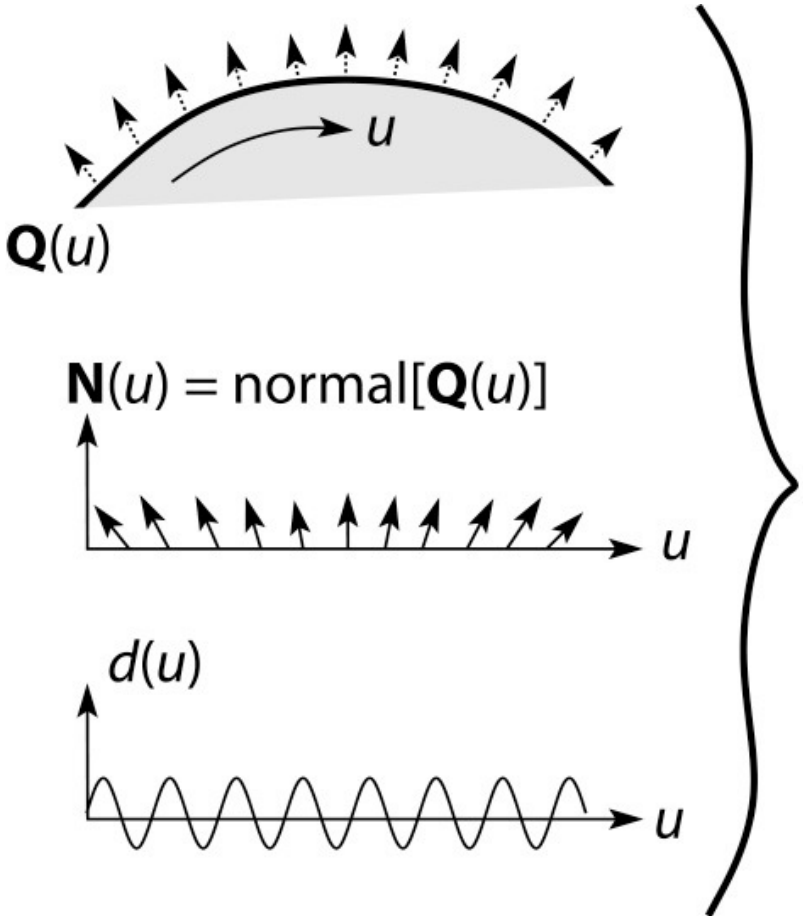
Se a superfície poder ser descrita como função de um para metro u: Q(u)

Sua normal, será geralmente também uma função: N(u)

Assim:

Displacement mapping (emboss)

Teremos assim uma nova superfície:



$$\tilde{\mathbf{Q}}(u) = \mathbf{Q}(u) + d(u)\mathbf{N}(u)$$

Bump map

Continuando com essa ideia pode-se pensar em modificar a normal da superfície (depois de fazer os tratamentos de hiddens , apenas na hora de produzir seu shading).

$$\tilde{\mathbf{N}} = \text{normal}[\tilde{\mathbf{Q}}(u)]$$

Multitexturing ocorre quando mais de um mapeamento é aplicado na face ao mesmo tempo.

Alterações na imagem final ao ser adicionada

Da fase de rendending, como:

Textura para cor difusa

textura do bump map

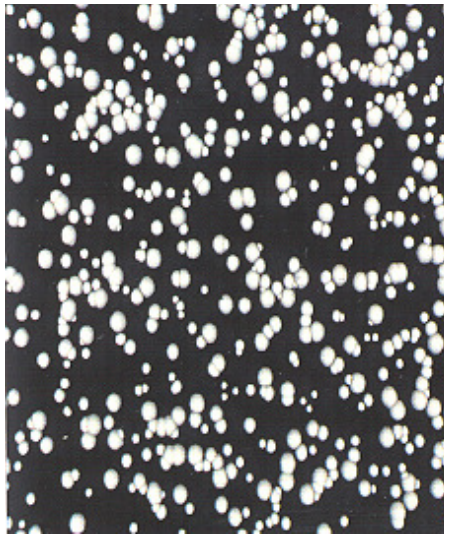
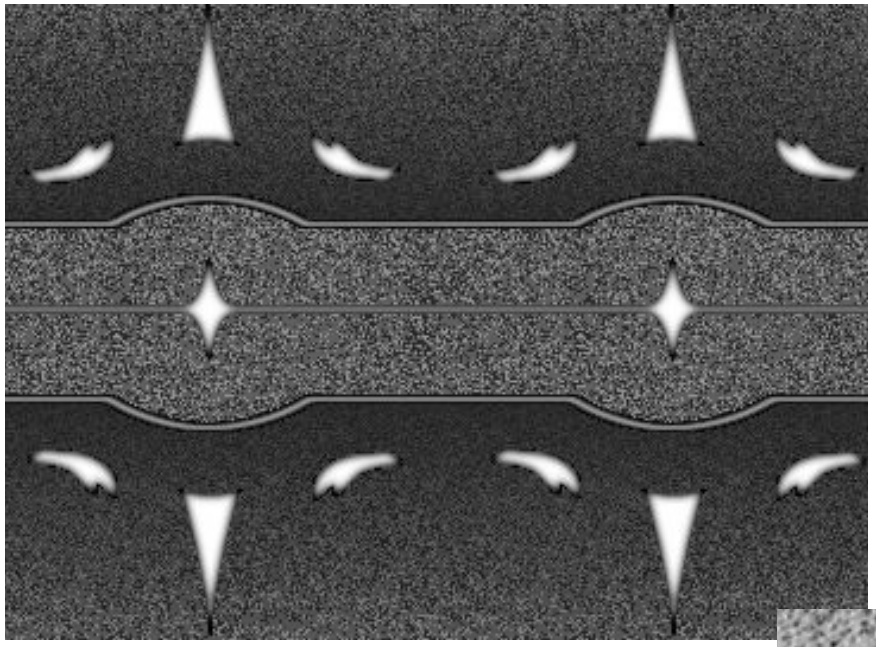


imagem final



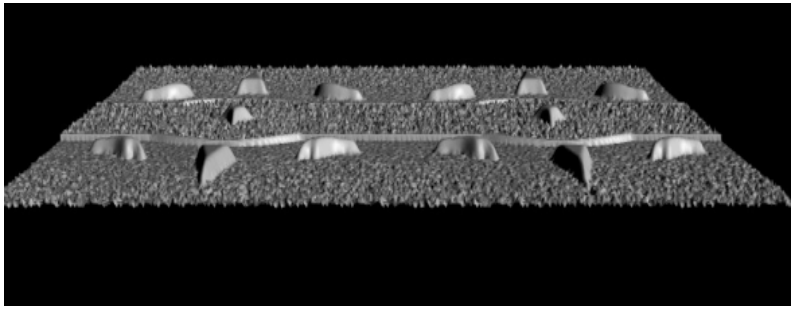
Mesma textura em fase plana

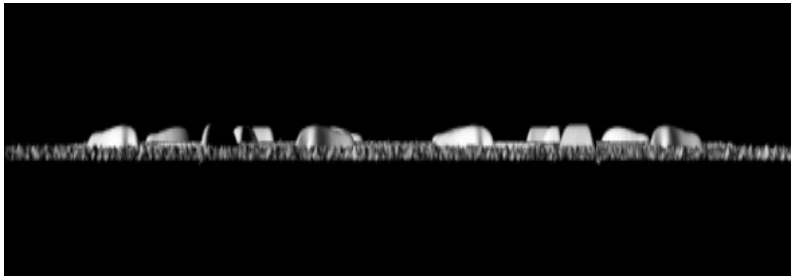
Como:

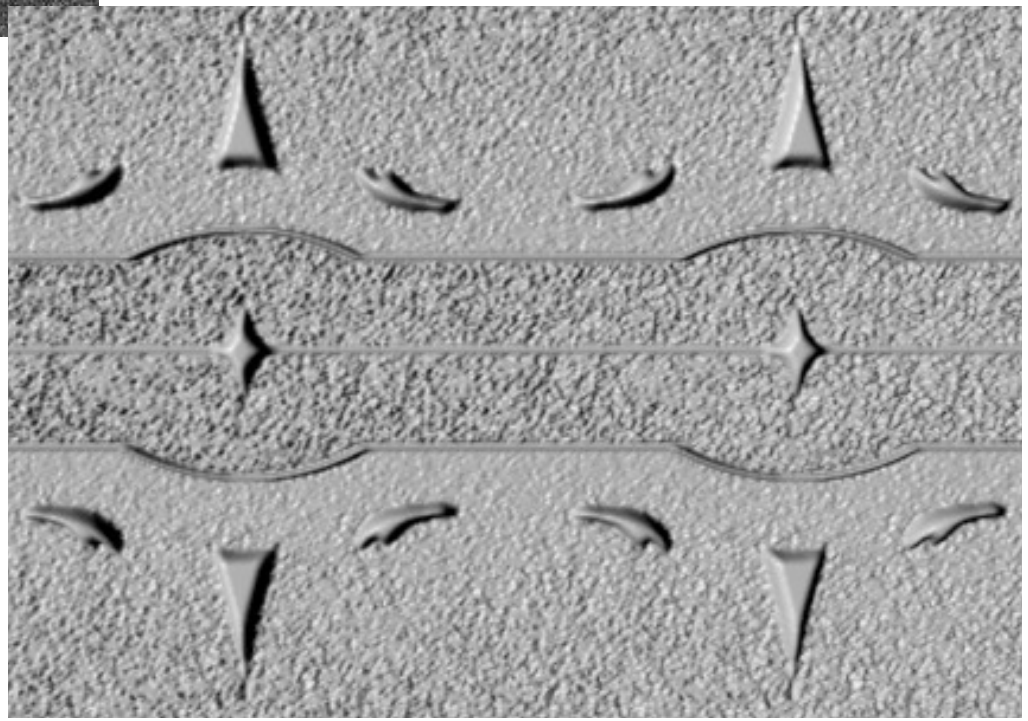
displacement map

+

bump map





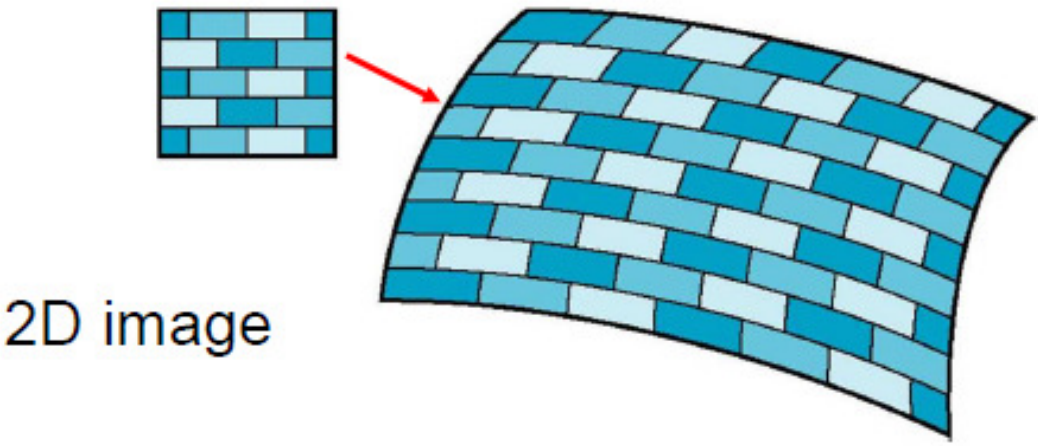


Sobre uma fase cilíndrica

Displacement map

Displacement map + bumpmap

Sistemas de coordenadas envolvidos:



$$x = x(s,t)$$
$$y = y(s,t)$$
$$z = z(s,t)$$

Forward mapping

Forwards x backwards mapping

Mapeamento direto e inverso

Dado um ponto da tela (pixel) queremos saber a que ponto do objeto ele corresponde (inverso),

e

dado um ponto do objeto, queremos saber a que ponto da textura ele corresponde (direto)

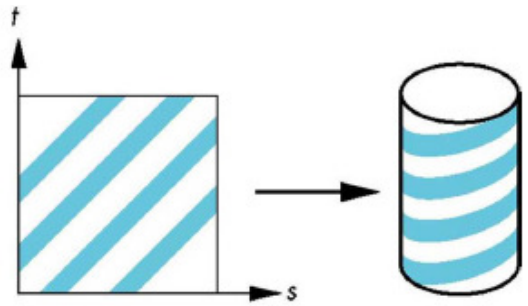
Mapeamento em 2 partes

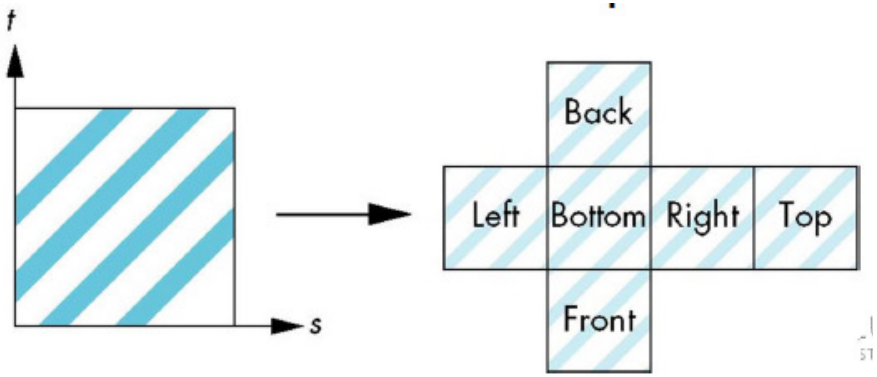
1- textura em uma forma intermediaria mais simples (cubo – projeções ortográficas, cilindro de altura h e raio r, esfera de raio r)

$$x = r \cos 2\pi u$$
$$y = r \sin 2\pi u$$
$$z = v/h$$

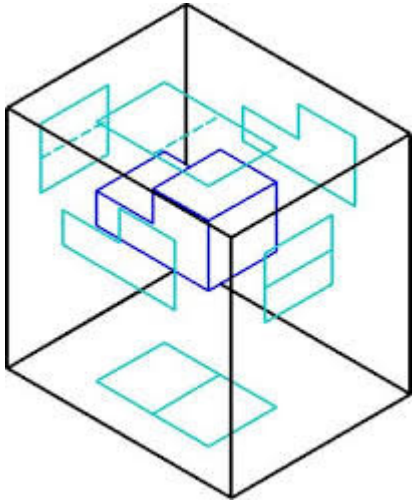
$$s = u$$
$$t = v$$

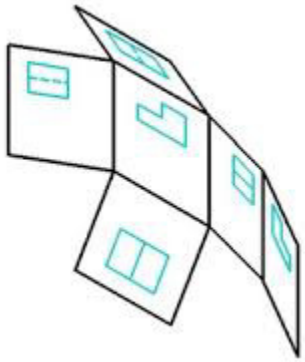
$$x = r \cos 2\pi u$$
$$y = r \sin 2\pi u \cos 2\pi v$$
$$z = r \sin 2\pi u \sin 2\pi v$$



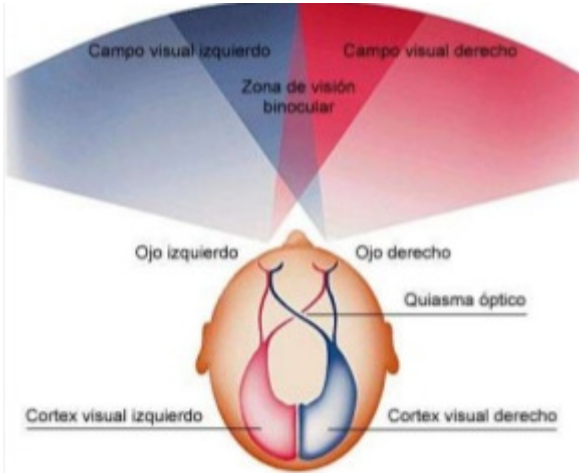


Ideia da projeção em diferentes direções



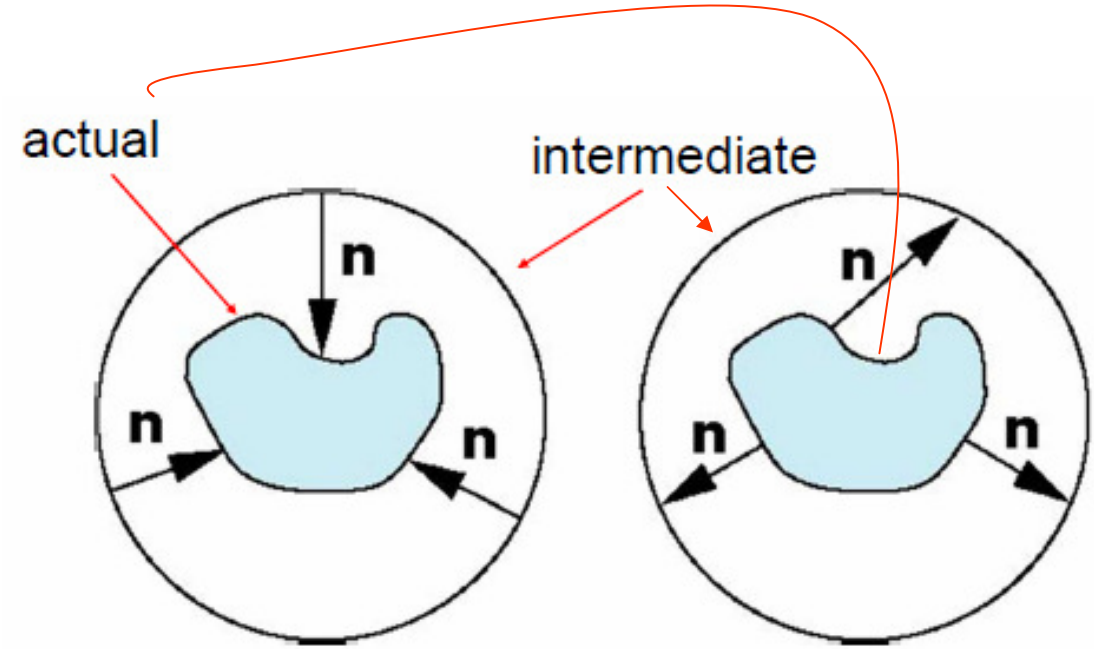


Campo visual



Mapeamento em 2 partes

2- da superfície mais simples para o objeto real, usando as normais (da intermediária para o objeto e do objeto para a intermediária)



Problemas nas texturas

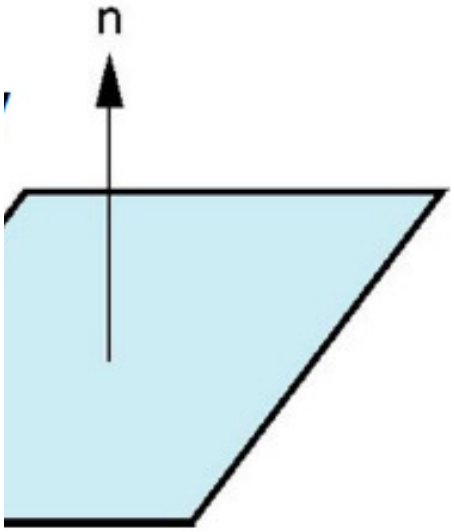
De objetos não convexos como o

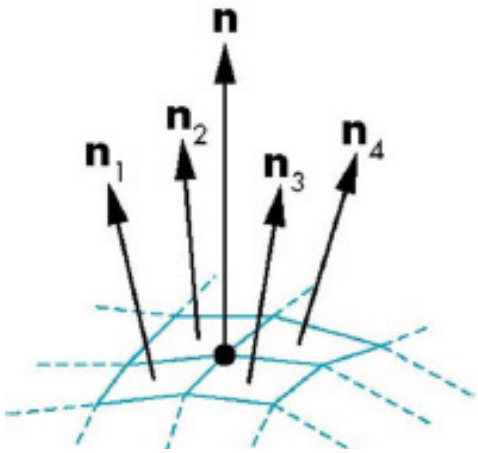
The Utah Teapot – simbolo do OpenGL

Criado por Martin Newell na University of Utah em 1975. http://www.computerhistory.org/

Tem sido usado como modelo 3D por 40 anos para verificar modelos de iluminação , cor, realismo, etc.

Normais





Superfícies planas:

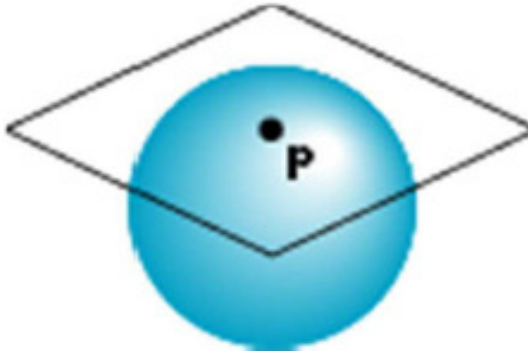
Equação do plano:

$ax+by+cz+d = 0$

Normal = (a,b,c)

Os por 3 pontos do plano: p0, p1, p2

$$\mathbf{n} = (p_2\text{-}p_0) \times (p_1\text{-}p_0)$$



Círculos: raio do ponto ao centro

Environment ou reflection map

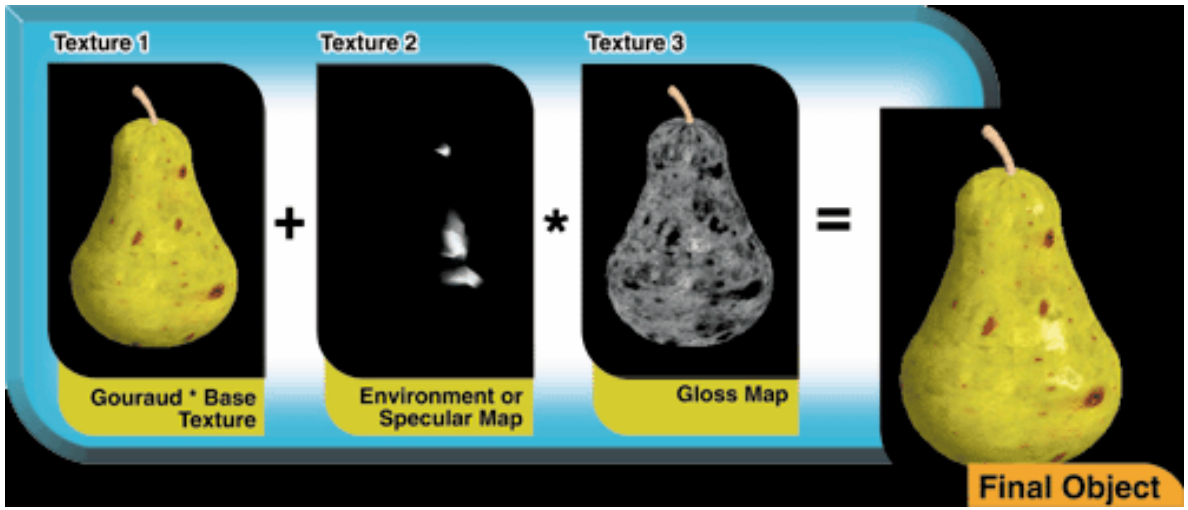
Usa para modelar o ambiente em uma superfície, como espelho.

Funciona bem com apenas um objeto na cena,

Dando uma ótima idéia de reflexão sem usar nenhum raio ou pode ser unido ao raytracing

Mapas podem ser combinados em diversos níveis

Para produzir um grau de realismo na cena de maneira simplificada



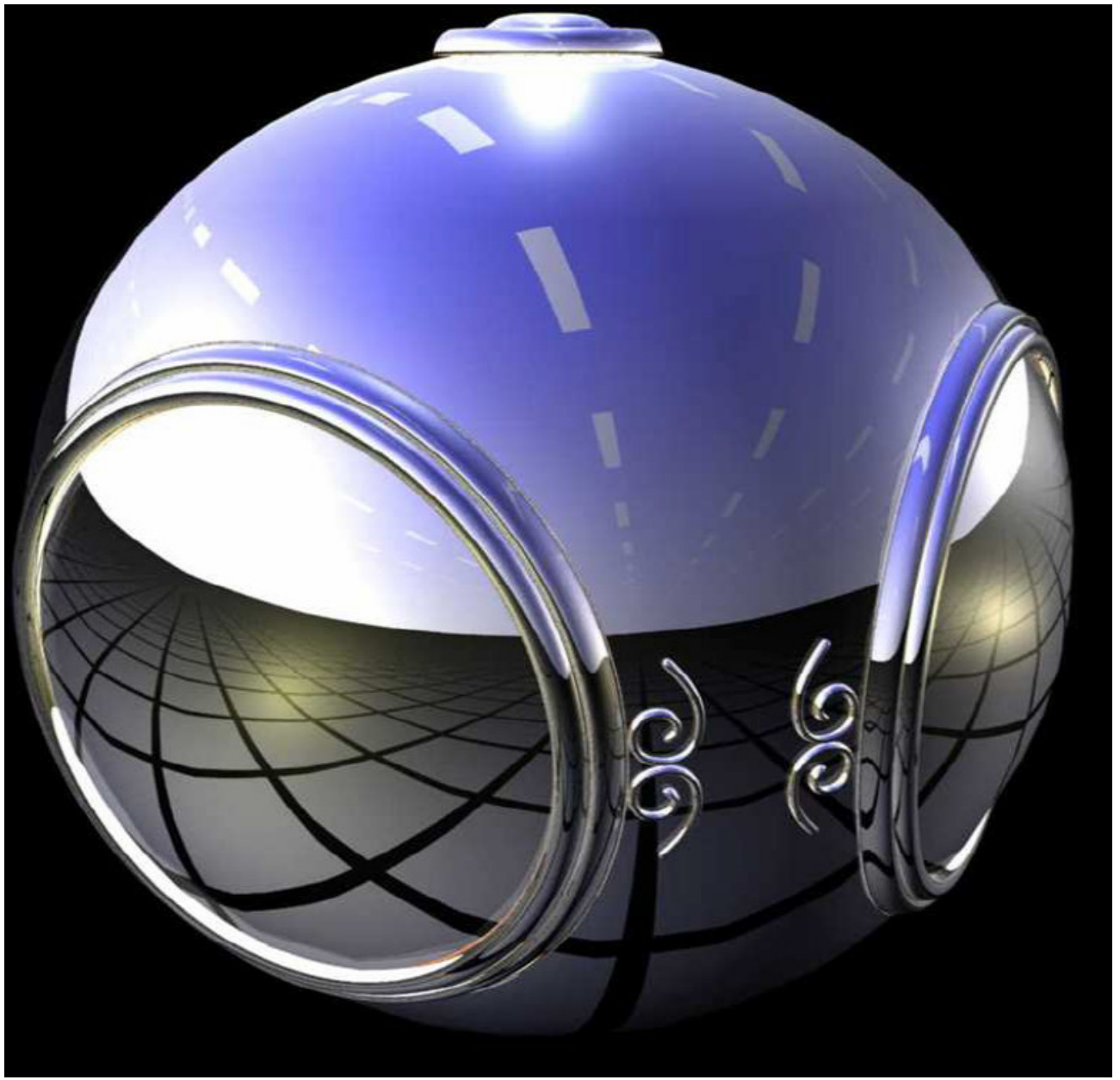
https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S_3-d_graphics_vi.pdf

Shading

Modelo Phong

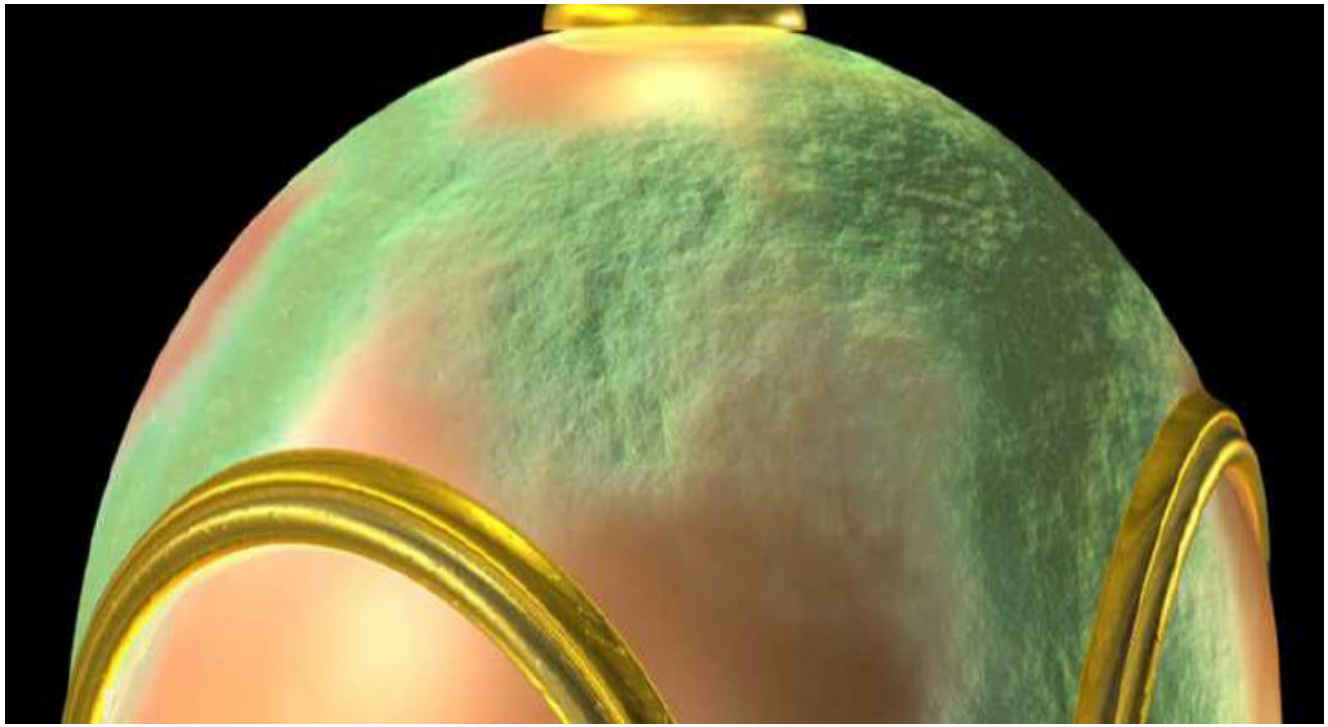
Texture map

Environment Mapping



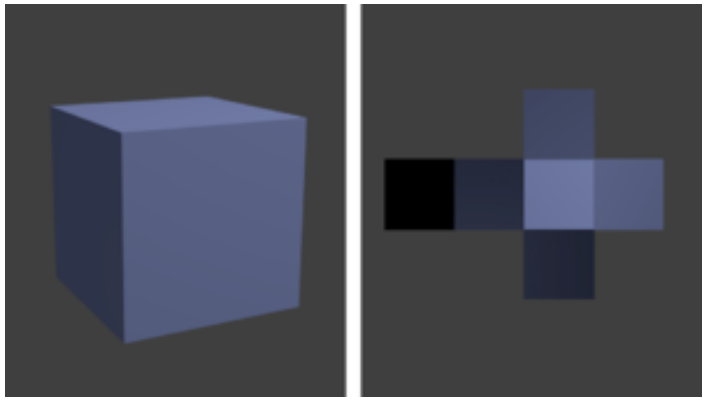
detalhes

Bump Mapping



Lightmap

Mapeamento que contém a intensidade luminosa das faces. Útil em objetos que permanecem estáticos em games. Geralmente flat , sem incluir a idéia da direção da iluminação . Presentes na maioria dos plug-ins 3D



Procedural texture generation method

As texturas podem ser geradas por programação (procedures) e não apenas por captura de texturas já existentes

Geradores de padrões fractais são muitos úteis para isso!

Level of details (mip maps)

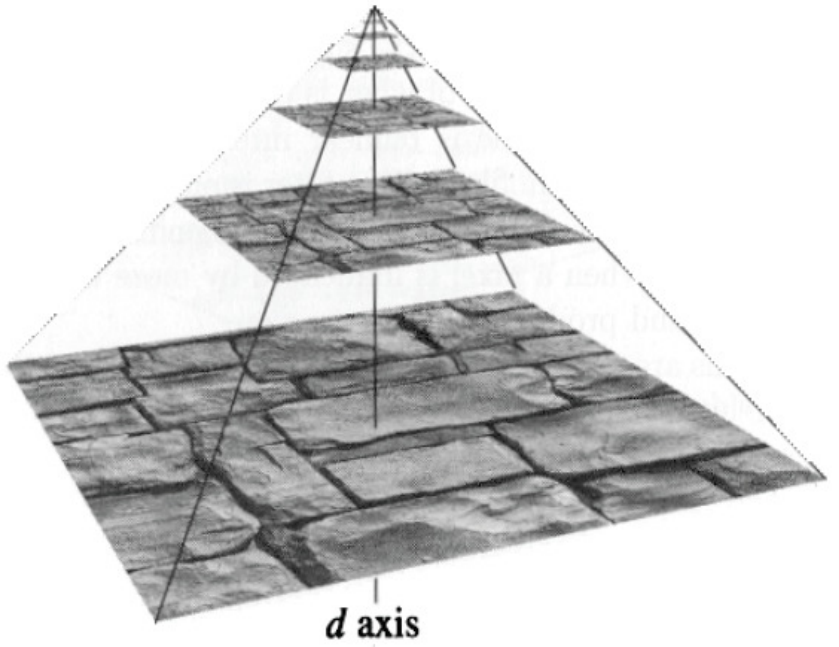
Alterado detalhes da textura com a distância ao observador

Também pode ser simulado com filtros

Que diminuem a resolução

"MIP" acronym of the phrase *multum in parvo* =

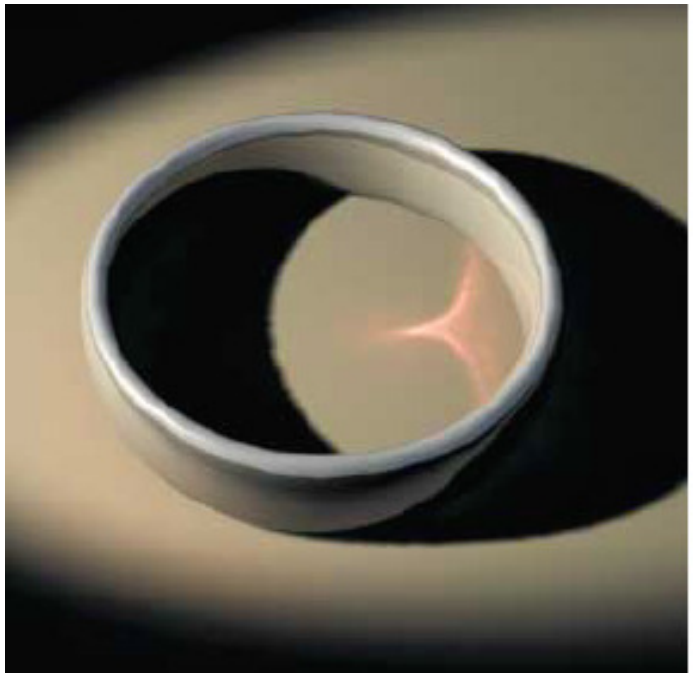
"much in little"

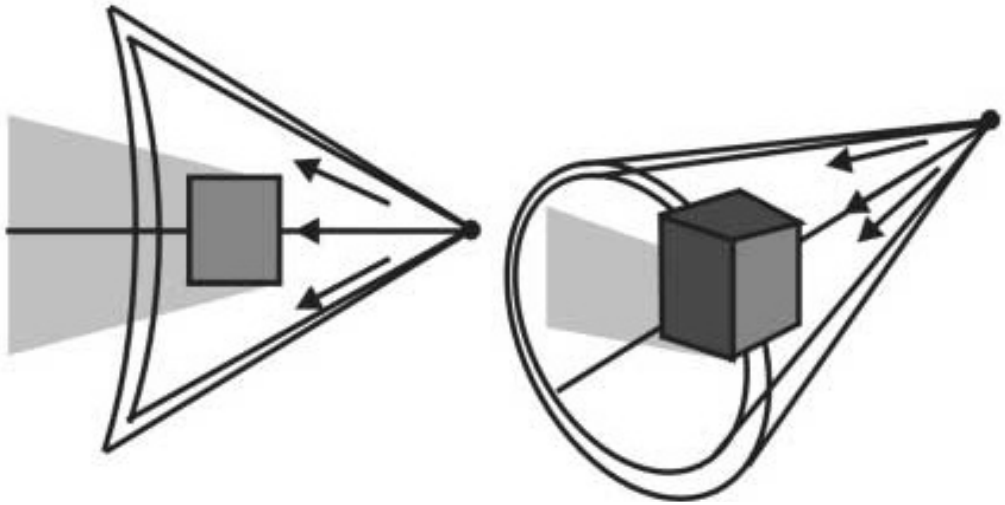


Sombras, refração, reflexão e efeitos específicos

Sombras,

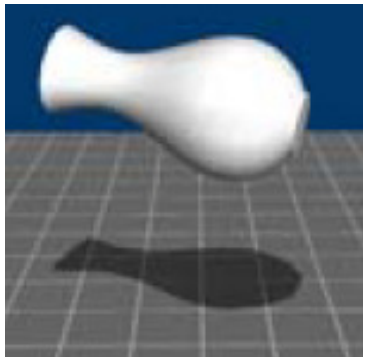
podem ser consideradas por diversos métodos, de simples projeções , passando por texturas até os métodos globais (seção 7.3.6 do livro texto tem boa revisão do assunto) !





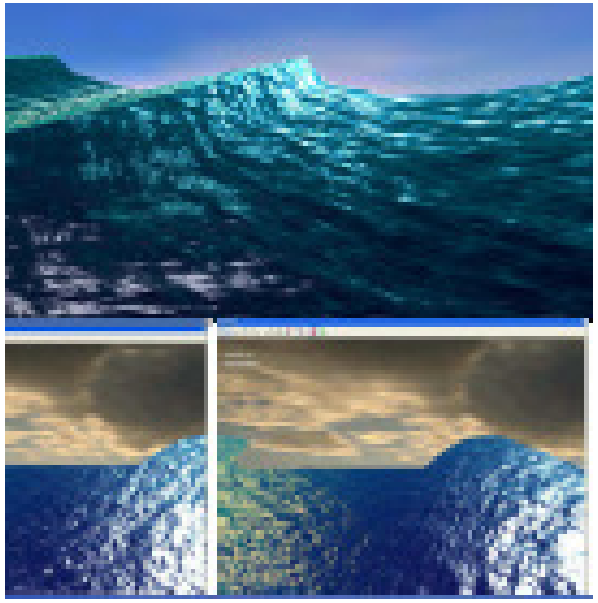
Sombras planas e projetadas:

Sombra=umbra e penumbra



Cautics

São padrões de luz (refletidas e refratadas) que parecem concentrar a luz em alguns pontos. Ocorrem em vidros, água, modelagens de ondas, piscinas e outras situações que concentramos raios luminosos



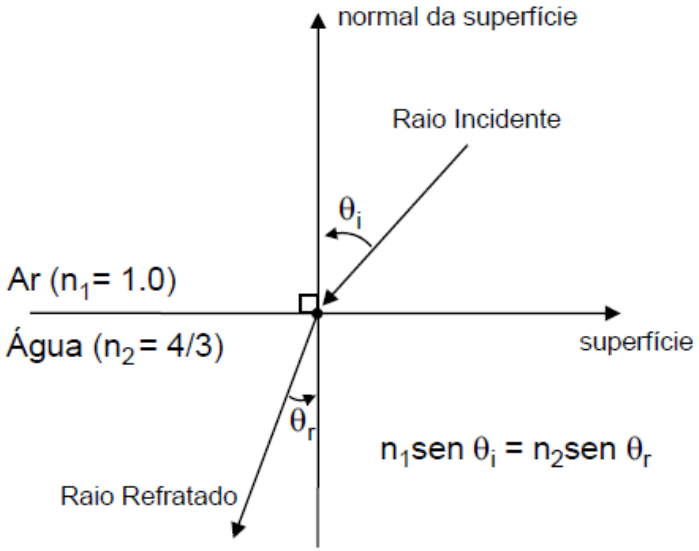
Refração

Quando o feixe de luz penetra em alguns materiais sua trajetória muda de ângulo de acordo com a diferença de densidade dos meios.

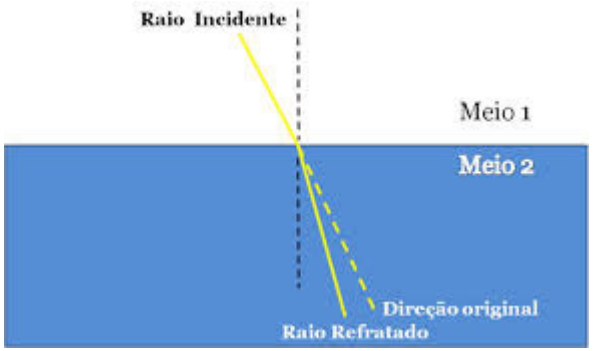
Lei da refração ou de Snell:

$$\frac{\text{sen}\,\theta_i}{\text{sen}\,\theta_r} = n_{21}$$

n_{21} é uma constante, chamada índice de refração ou IR



Exemplo de alguns IR:



Material	IR
Ar (em temperatura e pressão padrão ou STP)	1,0003
Água	1,33
Álcool etílico	1,36
Vidro	1,66
Plástico	1,51
Vidro Denso	1,52
Sal	1,53
Quartzo	1,46
Cristal	1,58
Diamante	2,42

Transparência

$$I = t\, I_1 + (1\text{-}t)\, I_2,\ 0 \leq t \leq 1$$

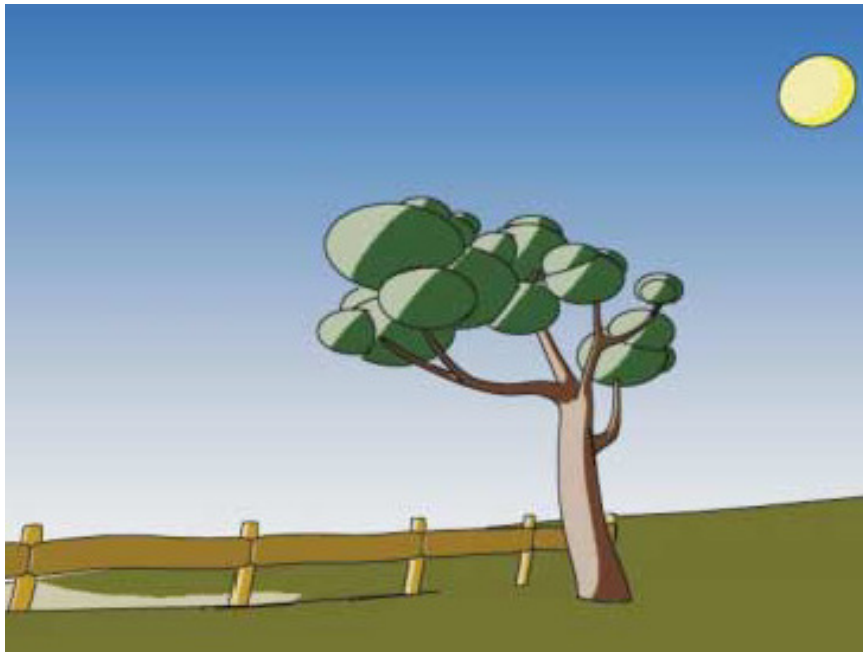
onde, I_1 é a superfície visível, I_2 é a superfície imediatamente atrás da superfície visível, e t é o fator de transparência para I_1. Se I_2 também é transparente, o algoritmo é aplicado recursivamente até encontrar uma superfície opaca ou o fundo da cena.

Há muito mais do que isso!

Vimos aqui apenas sobre um realismo fotográfico das imagens, mas há diversas outras formas e esse assunto esta sempre em constante evolução. Assim depois desta leve introdução continue na área! Você já tem a bagagem teórica que precisa para agora descobrir o resto sozinho!

Toon Shading

Stylistic rendering



Bibliografia:

D. F. Rogers, J. A. Adams. Mathematical Elements for Computer Graphics, 2dn Ed. , Mc Graw Hill, 1990

E. Azevedo, A. Conci, Computação Gráfica: teoria e prática, Campus ; - Rio de Janeiro, 2003

J.D.Foley,A.van Dam,S.K.Feiner,J.F.Hughes. Computer Graphics- Principles and Practice, Addison-Wesley, Reading, 1990.

A. H. Watt, F. Policarpo - The Computer Image , Addison-Wesley Pub Co (Net); 1998

https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S_3-d_graphics_vi.pdf

http://graphics.stanford.edu/papers/rad/

Bibliografia:

D. F. Rogers, J. A. Adams. Mathematical Elements for Computer Graphics, 2dn Ed. , Mc Graw Hill, 1990

E. Azevedo, A. Conci, Computação Gráfica: teoria e prática, Campus ; - Rio de Janeiro, 2003

J.D.Foley,A.van Dam,S.K.Feiner,J.F.Hughes. Computer Graphics- Principles and Practice, Addison-Wesley, Reading, 1990.

Y. Gardan. Numerical Methods for CAD , MIT press, Cambridge, 1985.

A. H. Watt, F. Policarpo - The Computer Image , Addison-Wesley Pub Co (Net); 1998

https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S_3-d_graphics_vi.pdf

http://graphics.stanford.edu/papers/rad/