

Pipeline Gráfico 3D

Produção de Imagens

- Batch
 - Cada imagem é gerada com a maior qualidade possível
 - o tempo de processamento não é uma prioridade
 - uso de técnicas orientadas ao foto-realismo
 - ex: filmes animados, efeitos especiais (TV, cinema)
- Interativa
 - Cada imagem deverá ser gerada num curto espaço de tempo (milisegundos)
 - input, animação, movimentação da câmara e objetos
 - aplicação de técnicas mais simples (polígonos)
 - ex: jogos, visualização interativa

Pipeline Gráfico 3D

Primitivas Gráficas 3D



Input:

- Modelo geométrico
 - descrição de todos os objetos, superfícies, fontes de luz e transformações
- Modelo de iluminação
 - descrição computacional das propriedades dos objetos, da luz e suas interações
- Especificação da ponto de vista
 - Posição de observação e volume de visão
- Especificação do visor
 - Grelha de pixels onde o plano da imagem será mapeado

Output:

- Cores - valores RGB 24 bits, p.ex.

Pipeline Gráfico 3D

Primitivas Gráficas 3D



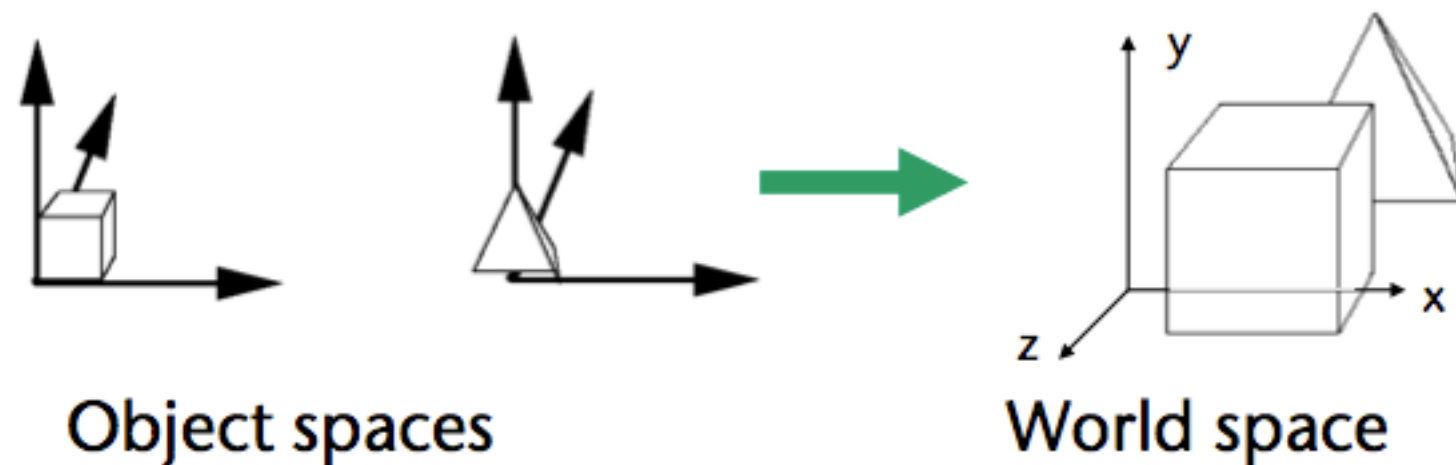
- As primitivas são processadas numa série de estágios ou fases
- Os resultados duma fase alimentam a fase seguinte
- A implementação concreta do pipeline admite variações
- Cada fase poderá estar implementada no hardware ou em software
- Algumas fases admitem a possibilidade de programação ou parametrização

Transformação de Modelação (Modeling Transformation)

Primitivas Gráficas 3D



- Objetos primitivos, modelados em referenciais próprios (object space)
- Transformações de modelação escalam, orientam e posicionam esses objetos num referencial comum do mundo (world space)



Primitivas Gráficas 3D



Exemplo OpenGL

```
// Setup da matriz model(view)
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

// transf. de modelação
glTranslated(...);
glRotated(...);
...
glScaled();

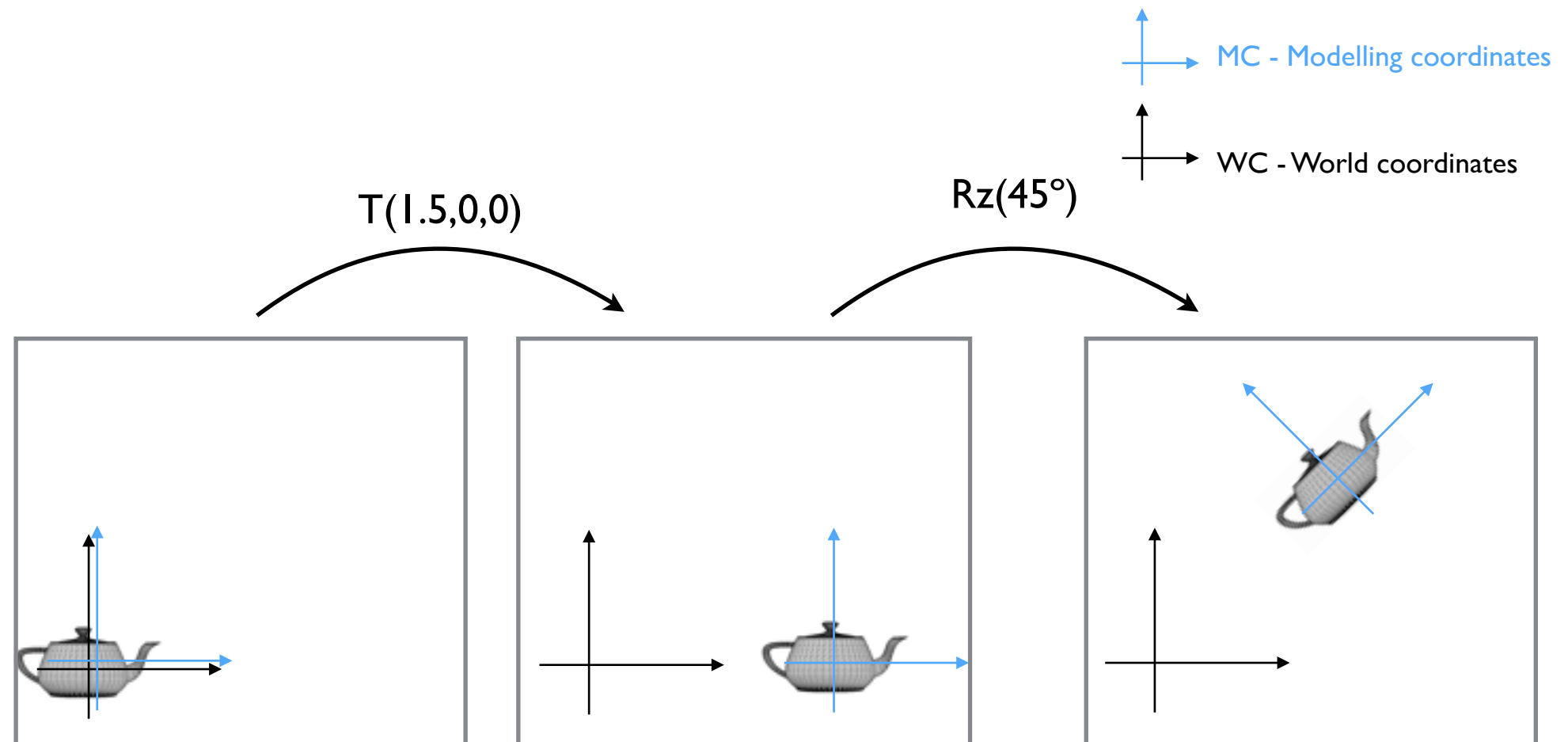
// primitiva
glBegin(...);
glVertex3d(...);
...
glEnd();
```

Transformação de Modelação (Modeling Transformation)

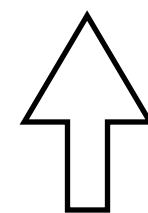
Primitivas Gráficas 3D



Ordem das Transformações



No referencial WC, a ordem de leitura das transformações é de baixo para cima



```
glRotated(45, 0, 0, 1);  
glTranslated(1.5, 0, 0);
```

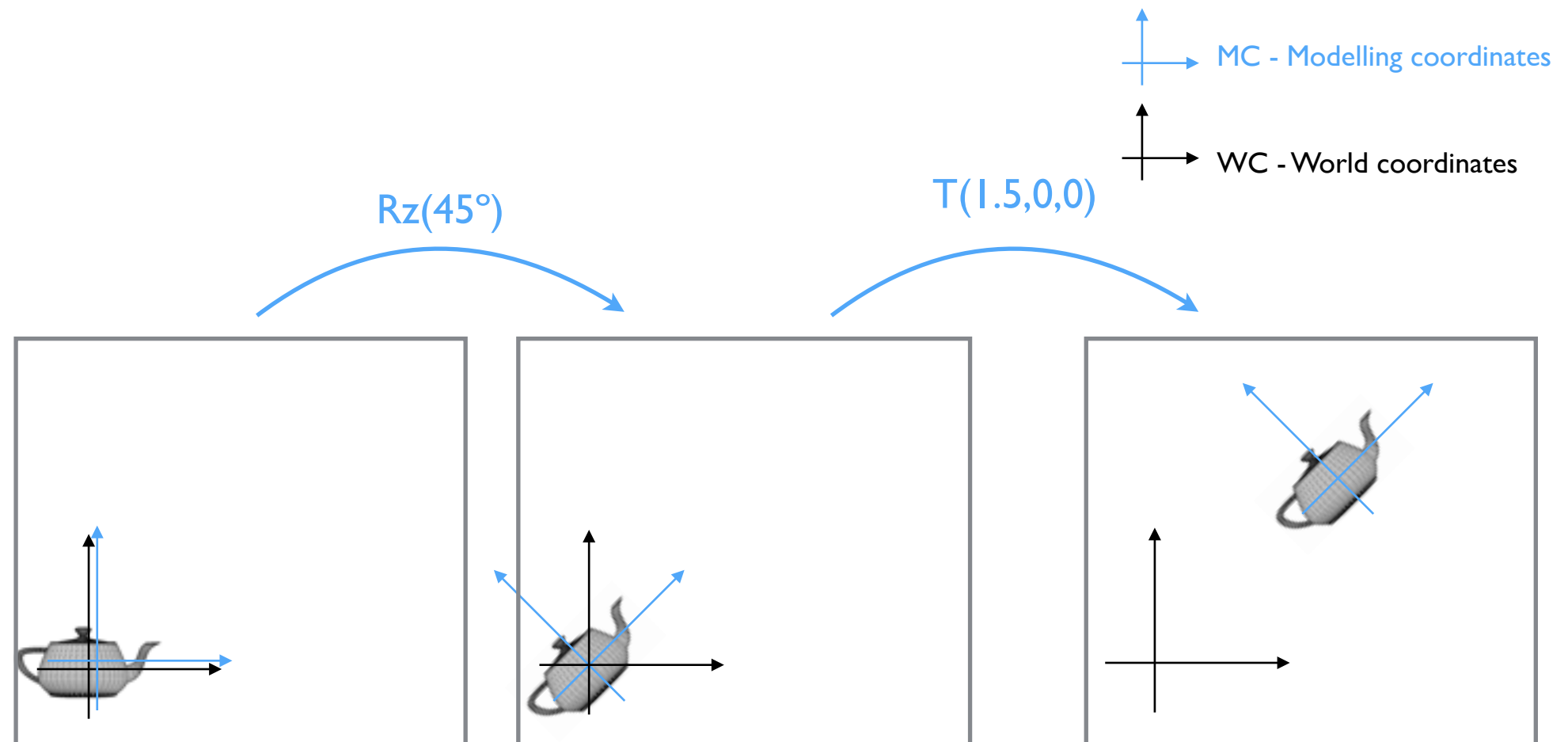
Na primeira figura, um dos referenciais (coincidentes) foram representados com defazamento para se poderem visualizar ambos

Transformação de Modelação (Modeling Transformation)

Primitivas Gráficas 3D



Ordem das Transformações



No referencial MC, a ordem de leitura das transformações é de cima para baixo



```
glRotated(45, 0, 0, 1);  
glTranslated(1.5, 0, 0);
```

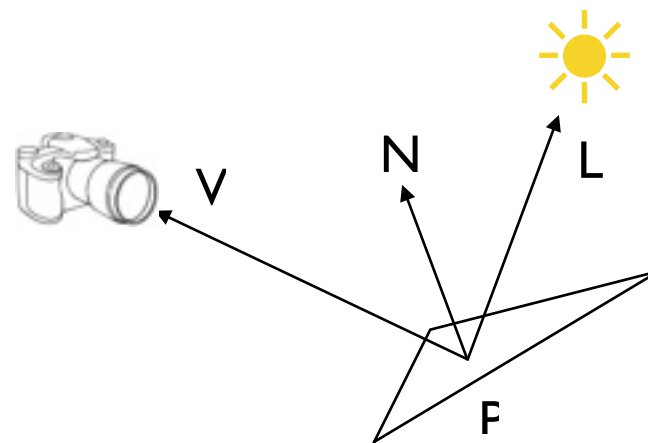
Na primeira figura, um dos referenciais (coincidentes) foram representados com defazamento para se poderem visualizar ambos

Iluminação (Lighting)

Primitivas Gráficas 3D



- Os vértices são iluminados (coloridos) de acordo com as propriedades do material, da superfície (normal) e fontes de luz
- Aplicado um modelo de iluminação local* (direta) que determina a luz refletida pelo objeto na direção de observação.



Legenda:

- L vetor apontando na direção da fonte de luz
- N vetor normal à superfície no ponto P
- V vetor apontando na direção do observador

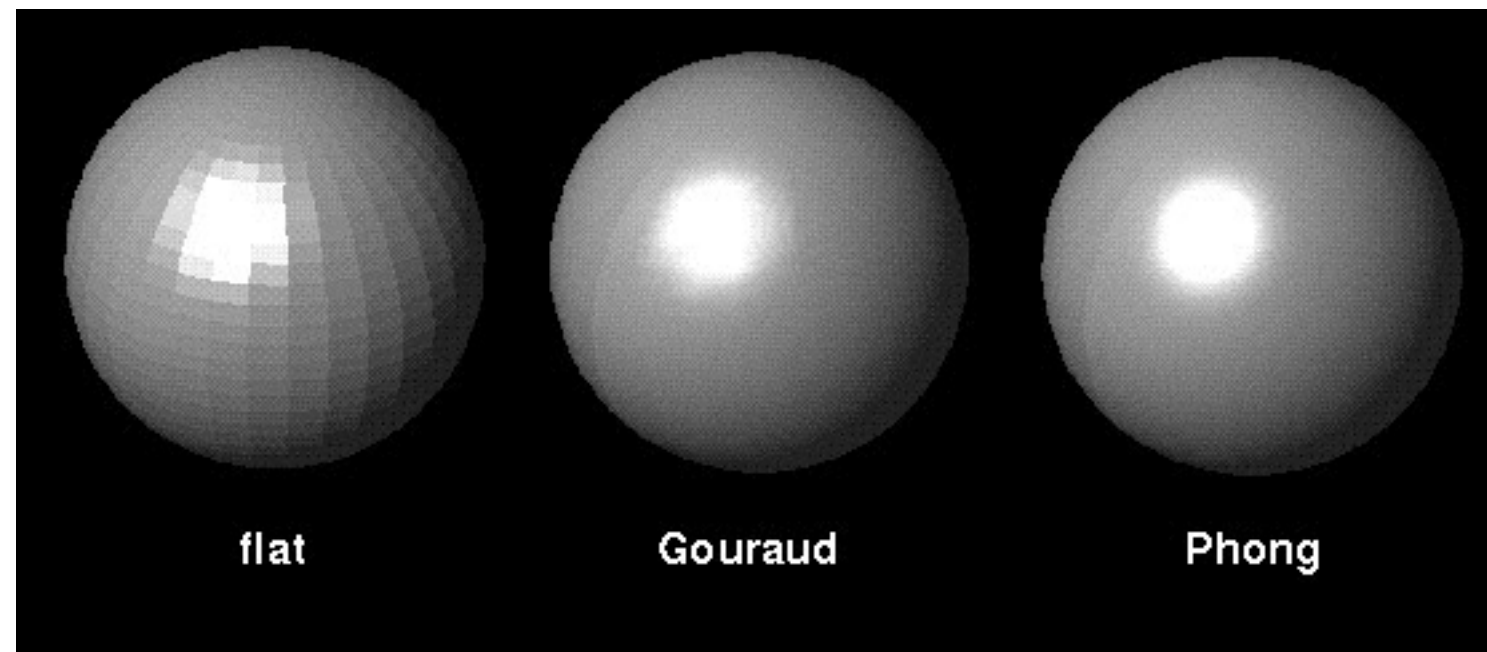
- Em OpenGL, é possível usar múltiplas luzes, as quais podem ser ligadas/desligadas de forma independente
- A posição/direção duma luz é tratada como os vértices de primitivas, sendo afetada pela matriz MODELVIEW.

* num modelo de iluminação global (indireta), são levadas em conta as múltiplas reflexões entre objetos.

Primitivas Gráficas 3D



- Os cálculos de iluminação podem ser efetuados:
 - uma vez por face (flat)
 - uma vez por vértice (Gouraud)
 - uma vez por pixel (Phong)



Transformação de Visualização (Viewing Transformation)

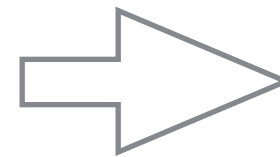
Primitivas Gráficas 3D



- Transformação de WC para Eye Coordinates (referencial do observador ou da câmara)
- O observador (e os objectos) são deslocados e orientados por forma a observar a cena segundo uma direção pré-estabelecida (normalmente o eixo dos z's)



World space



Eye space

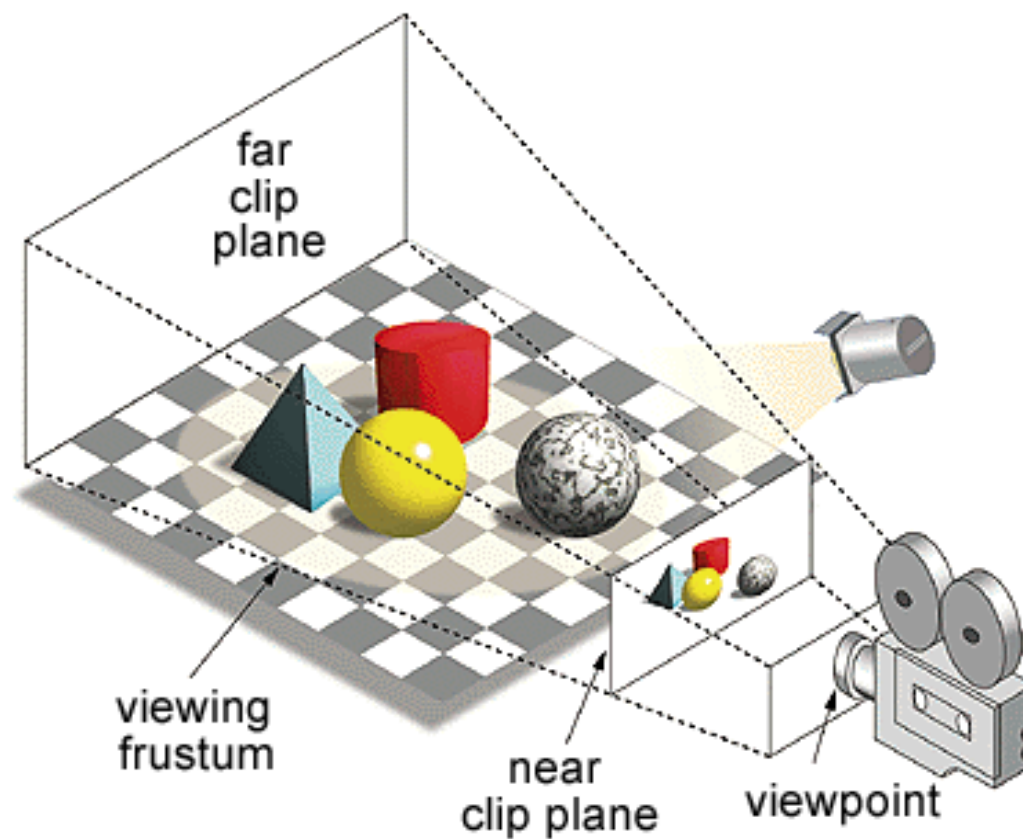
- Duas formas de encarar a transformação:
 - Transforma pontos do referencial do mundo para o referencial do observador
 - Transforma o referencial do observador no referencial do mundo

Transformação de Projeção (Projection Transformation)

Primitivas Gráficas 3D



- Já no referencial do observador, a projeção é simplesmente efetuada usando uma matriz de projeção perspectiva ou ortogonal (M_{per} ou M_{ort})
- Mas o plano de projeção é infinito! O que pretendemos ver na imagem final?
 - ➔ É necessário especificar a janela 2D do enquadramento...



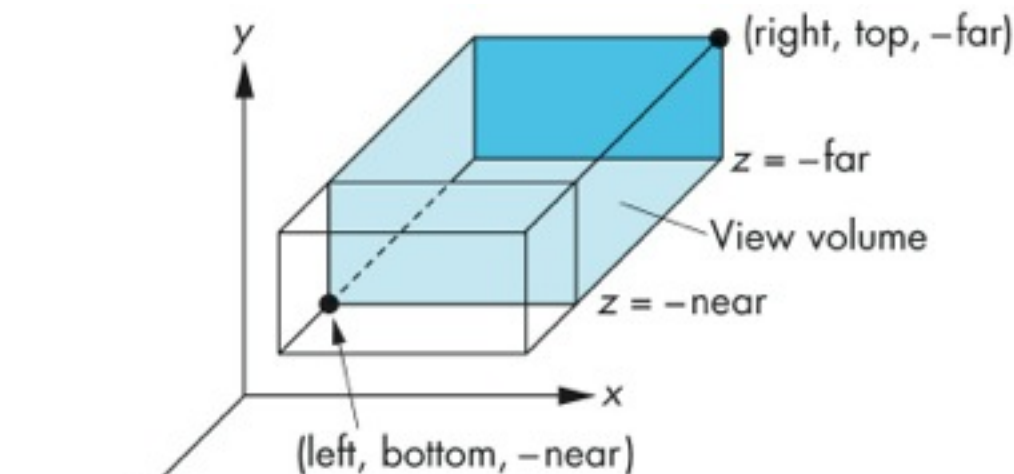
<http://encyclopedia2.thefreedictionary.com/3D+graphics>

Primitivas Gráficas 3D



OpenGL - Projeção Ortogonal

`glOrtho(left, right, bottom, top, near, far)`



*void **glOrtho**(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);*
Creates a matrix for an orthographic parallel viewing volume and multiplies the current matrix by it. (left, bottom, -near) and (right, top, -near) are points on the near clipping plane that are mapped to the lower-left and upper-right corners of the viewport window, respectively. (left, bottom, -far) and (right, top, -far) are points on the far clipping plane that are mapped to the same respective corners of the viewport. Both near and far can be positive or negative.

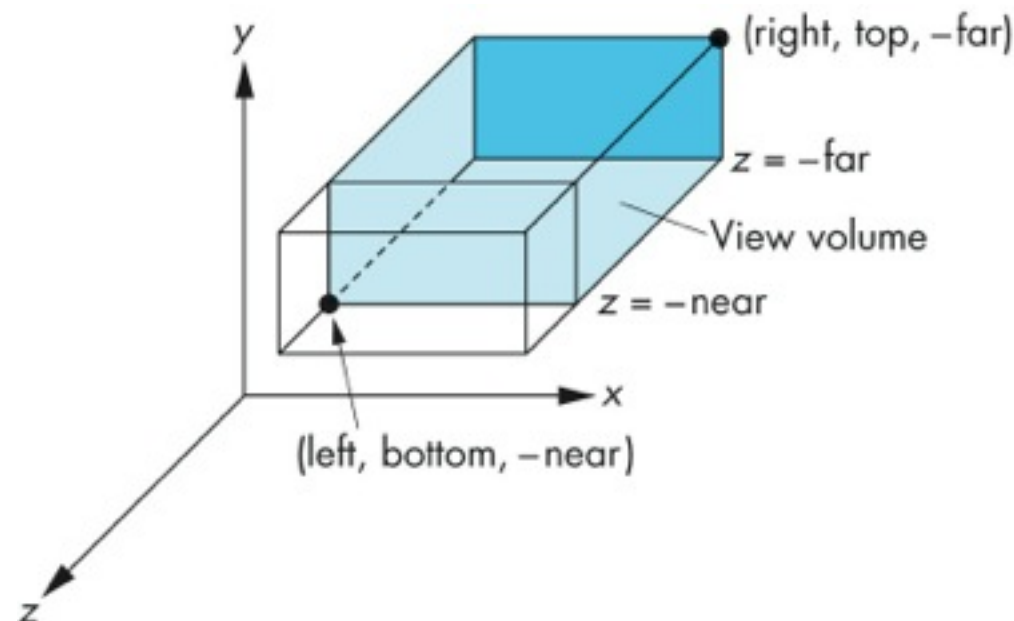
O volume especificado é transformado num volume canónico $[-1,1] \times [-1,1] \times [-1,1]$ através da transformação deduzida a partir dos parâmetros passados em `glOrtho()`

Primitivas Gráficas 3D



OpenGL - Projeção Ortogonal

`glOrtho(left, right, bottom, top, near, far)`



Qual a matriz M, que transforma o volume de visão no volume canónico?

$$M = S(2/dx, 2/dy, 2/dz) T(-cx, -cy, -cz)$$

(cx, cy, cz) - ponto médio do volume

$dx \times dy \times dz$ - dimensões do volume

$dx = \text{right} - \text{left}$

$dy = \text{top} - \text{bottom}$

$dz = \text{near} - \text{far}$

$cx = (\text{right} + \text{left}) / dx$

$cy = (\text{top} + \text{bottom}) / dy$

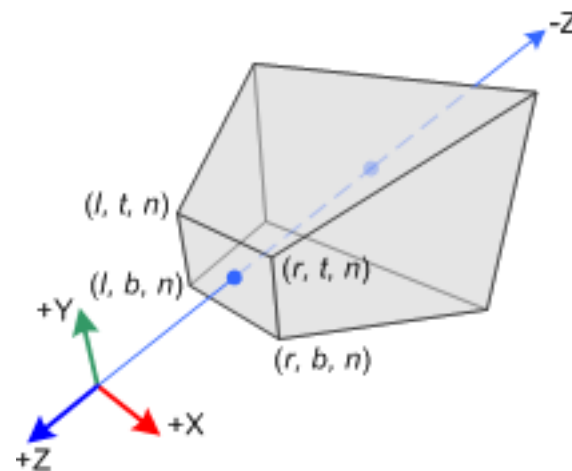
$cz = (\text{far} + \text{near}) / dz$

Primitivas Gráficas 3D



OpenGL - Projeção Perspetiva

`glFrustum(left, right, bottom, top, near, far)`



`void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);` Creates a matrix for a perspective-view frustum and multiplies the current matrix by it. The frustum's viewing volume is defined by the parameters: (left, bottom, -near) and (right, top, -near) specify the (x, y, z) coordinates of the lower-left and upper-right corners of the near clipping plane; near and far give the distances from the viewpoint to the near and far clipping planes. They should always be positive.

Alternativa:

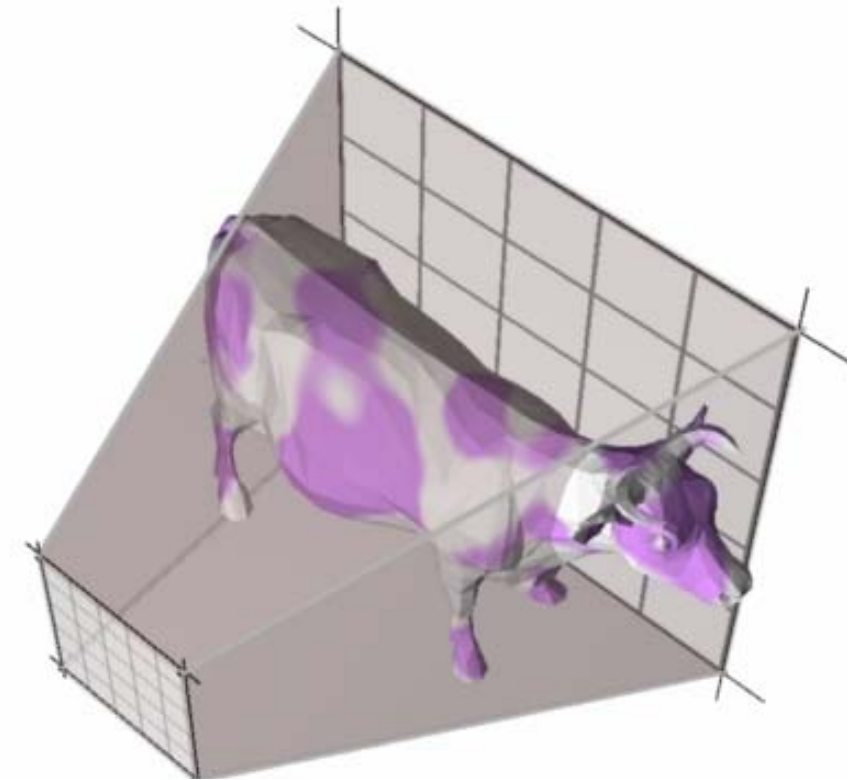
```
glOrtho(left, right, bottom, top, near, far);  
glLoadMatrix(mper);  
// mper é a projeção perspetiva com plano projeção z=near  
// mas mantendo a coordenada z!
```

Primitivas Gráficas 3D



Motivação?

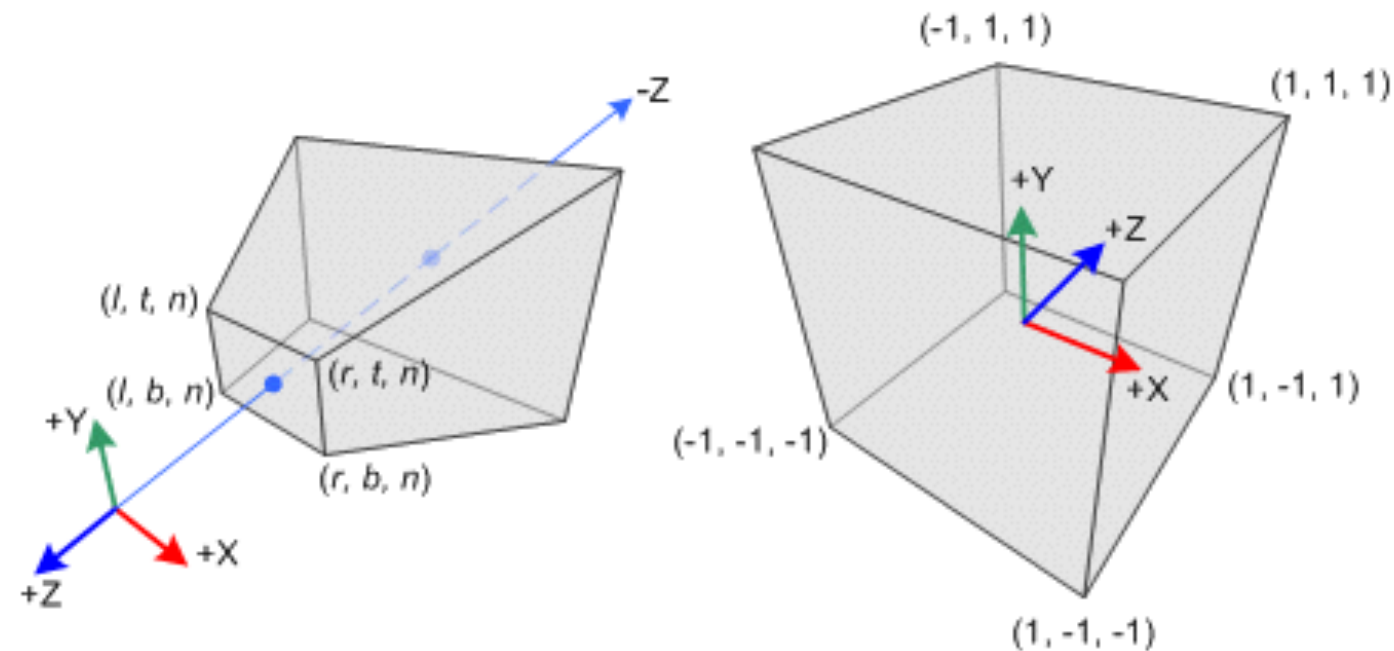
- Eficiência
 - Evitar desperdiçar tempo com o processamento de objetos fora do volume de visão
- Evitar situações degeneradas:
 - divisão por 0
 - projeção de objetos atrás do observador



taken from: MIT course number 6.837 (2003)

Recorte (Clipping)

Primitivas Gráficas 3D



http://www.songho.ca/opengl/gl_projectionmatrix.html

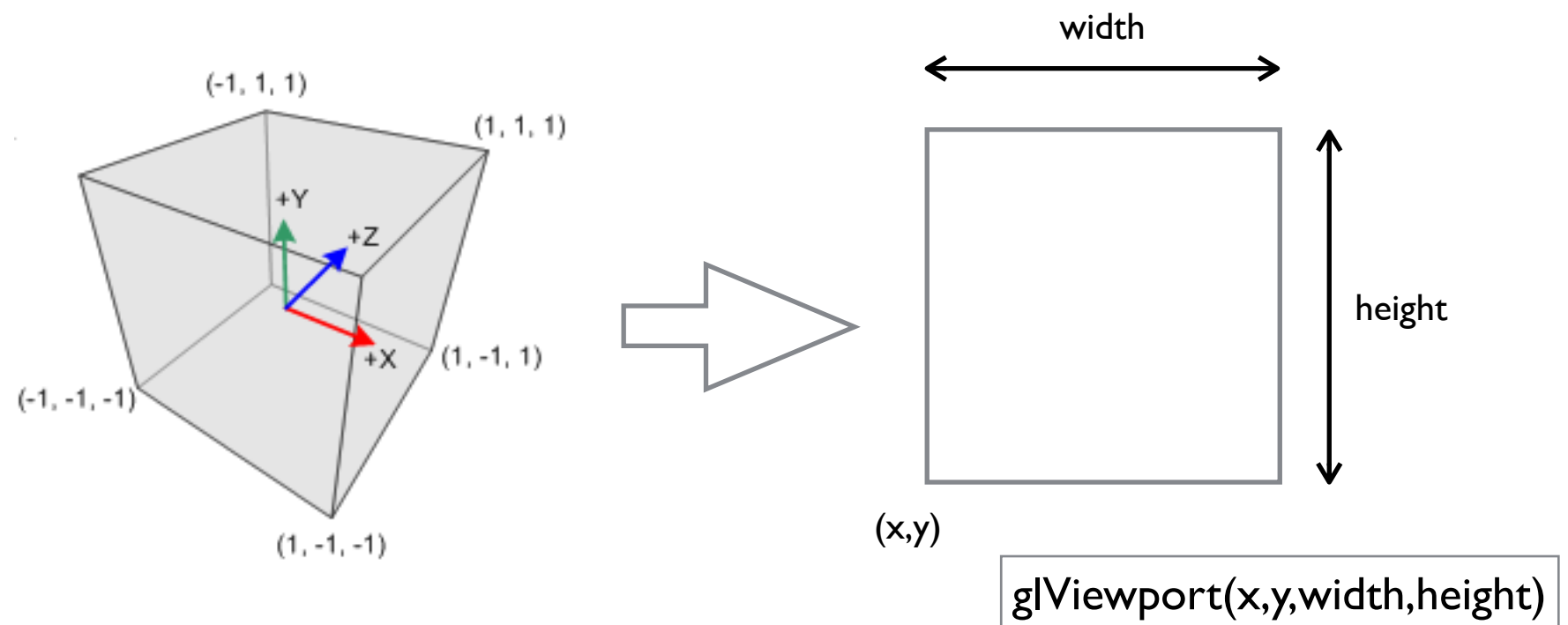
- As primitivas (ou partes) visíveis na imagem final são as que estão contidas no volume de visão
- O processo que elimina as partes fora do volume de visão é denominado por recorte (clipping)
- É matematicamente mais simples efetuar o recorte em volumes canónicos do que para um caso geral
- Um tronco de pirâmide pode ser transformado num paralelepípedo...

Transformação para o Visor (Viewport Transformation)

Primitivas Gráficas 3D



- Transforma o volume canónico num paralelepípedo com as dimensões do visor, e canto inferior esquerdo em $(x,y,0)$.
- Os valores de z são transformados para o intervalo $[0,1]$
- Pergunta: Qual a transformação composta que alcança este efeito?

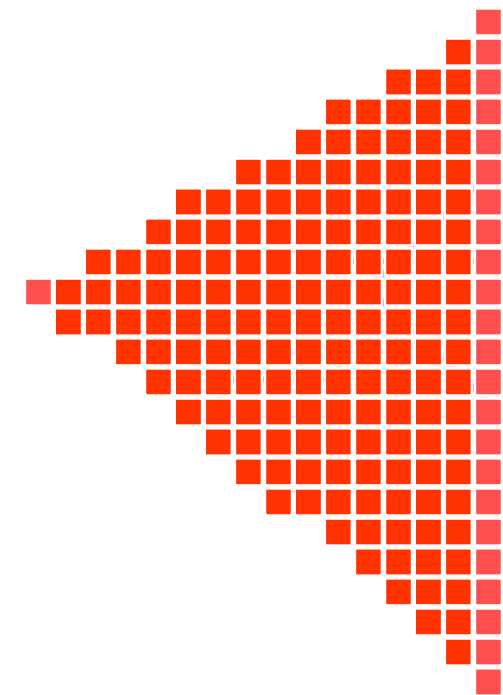


Conversão por Varrimento (Scan conversion)

Primitivas Gráficas 3D



- Conversão de polígonos em pixéis
- Os atributos nos vértices são interpolados nos pixéis interiores (cor, profundidade, ...)



Primitivas Gráficas 3D



- Efetuar o teste de visibilidade ao nível de cada pixel
- algoritmo z-buffer
- Composição: misturar com os pixéis já desenhados (ex. transparências)
- Escritas mascaradas

