OpenGL – Conceitos Básicos

Prof. Jonh Edson R. de Carvalho

Tópicos da Apresentação

- Visualização 3D : Projeções
- Modelo da Câmera Fixa
- Transformações Geométricas

Visualização 3D : Projeções

- Em uma cena 3D temos um número maior de opções na geração de uma cena.
- De frente, de trás, por cima, etc...
- O observador pode estar no meio de um conjunto de objetos, por exemplo, um prédio.
- As representações 3D devem ser projetadas numa superfície plana.
- Observador é fixo, geralmente sobre o eixo z.

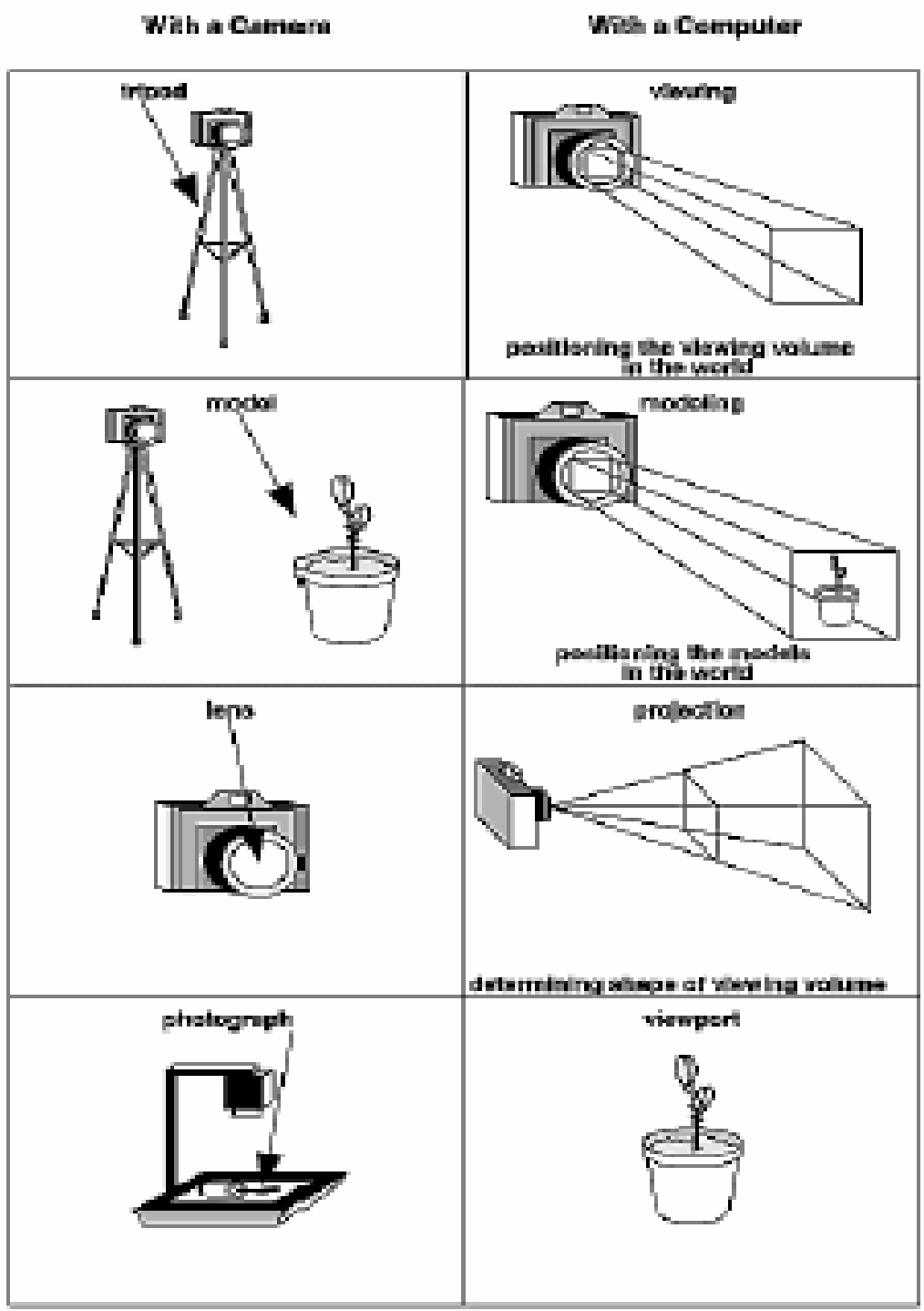
Visualização 3D : Projeções

- Geração de visualização de uma cena 3D é análoga ao processo de fotografia:
  - posicionar a câmera no espaço (transformação de Visualização)
  - decidir a orientação da câmera
    - em qual direção apontá-la e rotacioná-la
  - Acerte as lentes da câmera ou ajuste o zoom (transformação de projeção)

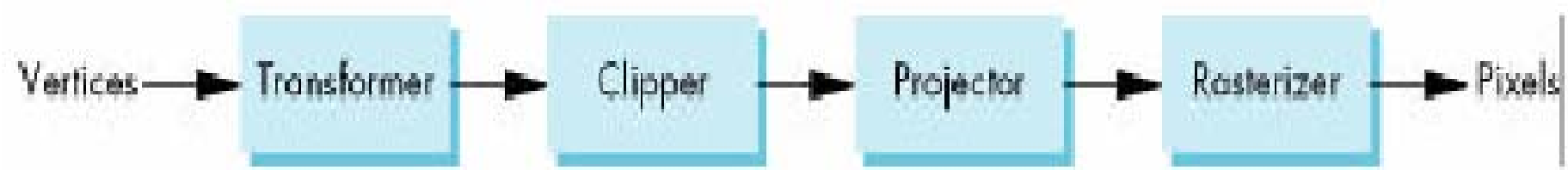
Visualização 3D : Projeções

- apertar o botão: a cena é transferida para uma janela obedecendo o tamanho da abertura da câmera.
- luz da superfície visível é projetada sobre o filme da câmera.

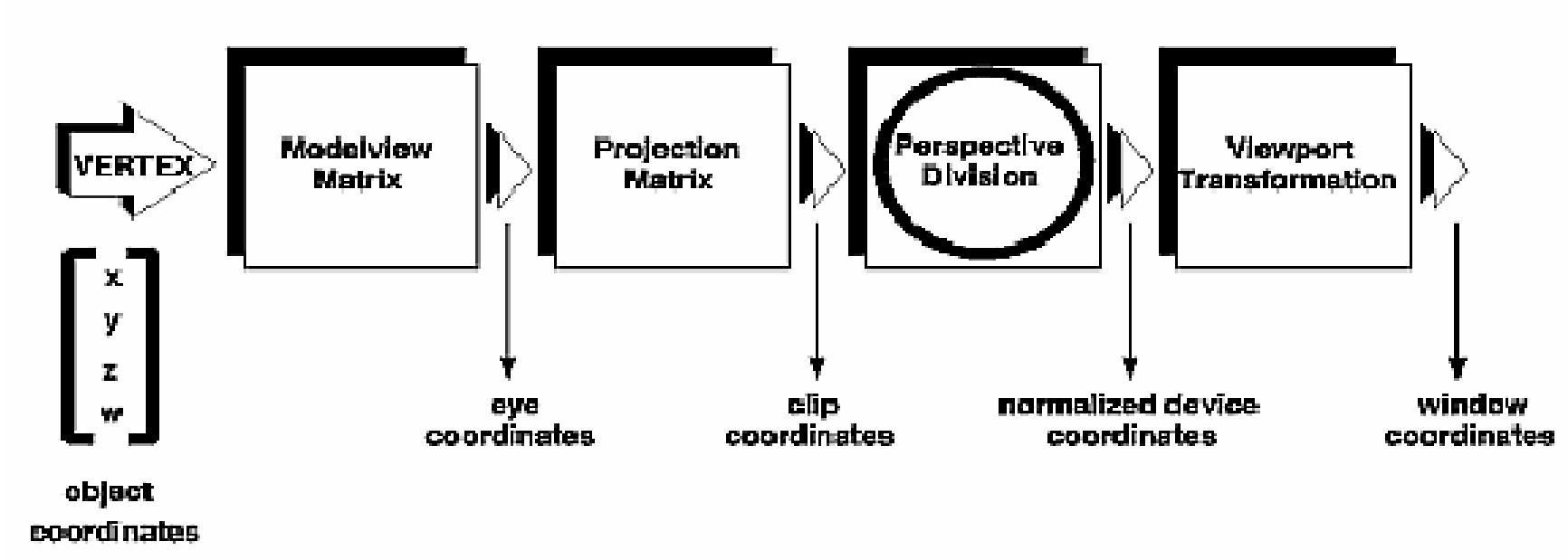
Câmera Fixa


With a Camera
With a Computer
tripod
viewing
positioning the viewing volume in the world
model
modeling
positioning the models in the world
lens
projection
determining shape of viewing volume
photograph
viewport

Pipeline de uma Aplicação Gráfica



Estágios de Transformação de Vértices



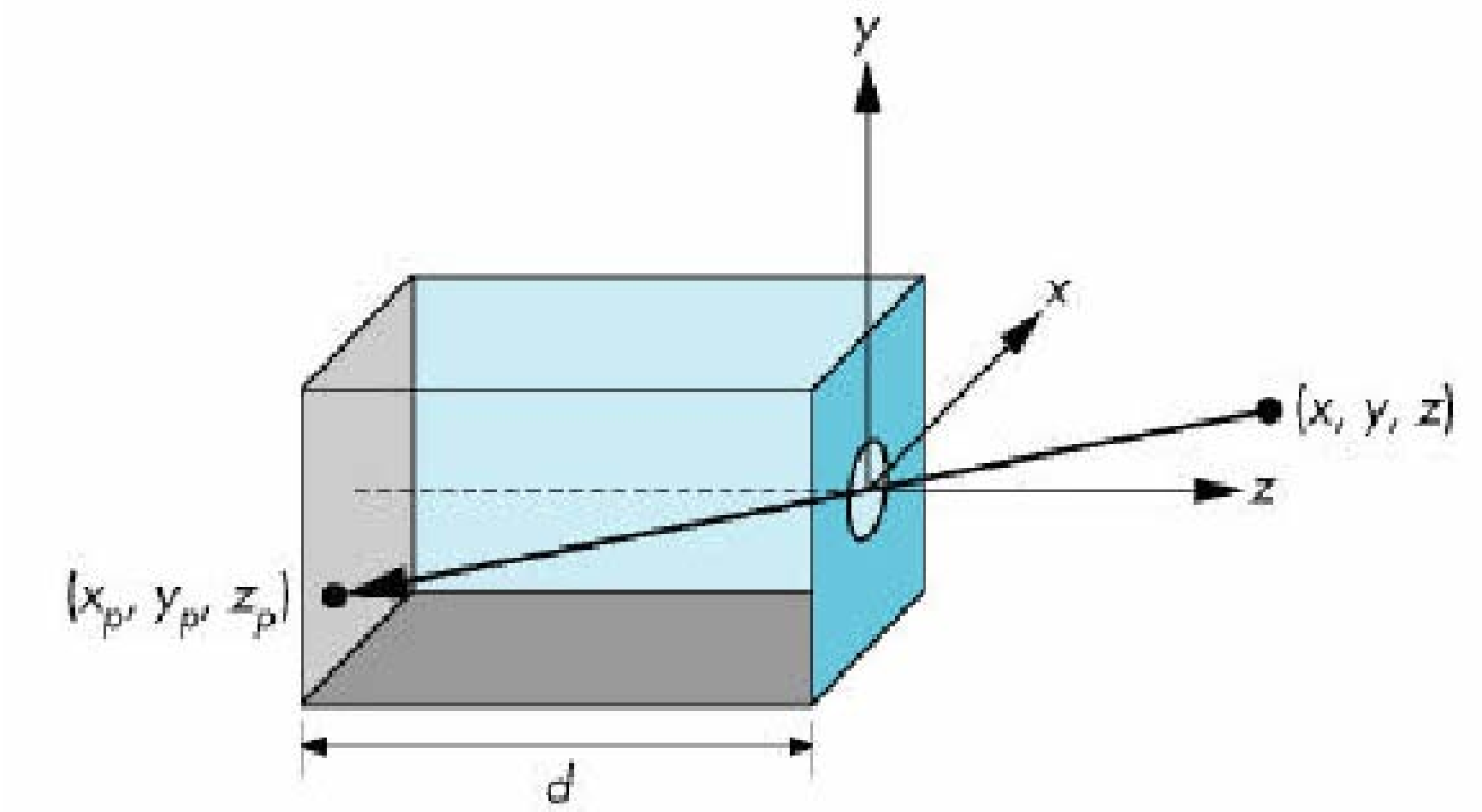
Estágios de Transformação de Vértices

- Para especificar a visualização (viewing), modelagem (modeling), e as transformações de projeção (projection), é preciso construir uma matriz 4 x 4 **M,** a qual é multiplicada pelas coordenadas de cada vértice **v** na cena para realizar a transformação **v' = M . V** .
- As transformações de visualização e modelagem especificadas são combinadas para formar a matriz modelview, que é aplicada nas coordenadas iniciais do objeto para gerar as coordenadas do olho (eye coordinates).
- Depois, se foi especificado arbitrariamente planos de corte (clipping planes) para remover certos objetos da cena ou para prover vistas cortadas de objetos (cutaway views), estes planos de corte são então aplicados.
- Depois disso, o OpenGL aplica a matriz de projeção (projection matrix) para obter as coordenadas cortadas (clip coordinates). Esta transformação define o volume de visualização (viewing volume). Os objetos fora deste volume são cortados e portanto não são desenhados na cena final.

Estágios de Transformação de Vértices

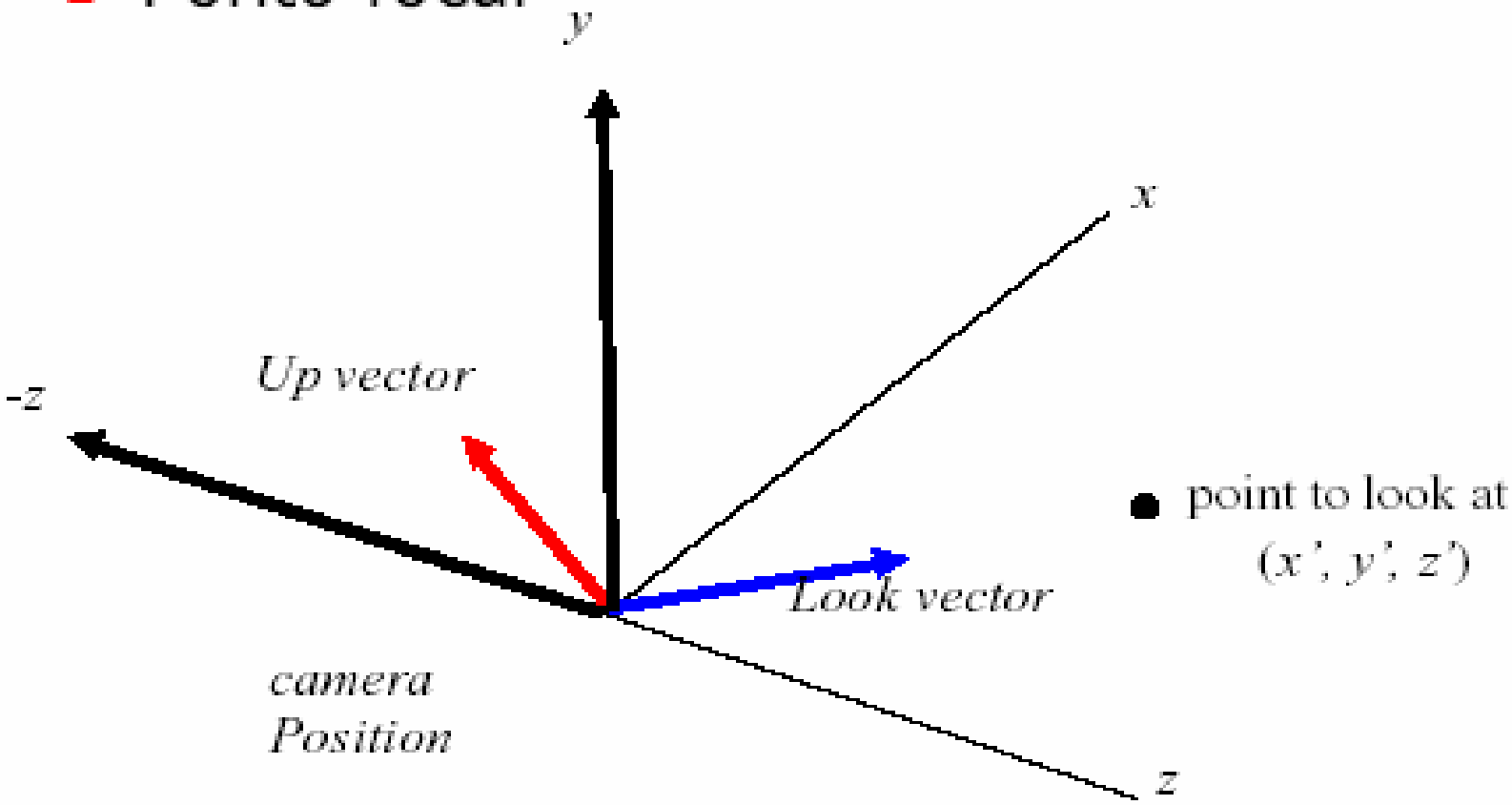
- Depois deste ponto a divisão de perspectiva (perspective division) é executada dividindo os valores das coordenadas (x,y,z,w) por w, para produzir as coordenadas normalizadas do dispositivo (normalized device coordinates).
- Finalmente as coordenadas transformadas são convertidas para as coordenadas da janela (window coordinates) através da aplicação da transformação de viewport.
- Pode-se manipular as dimensões do viewport a fim de aumentar, diminuir, achatar a imagem final.

The Pinhole Camera

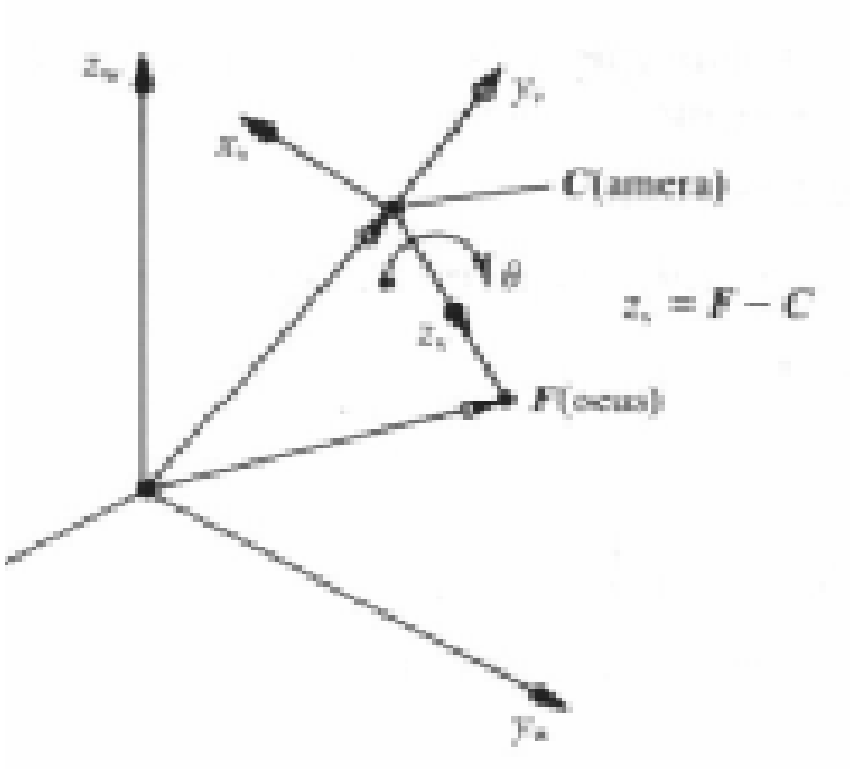


Parâmetros de uma Câmera Virtual

- Direção
  - Ponto focal



Parâmetros de uma Câmera Virtual



Posição da Câmera no Espaço

- O primeiro parâmetro relacionado a uma camera virtual é a sua posição no espaço.
- Essa posição pode ser caracterizada por um ponto no espaço 3D.
- Nesse ponto será posicionada a camera virtual, denominado ***view point***.
- O *view point* pode ser definido explicitamente por suas coordenadas cartesianas (***x,y,z***). Uma outra forma de defini-lo é a partir de suas coordenadas esféricas (o equivalente as coordenadas polares bidimensionais).

Direção de Observação

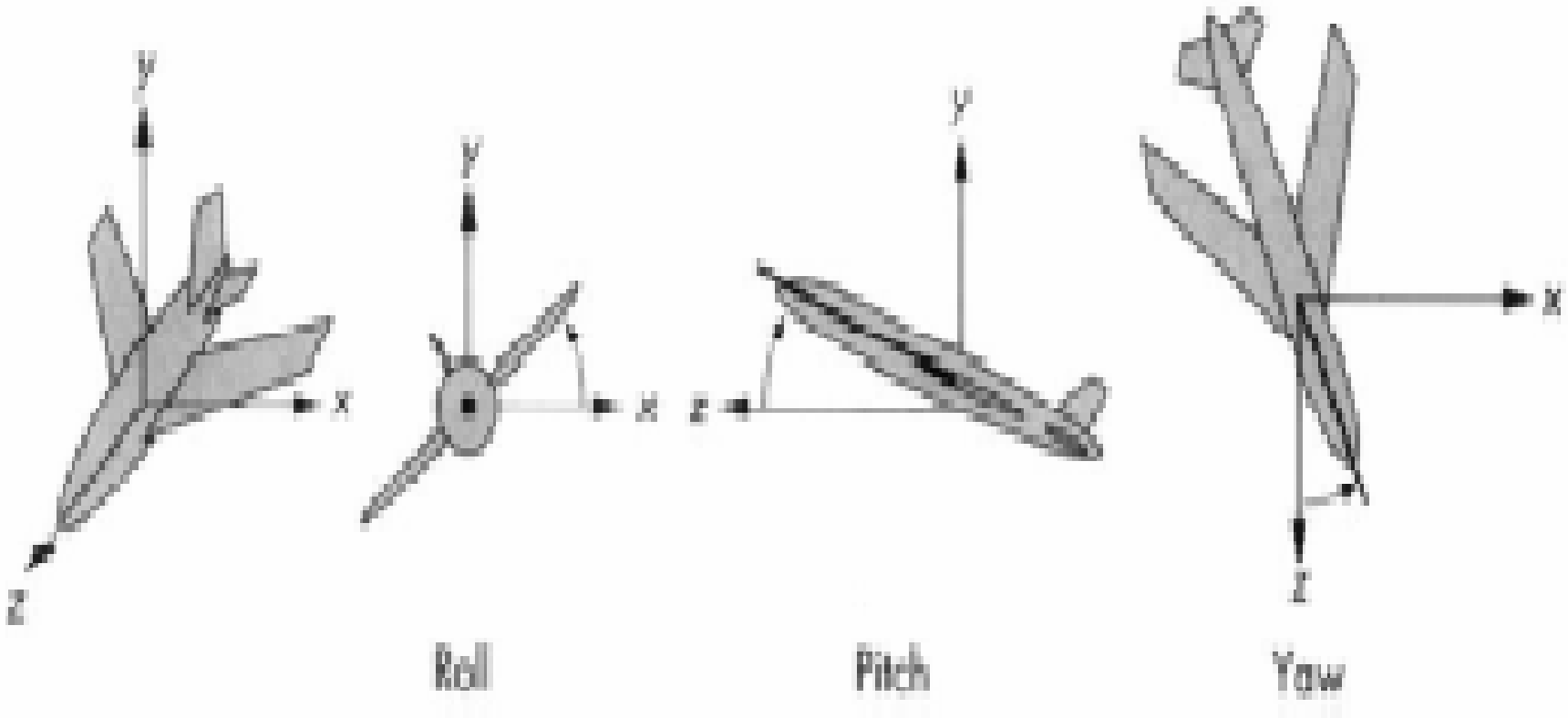
- Uma vez fixado o *view point* em um ponto ***C*** no espaço, temos que definir a direção para a qual iremos apontar a camera.
- Essa direção será dada a partir de um vetor, cuja origem é o ponto ***C*** e por um outro ponto.
- Esse ponto é denoninado ***foco*** e está associado ao ponto ***F.***
- A definição de um ponto no espaço associado a direção de observação é simples, porém pouco intuitiva.

Direção de Observação

- Uma forma alternativa, e bem mais intuitiva, pode ser definida com base em dois ângulos, relativos a rotações em torno do ponto ***C***.
- Essas rotações são definidas da seguinte forma :
  - suponha que o vetor que define a direção de observação é unitário. Nesse caso o lugar geométrico de todos os possíveis pontos extremos desse vetor é uma esfera de raio 1, centrada no ponto ***C***. Portanto, para definirmos o ponto extremo do vetor direção sobre essa esfera, basta especificar um angulo relativo a elevação do ponto e outro relativo ao seu deslocamento horizontal.

Direção de Observação

- Esses movimentos são denominados ***tilt*** e ***pan*** (ou ***pitch***). A figura abaixo ilustra esses dois movimentos.

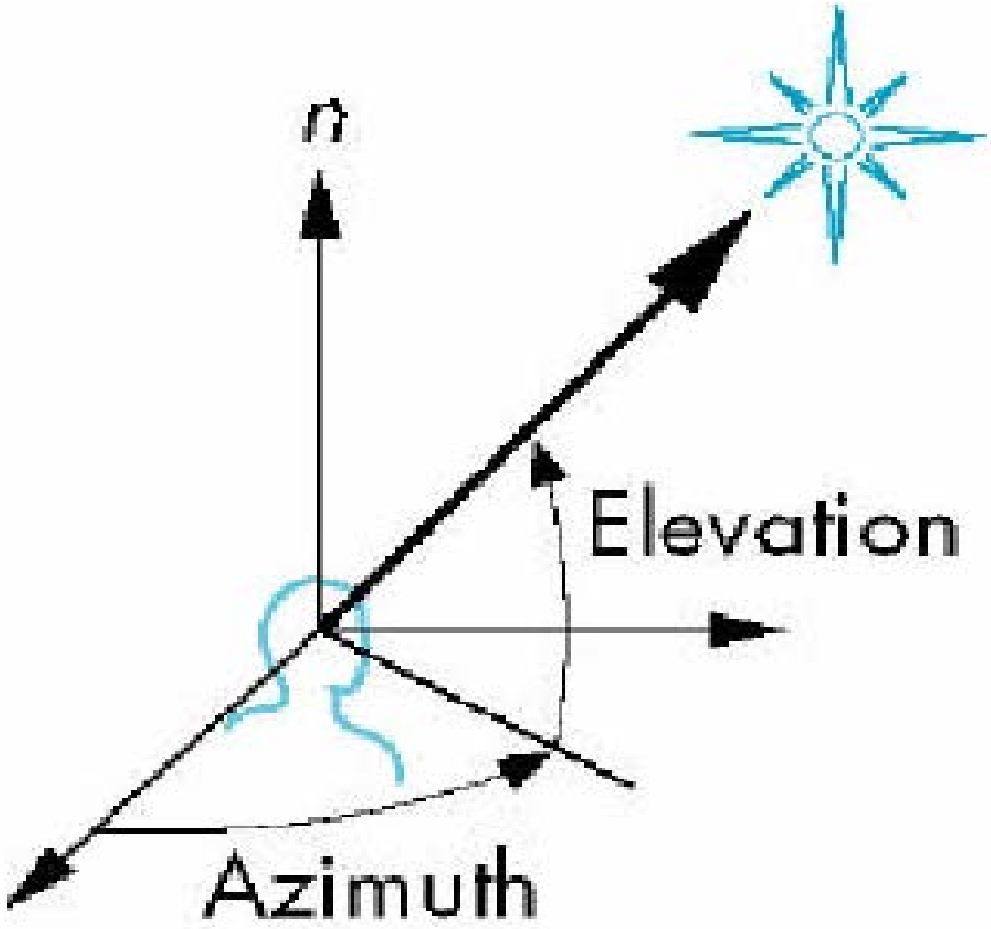


Orientação da Câmera

- Uma vez definida a localização da camera e sua direção de observação ainda temos um grau de liberdade :
  - a orientação da camera relativa a direção de observação.
  - Em outras palavras, dada uma direção de observação, é possível ainda rotacionar a camera em relação a esse vetor de um angulo θ, como mostra a figura no slide 12.
  - Na prática, essa rotação é equivalente ao posicionamento da camera :
    - horizontal, vertical, ou “cabeça para baixo”, por exmplo. Esse movimento de rotação é denominado ***roll*** e pode ser visto também na figura anterior.

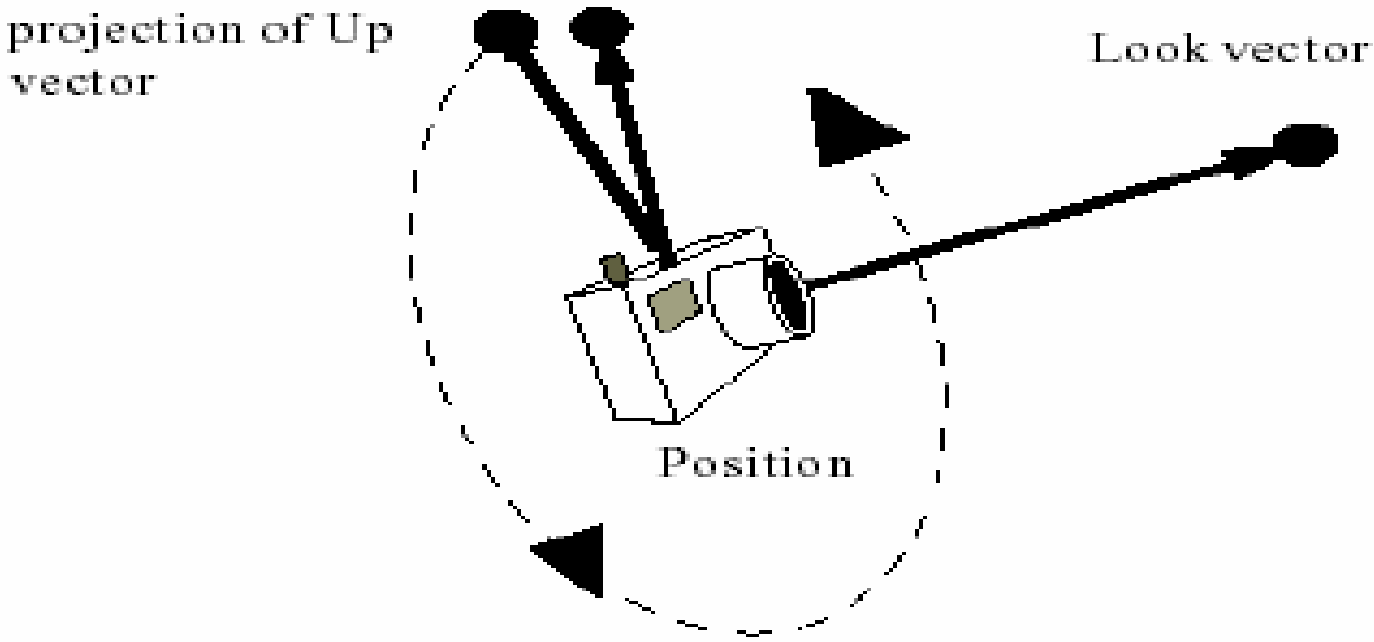
Direção de Observação

- Direção



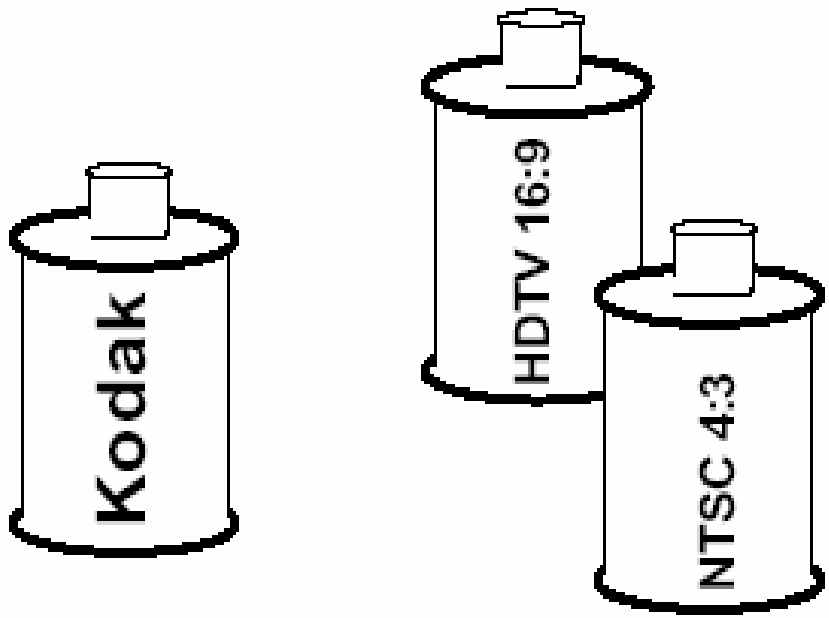
Direção de Observação

- Direção



Parâmetros de uma Câmera Virtual

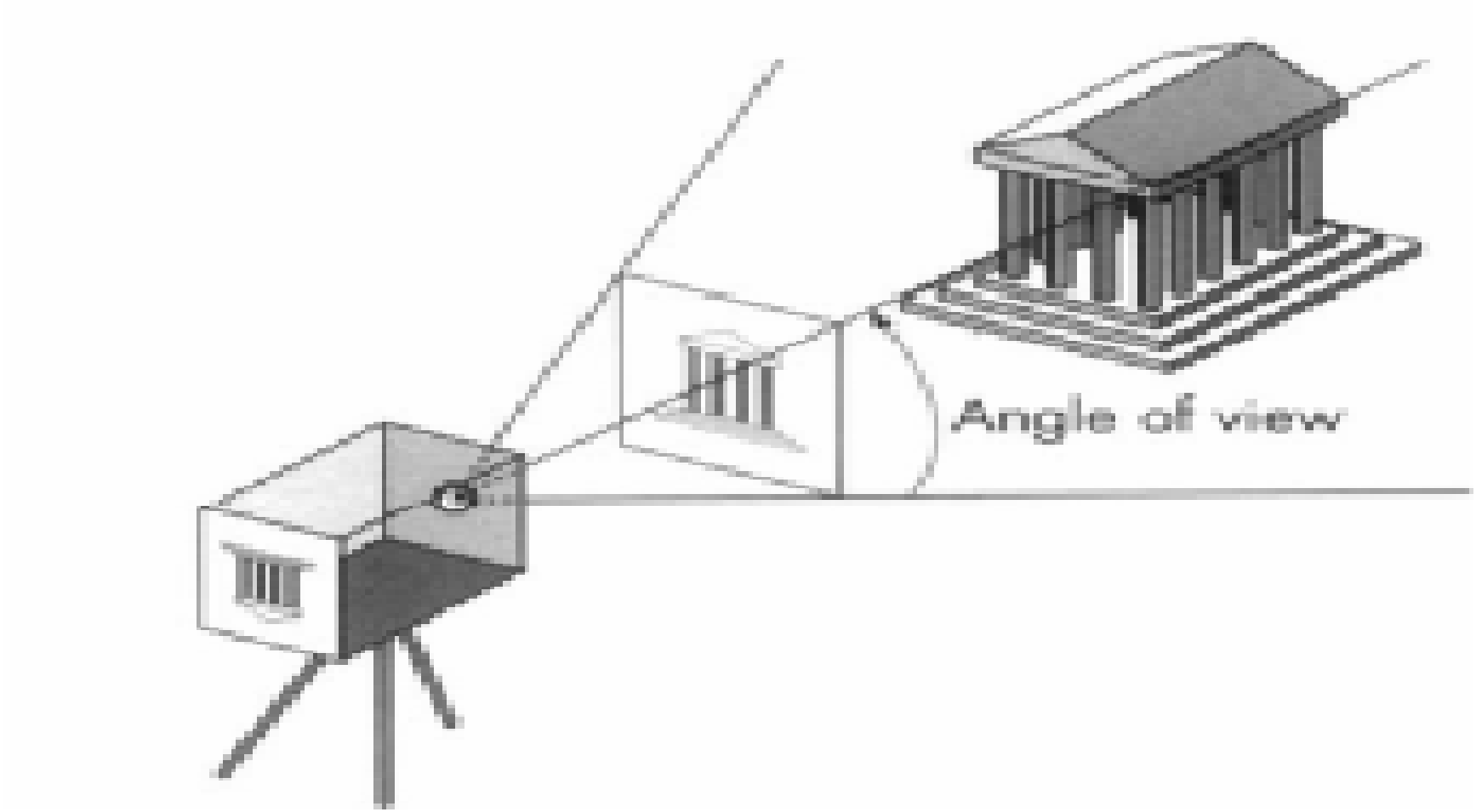
- Razão de Aspecto
  - Conceito analogo ao tamanho do filme utilizado em uma camera real
  - Determina a proporção entre a largura e altura da imagem
    - 1:1 => Imagem quadrada
    - 2:1 => Cinema (letterbox)
    - 4:3 => TV NTSC
    - 16:9 => HDTV



Parâmetros de uma Câmera Virtual

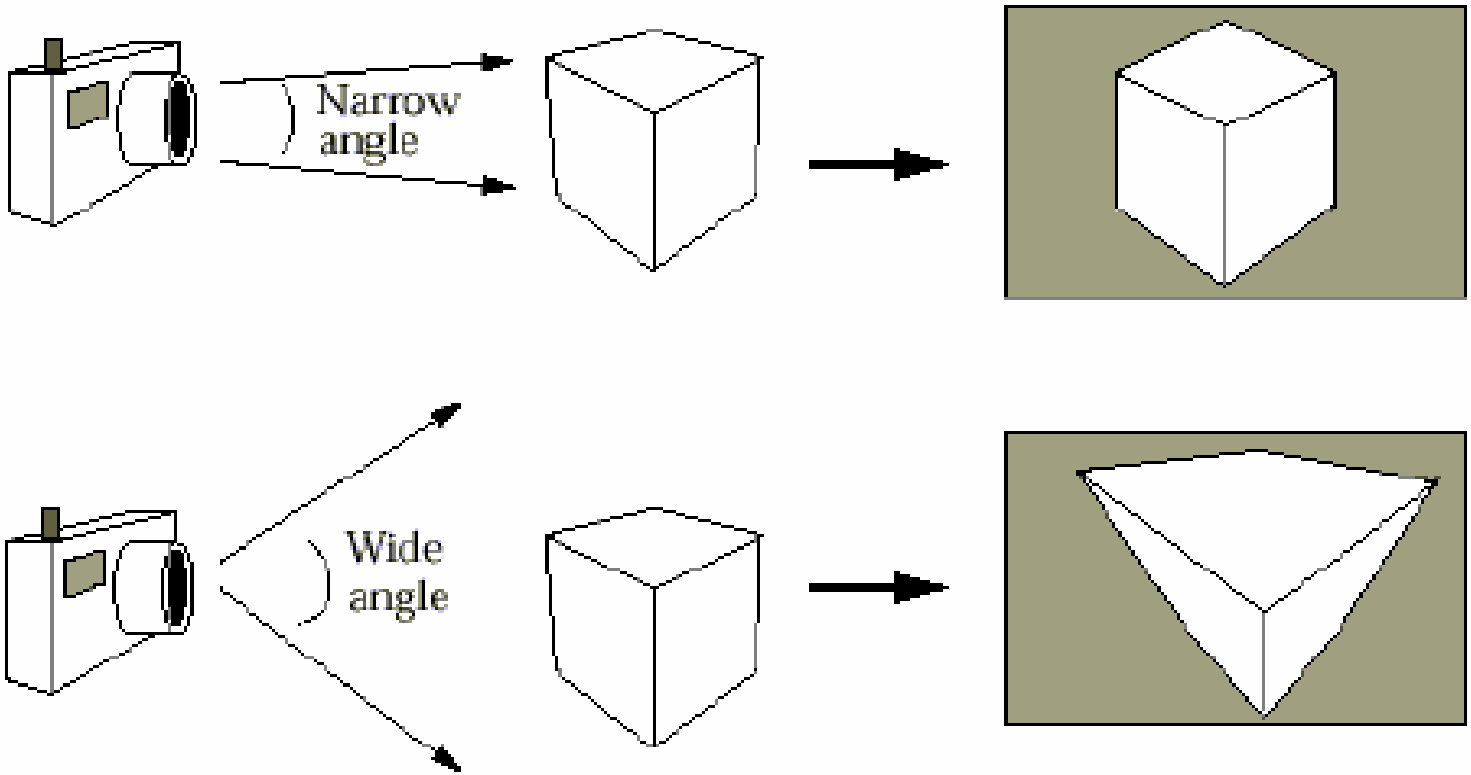
- Dados o ponto e a direção de observação ainda não podemos caracterizar de forma precisa quais objetos serão visíveis. Tão pouco como será a imagem gerada.
- Em uma camera real, podemos variar o tipo de lente utilizada para obter diferentes efeitos na imagem final. A definição de uma lente está associada a diversos parâmetros relativos a imagem. O primeiro é o ***campo de visão*** (***FOV – Field Of Vision***).
- O *campo de visão* define um ângulo dentro do qual os objetos podem ser vistos, como na figura a seguir. Logo, dada uma lente podemos definir a região da cena que será visível, e caracterizar a fase de recorte. Essa região tem a forma de um cone de pirâmide de base retangular.

Parâmetros de uma Câmera Virtual



Parâmetros de uma Câmera Virtual

- Angulo de Visão

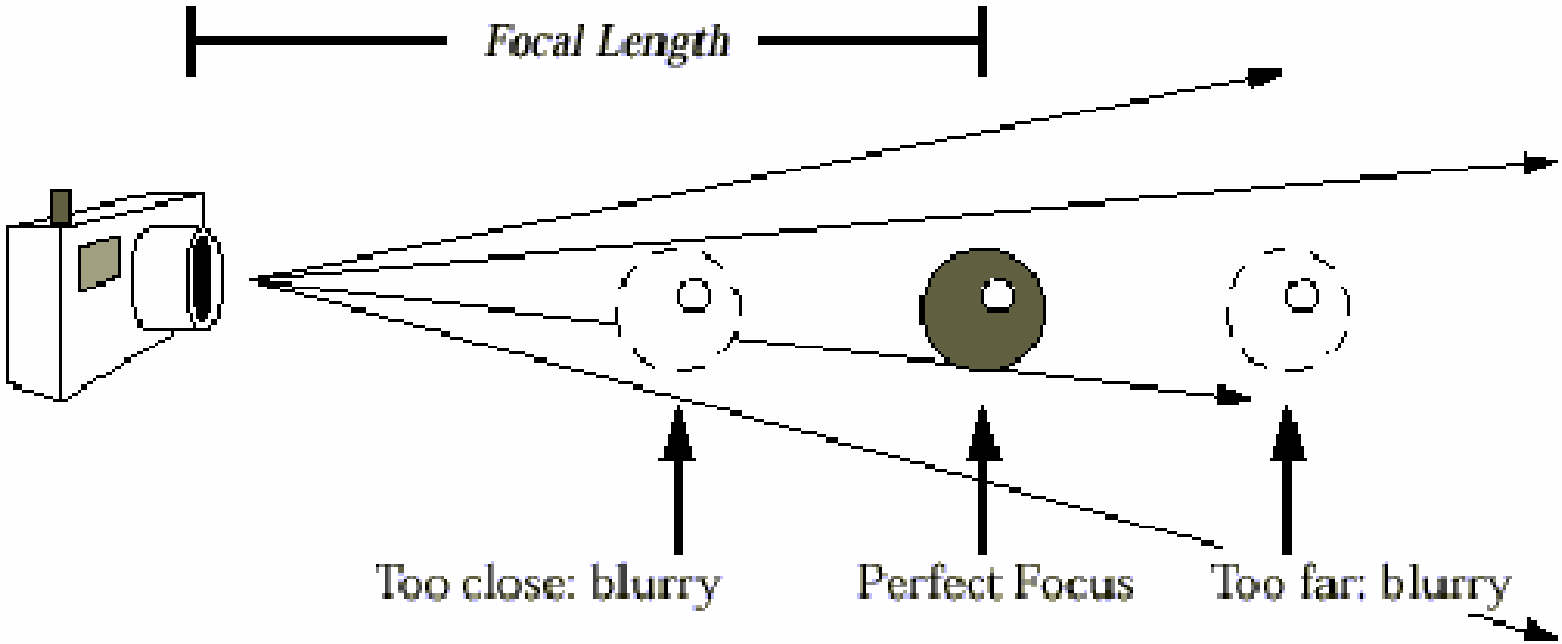


Parâmetros de uma Câmera Virtual

- Em uma máquina fotográfica real, objetos muito próximos ou muito distantes tendem a aparecer “deformados” na fotografia. Dizemos que esses objetos estão “fora de foco”, pois se encontram em uma região onde a imagem definida através da lente não fica nítida.

Parâmetros de uma Câmera Virtual

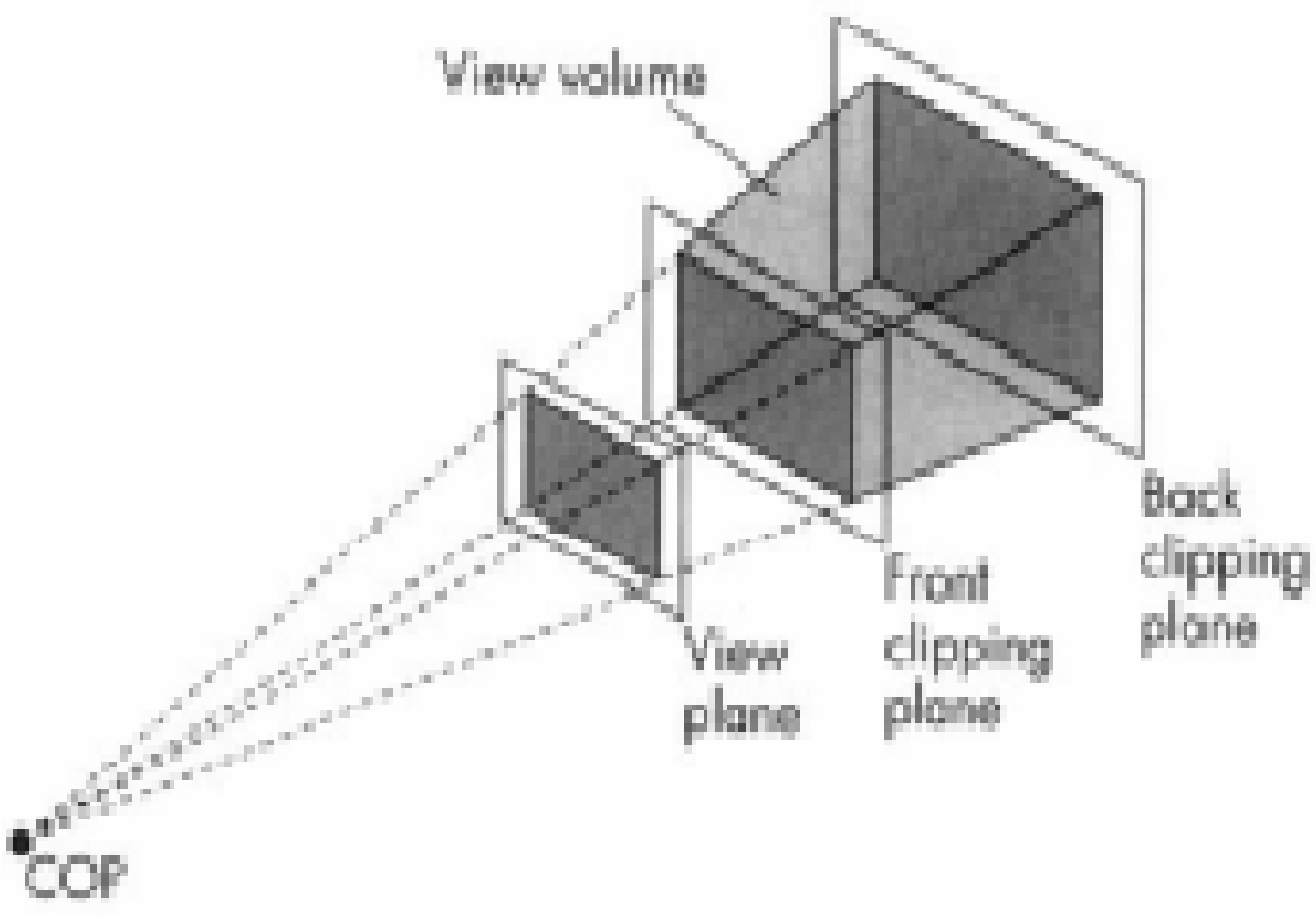
- Controle de Foco



Parâmetros de uma Câmera Virtual

- Da mesma forma, iremos associar uma região do campo de visão onde os objetos aparecerão sem distorções na imagem final. Essa faixa é definida por dois planos : o ***plano de recorte frontal*** e o ***plano de recorte dorsal*** (respectivamente *front clipping plane* e *back clipping plane* na figura a seguir).
- Esses planos também podem ser chamados plano mais próximo (*near plane*) e plano mais distante (*far plane*). Nesse caso a região visível passa a ser definida pelo tronco de pirâmide (figura a seguir). Essa porção do espaço visível é denominada ***Volume de Visão*** (*view volume*).

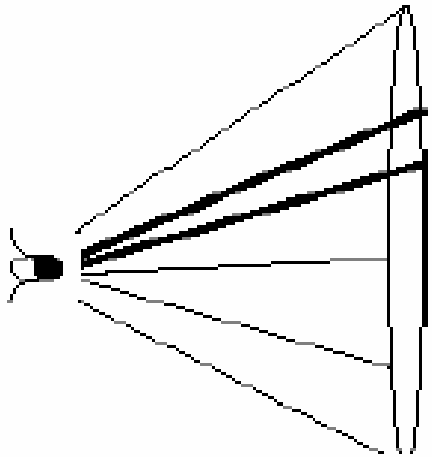
Parâmetros de uma Câmera Virtual



Parâmetros de uma Câmera Virtual

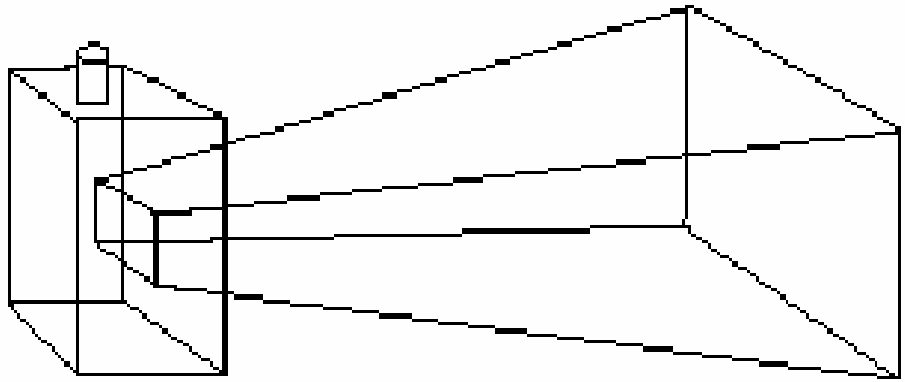
- Volume de Visão

Olho



Volume de Visão em Cone

Camera Sintetica



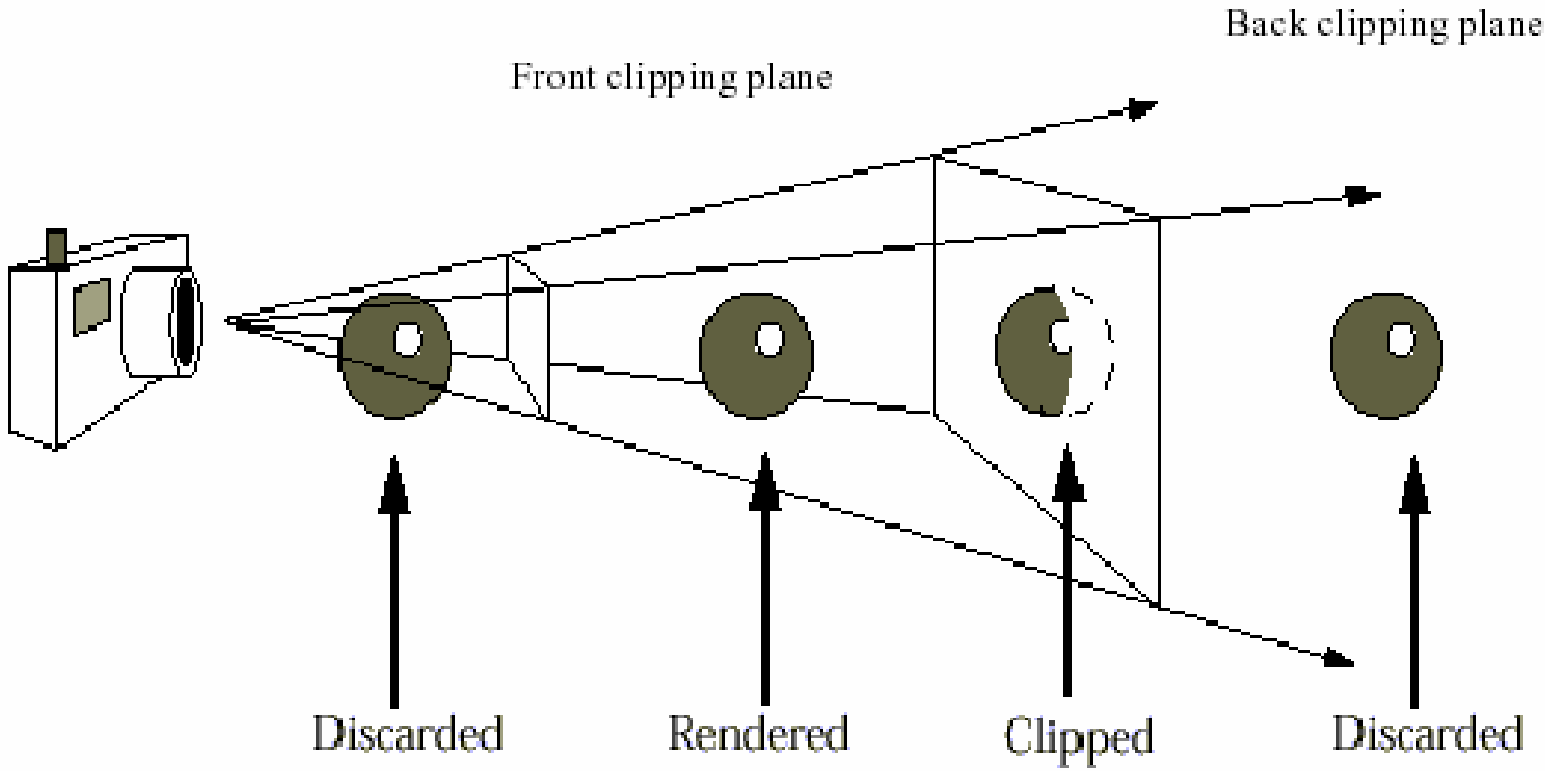
Aproximação do Volume de Visão

Parâmetros de uma Câmera Virtual

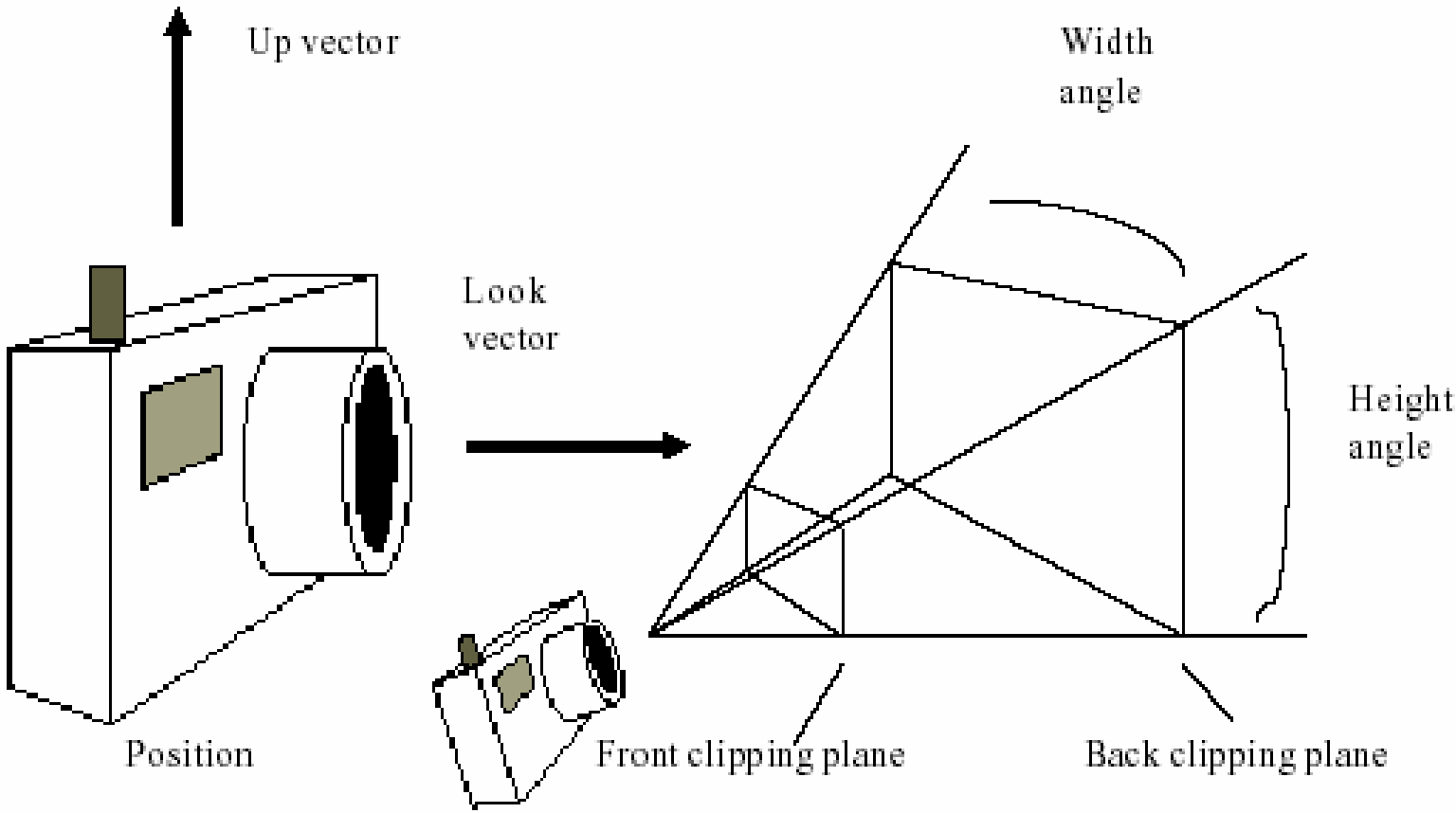
- Em resumo, a fase de recorte elimina os objetos que não se encontram dentro do campo de visão da camera virtual, além dos objetos que se encontrariam fora de foco.

Parâmetros de uma Câmera Virtual

- Volume de Visão



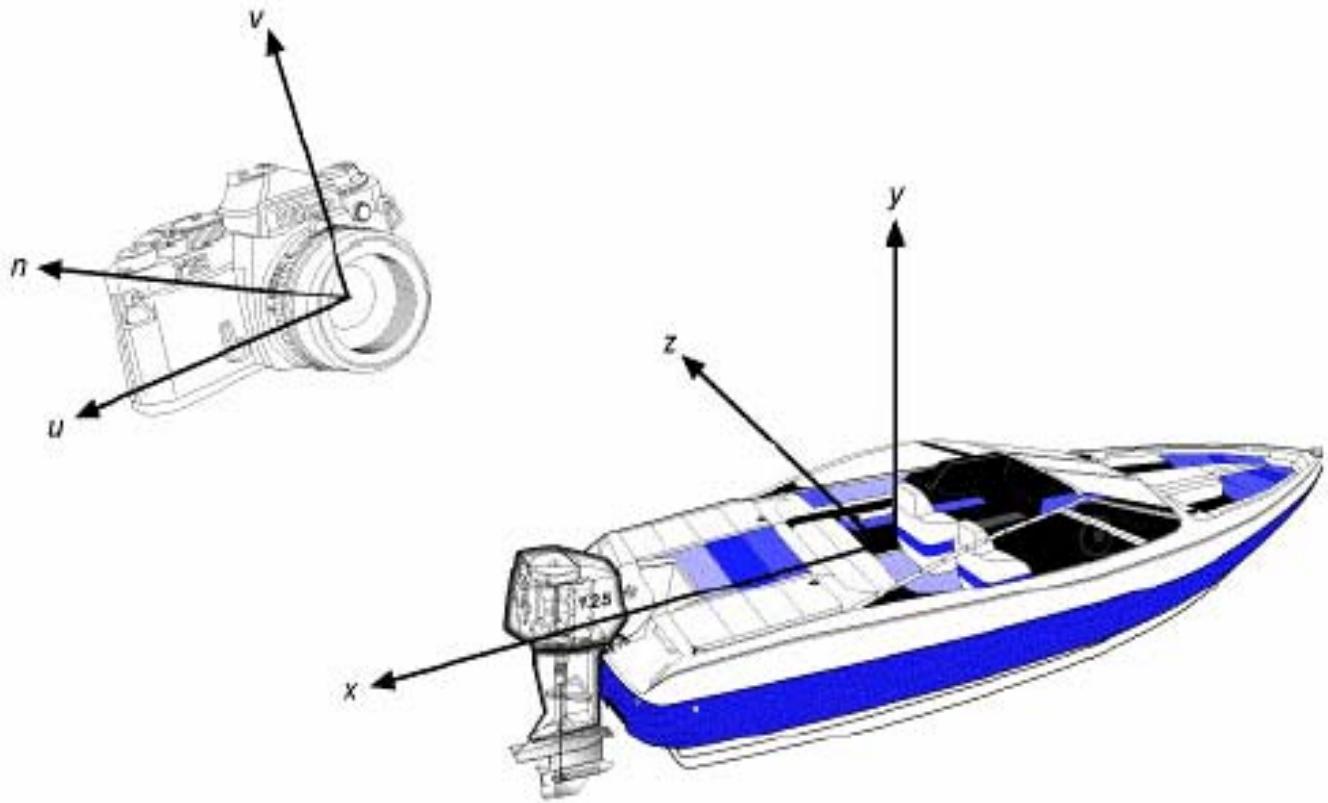
Parâmetros de uma Câmera Virtual



Sistema de Coordenadas da Câmera

- A definição de uma câmera virtual em um sistema de visualização está diretamente associada a existência de um segundo sistema de coordenadas dentro da aplicação – relacionado com a camera virtual – tal como na figura a seguir.

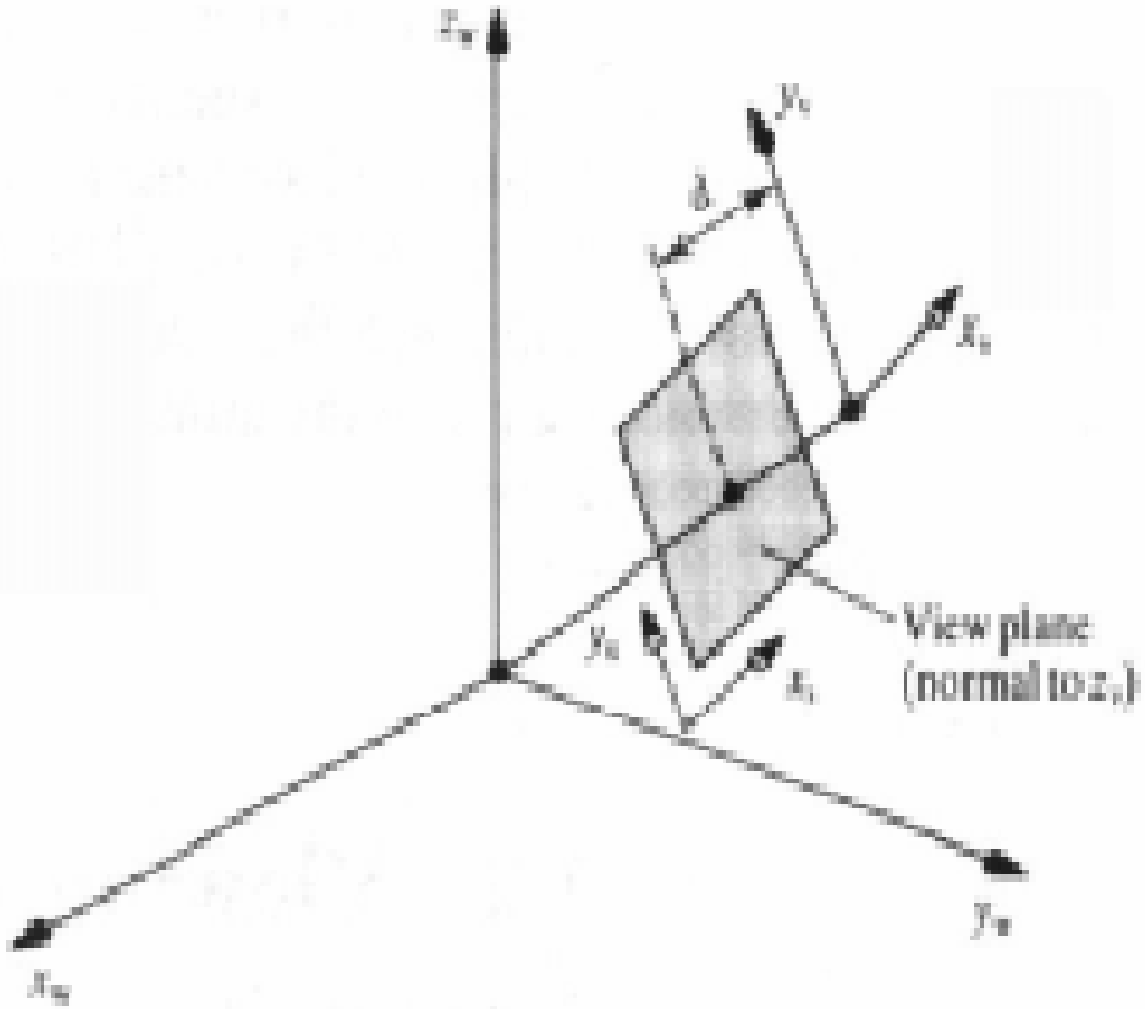
Sistema de Coordenadas da Câmera



77 - Caracterização dos dois sistemas de coordenadas definidos por uma camera virtual dentro de um sistema de visualização.

Sistema de Coordenadas da Câmera

- Dessa forma temos :
- Sistema de Coordenadas Global
  - É nesse sistema que a cena (objetos e camera) é descrita. Esse sistema é definido pelos eixos coordenados ***xw***, ***yw*** e ***zw*** (figura a seguir).
- Sistema de Coordenadas da Camera
  - É definido a partir do posicionamento da camera virtual dentro do sistema
  - de coordenadas global. Será com base nesse sistema de coordenadas que o recorte e a projeção serão defindos. Esse sistema é definido pelos eixos coordenados ***xc***, ***yc*** e ***zc*** (figura a seguir).
- Sistema de Coordenadas da Imagem
  - É o sistema onde estão definidos os pixels que irão compor a imagem. É definido pelos eixos ***xs*** e ***ys*** (figura a seguir).



Definição dos sistemas de coordenadas global da camera e da imagem.

Utilização de um Sistema de Visualização

- Uma vez definidos os parâmetros necessários a construção de um sistema de visualização uma pergunta aparece : como interagir com esse sistema ?
- Duas abordagens podem ser utilizadas :
  - Baseada na camera
    - Camera é estática e os objetos se movimentam no ambiente.
  - Baseada nos objetos
    - Os objetos são estáticos e a camera se movimenta dentro da cena.
- A figura a seguir ilustra a diferença entre as duas abordagens.

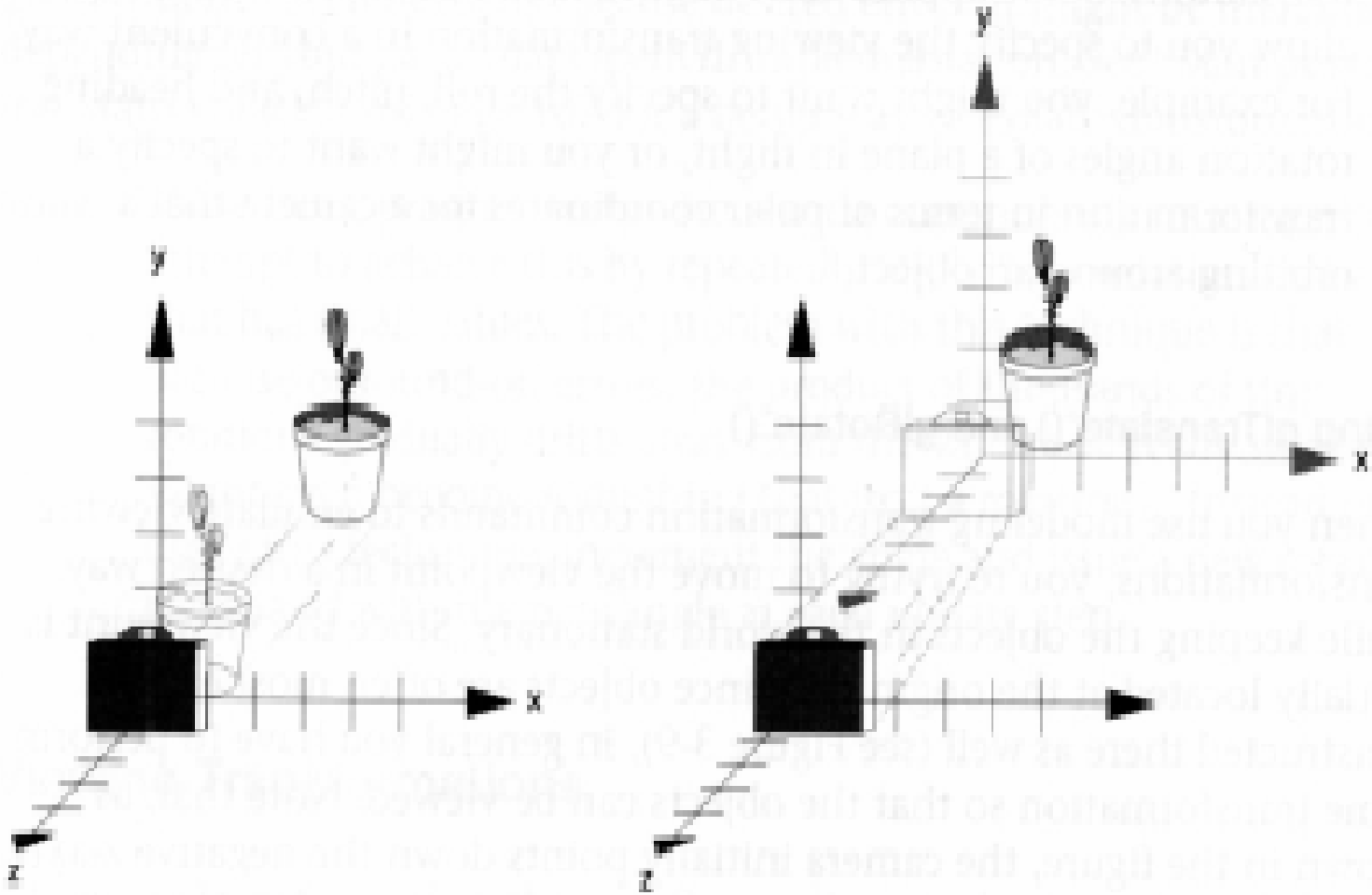


Figura 82 – Duas abordagens para movimentação em um sistema de visualização : objetos se movem e a camera fica estática (à esquerda) ou objetos ficam estáticos enquanto que a camera se movimenta (à direita).

Utilização de um Sistema de Visualização

- No primeiro caso podemos fazer uma analogia com um fotografo de estúdio.
- Nesse caso pode ser mais simples posicionar os objetos em relação a que fica fixa em um tripé.
- No entanto, essa abordagem não se mostra adequada para fotografias da natureza ou mesmo de objetos arquitetônicos, como edificações por exemplo.
- Nesses casos o único modelo possível é movimentar a camera para obter o melhor posicionamento, já que os objetos estão “fixos” na cena.

Utilização de um Sistema de Visualização

- Do ponto de vista matemático, as duas abordagens são equivalentes. No entanto, a interface fornecida por uma aplicação que trabalhe com uma ou outra abordagem pode ser bastante diferente.
- Uma vez definido o posicionamento e as características da camera virtual – o que determina que objetos serão visíveis – a fase subsequente é a da geração da imagem.
- As ferramentas para essa tarefa são as ***Transformações Projetivas.***

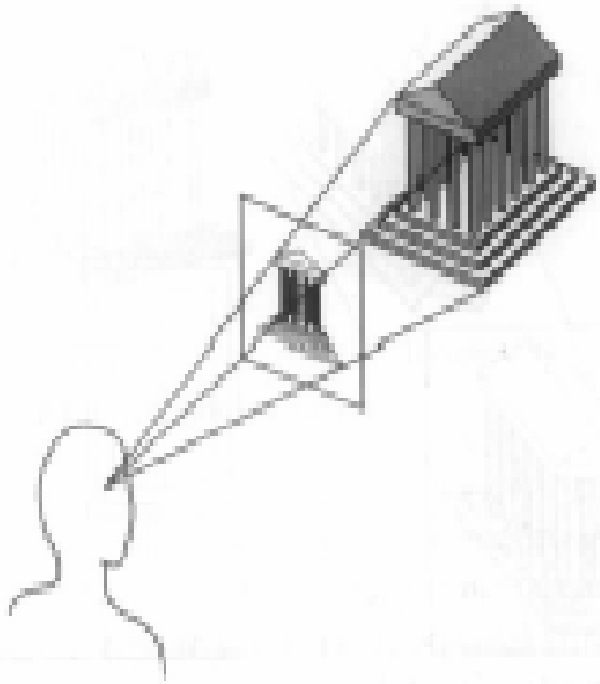
Projeções Geométricas Planares

- O processo de geração de uma imagem bidimensional, obtida a partir de um objeto tridimensional, é bastante conhecido. Seus princípios básicos são os mesmos associados ao processo de formação de uma imagem pelo sistema óptico humano.
- Como premissa temos a propagação retilínea da luz. Portanto, para que um objeto qualquer possa ser visualizado é necessário que os raios de luz que partem dele (por um processo de reflexão ou emissão de luz) atinjam o sistema óptico do observador.
- Esses raios de luz podem ser considerados como retas que partem de cada ponto localizado na superfície visível do objeto.

Projeções Geométricas Planares

- Imagine um plano se interpondo entre o objeto e o observador (figura a seguir).
- Todas as retas (raios de luz) que partem do objeto e chegam no observador interceptam esse plano.
- Se calcularmos essas interseções e registrarmos a cor (intensidade de luz) do ponto correspondente no objeto, teremos a sua imagem bidimensional.

Projeções Geométricas Planares

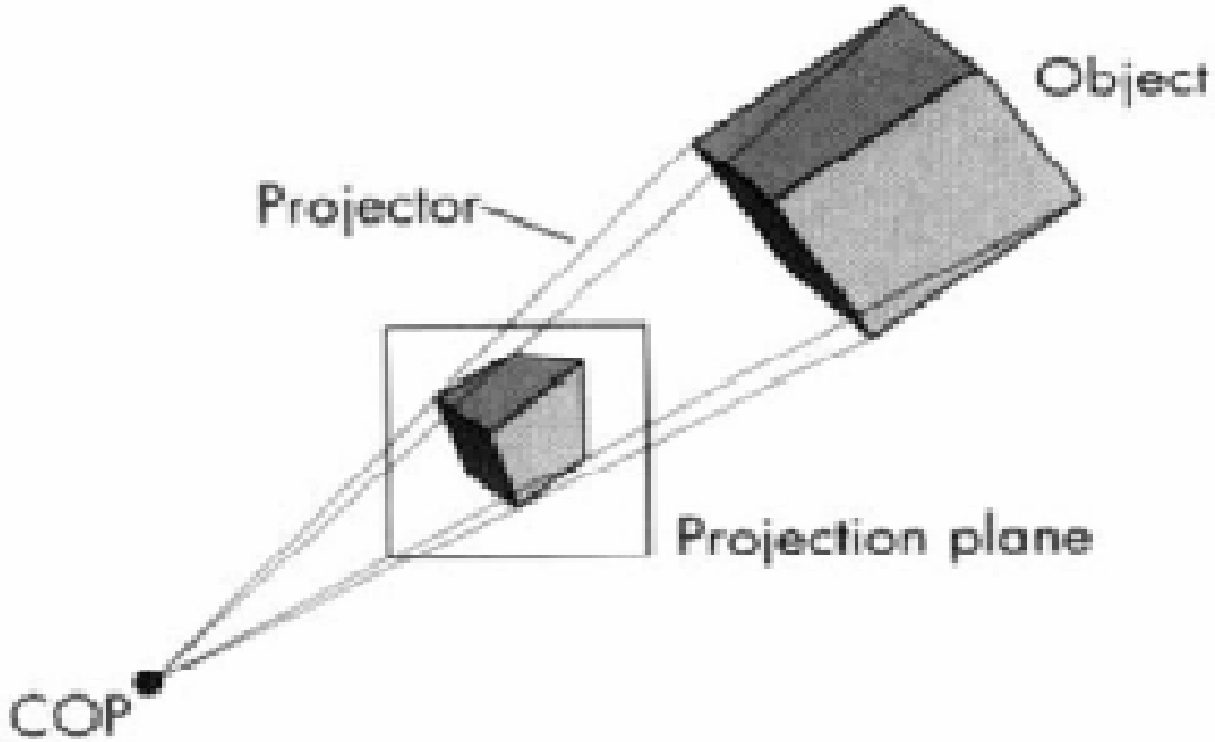


- Processo de projeção : formação de uma imagem bidimensional a partir de um objeto tridimensional.

Projeções Geométricas Planares

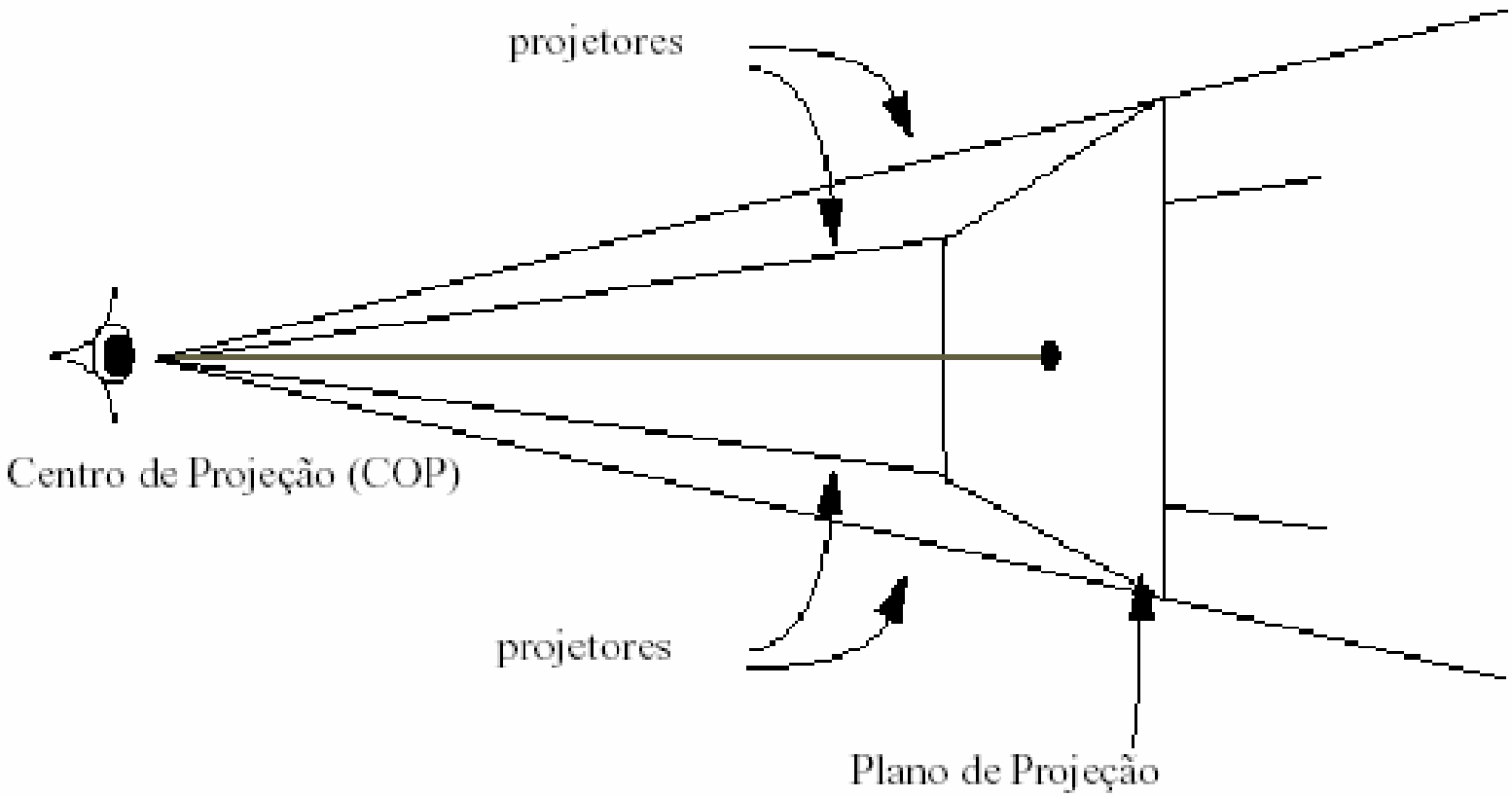
- **Componentes básicos de uma projeção:**
- No processo descrito anteriormente, temos uma série de elementos
- envolvidos. Os elementos necessários para a construção de uma projeção são :
  - ***Objetos***
  - ***Projetores***
    - São os raios de luz que partem da superfície do objeto na direção do observador.
  - ***Plano de Projeção***
    - É o lugar geométrico onde a imagem será gerada.
  - ***Centro de Projeção (COP – Center of Projection)***
    - Nada mais é que o ponto no espaço onde o observador está localizado.
- Na figura podemos observar cada um desses componentes.

Projeções Geométricas Planares



Componentes básicos envolvidos no processo de projeção.

Projeções Geométricas Planares



Projeções Geométricas Planares

- Imaginando o processo de geração da projeção (imagem) como um algoritmo temos:
- Algoritmo GerarProjeção
  - Para cada ponto da superfície do **objeto** faça
    - Traçar um reta (**projetor**) que liga esse ponto ao **COP**;
    - Calcular a interseção do **projetor** como o **plano de projeção**;
    - Atribuir a esse ponto a cor (intensidade de luz) a partir do ponto correspondente da superfície do objeto
  - fim-para
- fim.

Projeções Geométricas Planares

- O conjunto de pontos gerados por esse algoritmo define a ***projeção do objeto***, ou seja, sua a imagem bidimensional.
- Um caso especial ocorre quando o ponto utilizado como centro de projeção
- está a uma distância muito grande dos objetos e do plano de projeção. Imagine, a partir da figura anterior, que o ***COP*** comece a se distanciar. A tendência dos projetores é de se afastarem entre si.
- No entanto esse afastamento tem um limite. No caso do ***COP*** ser levado para uma distância infinita os projetores se tornam retas paralelas. Nesse caso temos a situação mostrada na figura a seguir. Portanto, não temos caracterizado um ponto para o qual os projetores convergem (já que ele está no infinito) mas sim uma ***direção de projeção (DOP - Direction Of Projection)***.
- Conforme veremos a seguir essas duas situações caracterizam dois tipos de projeção : ***perspectiva*** e ***paralela.***

Projeções Geométricas Planares

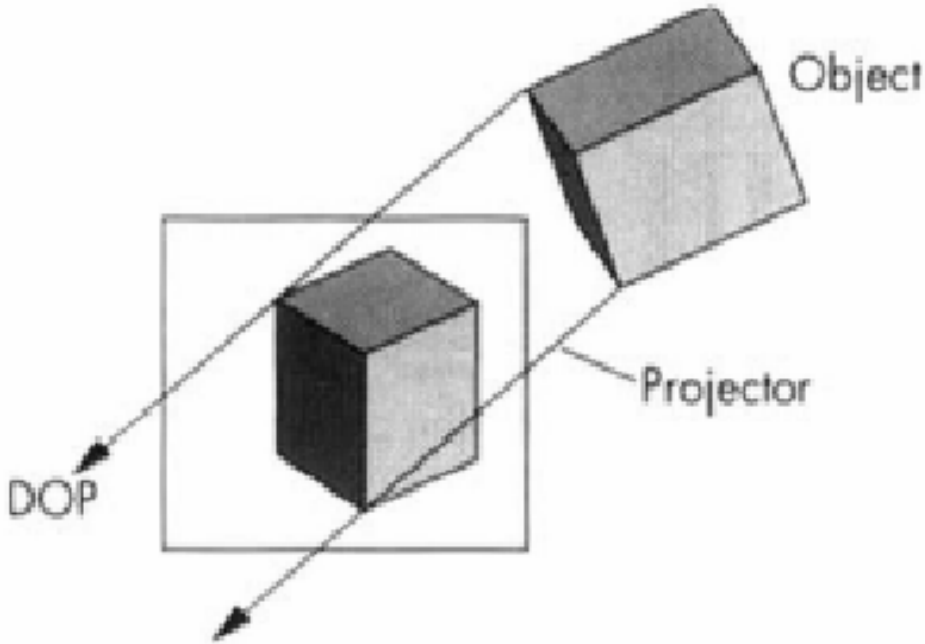
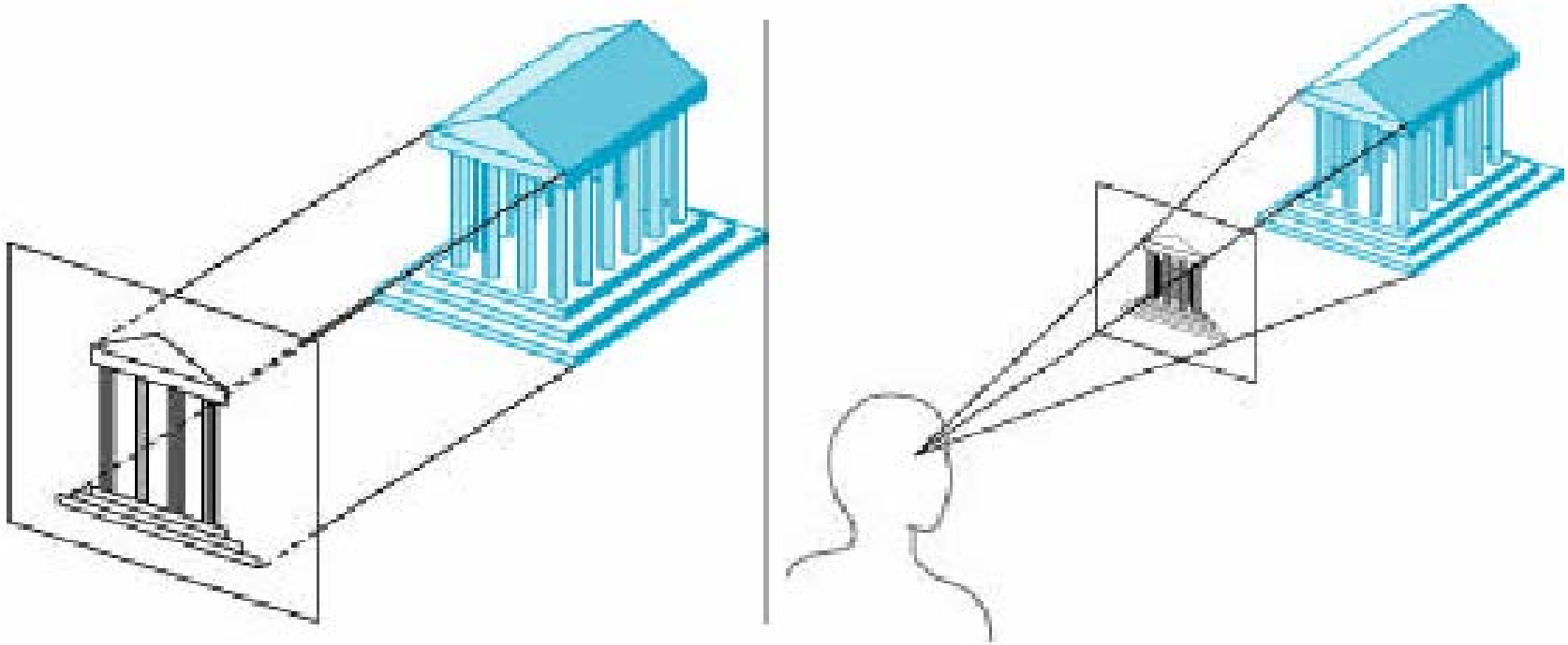


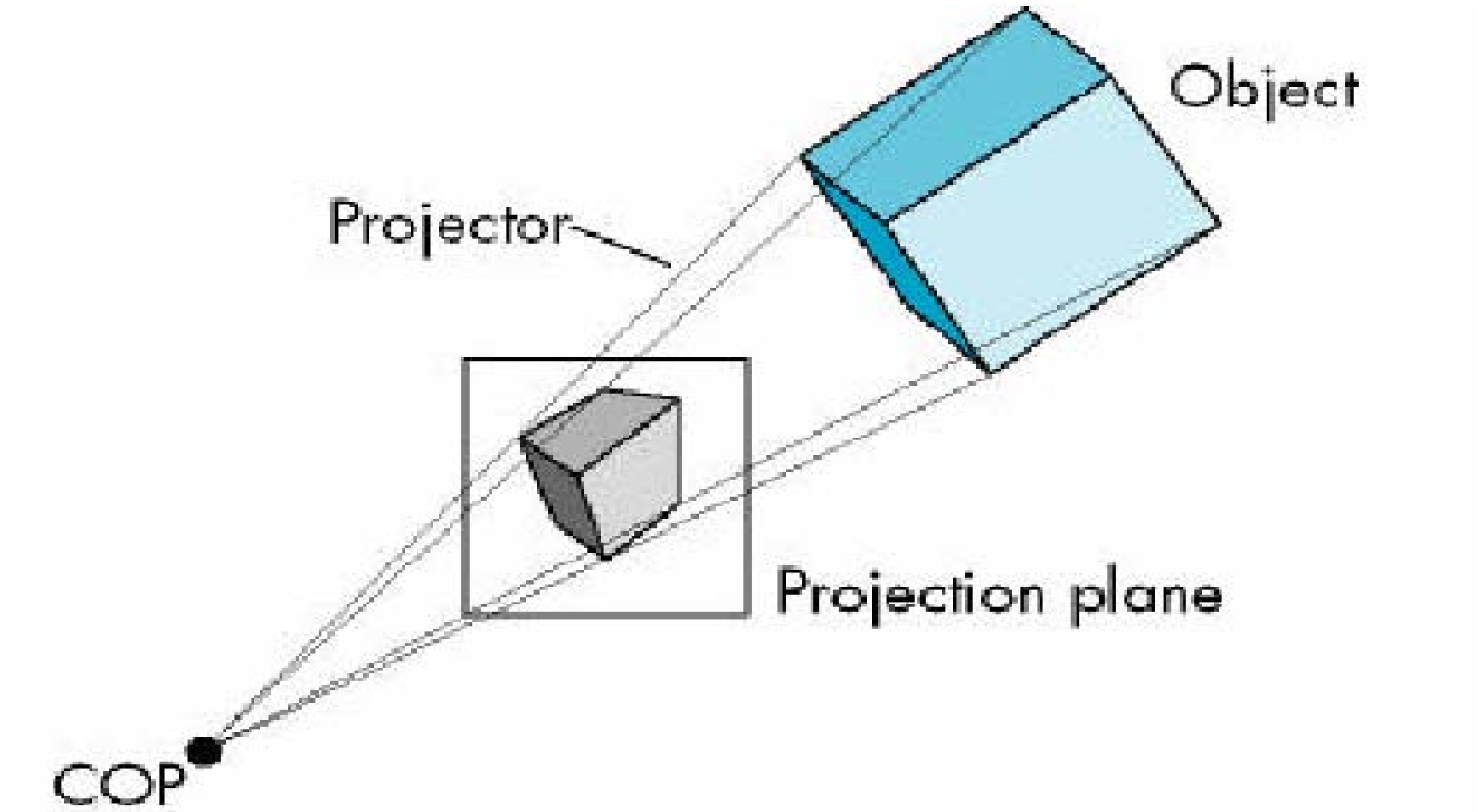
Figura 85 - Quando a distância do centro de projeção ao plano de projeção tende ao infinito temos projetores paralelos segundo uma *direção de projeção* (*DOP*).

Tipos de Projeções

- Dois tipos básicos
  - Paralela
  - Perspectiva



Projeção Perspectiva



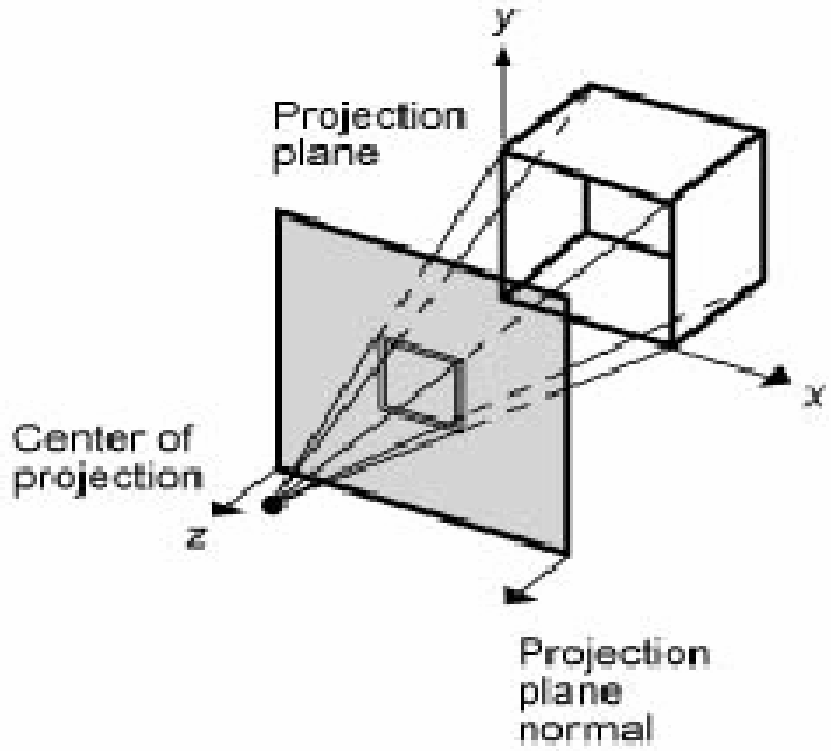
Projeção Perspectiva

- O centro de projeção está localizado a uma distância finita dos objetos e do plano de projeção.
- Uma projeção perspectiva apresenta projetores convergirem em um ponto, o ***centro de projeção. (figura a seguir)***

Projeção Perspectiva

- O centro de projeção está localizado a uma distância finita dos objetos e do plano de projeção.
- Uma projeção perspectiva apresenta projetores convergirem em um ponto, o ***centro de projeção. (figura a seguir)***
- A principal característica da projeção perspectiva é considerar a relação entre o tamanho da projeção de um objeto proporcional a distância objeto / observador.

Projeção Perspectiva



Formação da imagem em uma projeção perspectiva.

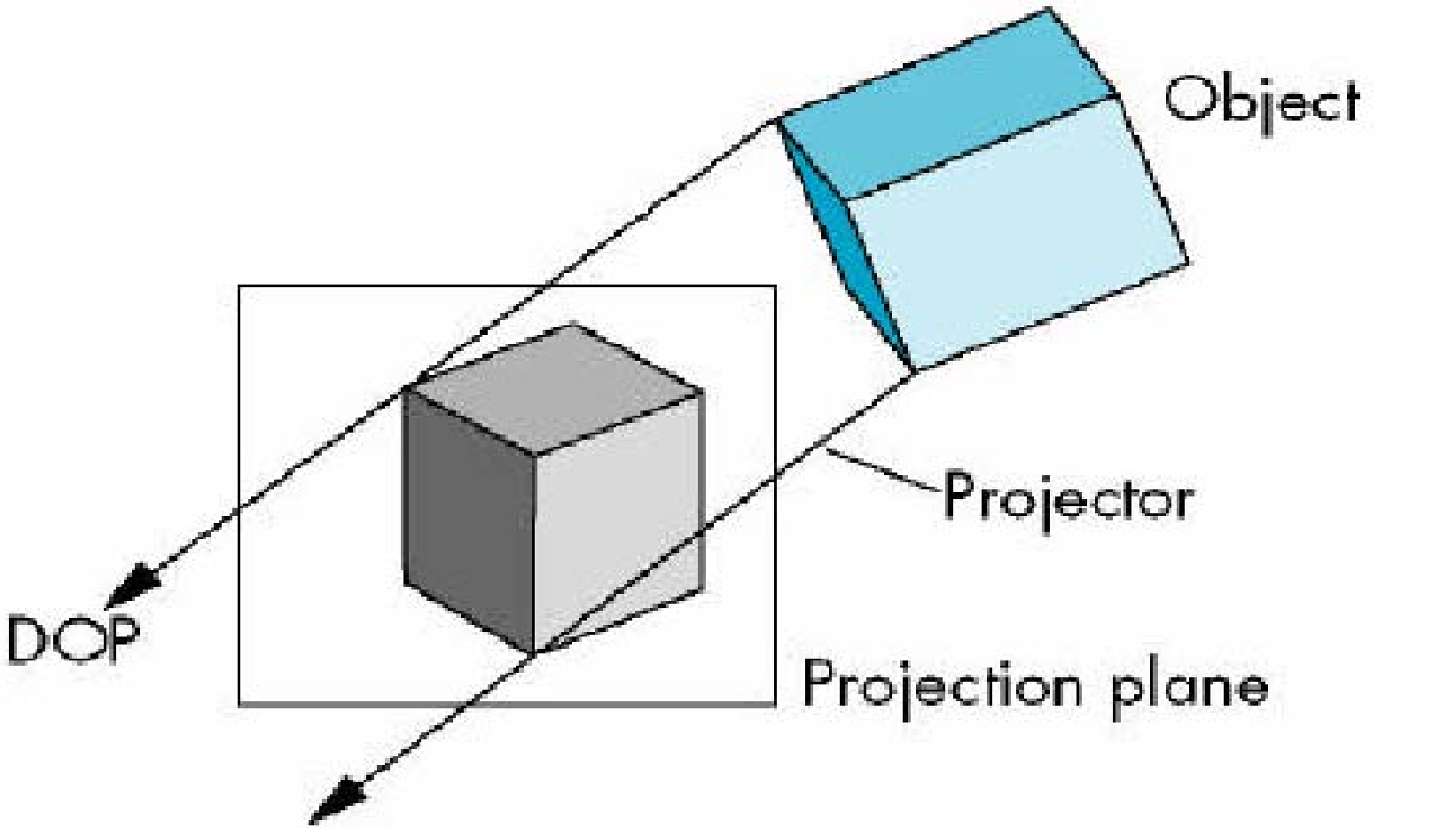
Projeção Perspectiva

- Características
  - Projetores se encontram no *Centro de Projeção*;
  - Preserva linhas;
  - Não preserva ângulos;
  - Não preserva medidas;
  - Equações não lineares;
  - Não inversível;
  - Mais realista
    - Tamanho da imagem inversamente proporcional a distancia do objeto ao plano de projeção

Projeção Perspectiva

- Utilizada em:
  - Propaganda
  - Apresentação de desenhos de arquetura, desenho industrial e engenharia
  - Arte final

Projeção Paralela



Projeção Paralela

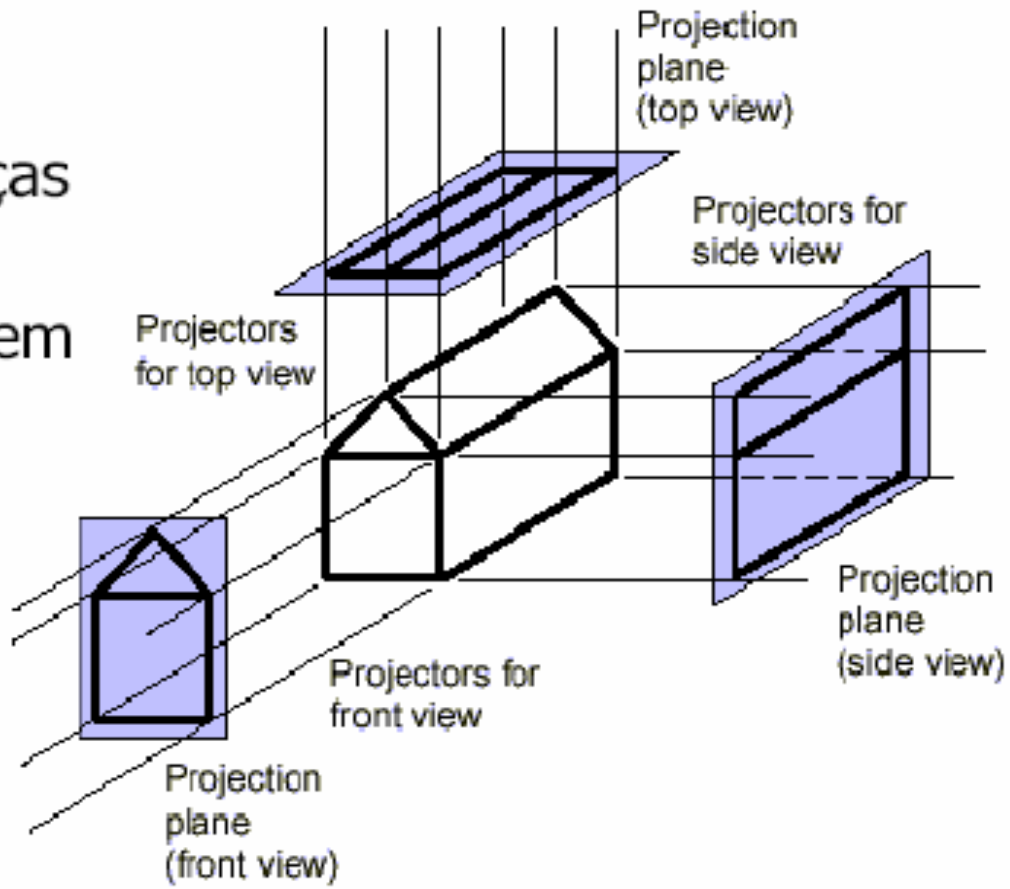
- Paralelas – considera-se uma direção de projeção, dada (normalmente) por um vetor unitário DOP – "Direction Of Projection" (dop_x,dop_y,dop_z).
- As projeções paralelas podem ser:
  - Ortográficas, quando a direção de projeção é perpendicular ao plano de projeção.
  - Oblíquas, quando a direção de projeção forma um angulo α com o plano de projeção.

Projeção Paralela

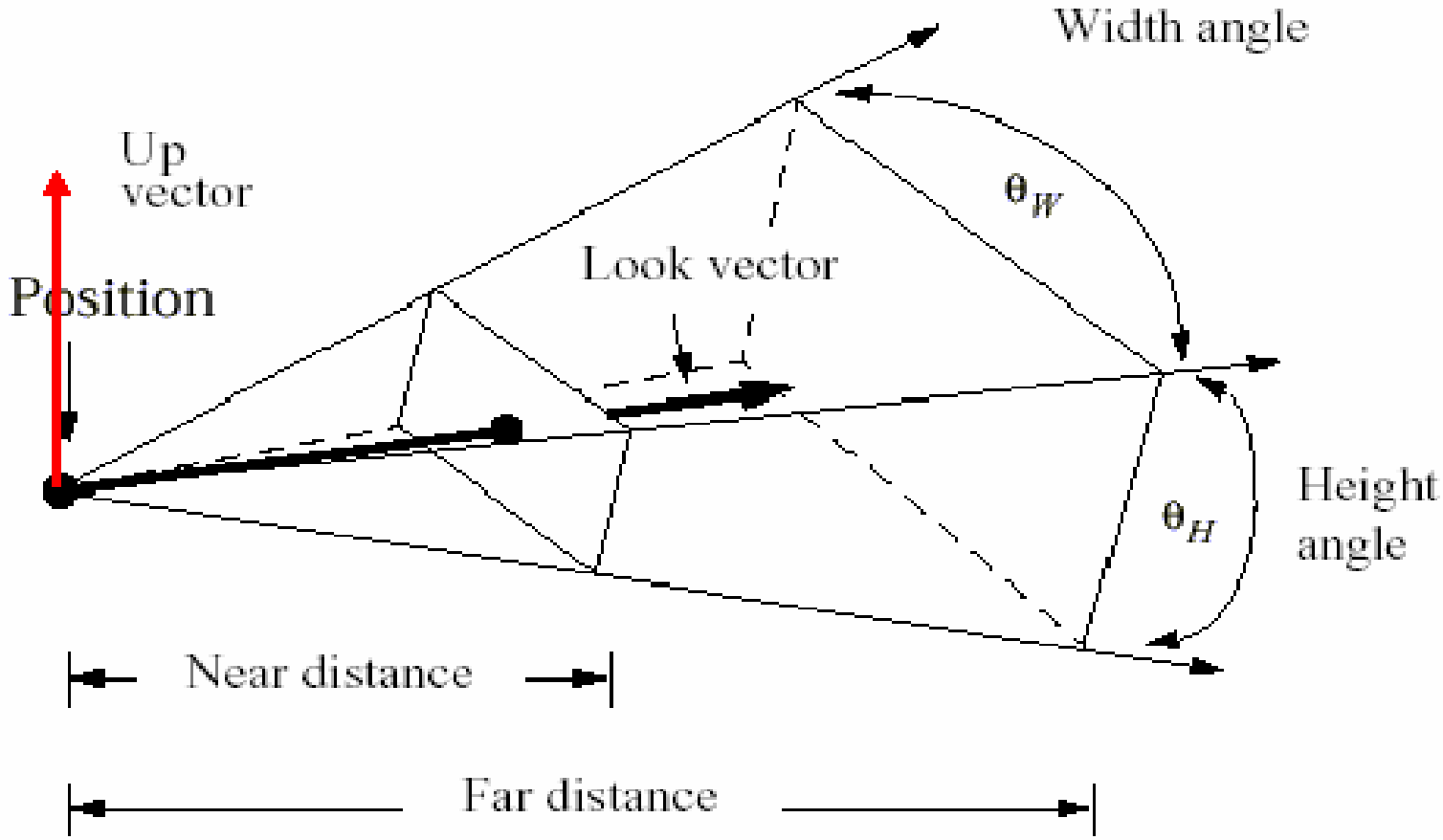
- Características
  - Projetores se encontram no infinito -> *Direção de Projeção*;
  - Preserva linhas;
  - Pode preservar angulos;
  - Pode preservas medidas.
  - Equações lineares;
  - Não inversível;
  - Resultado pouco realista.

Projeção Paralela

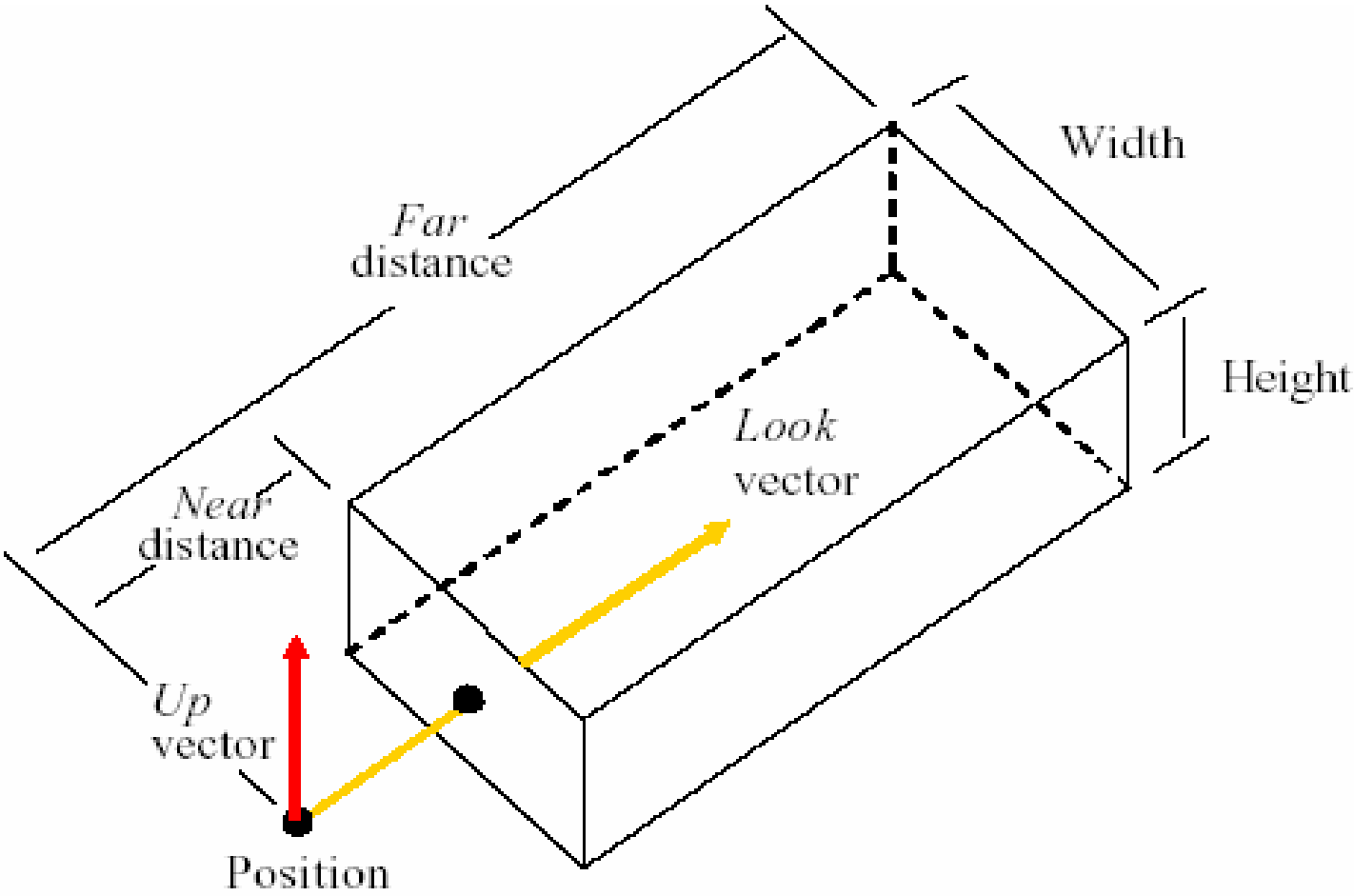
- Utilizada em:
  - Desenhos de maquinas e peças mecanicas
  - Plantas baixas em arquitetura



Resumo

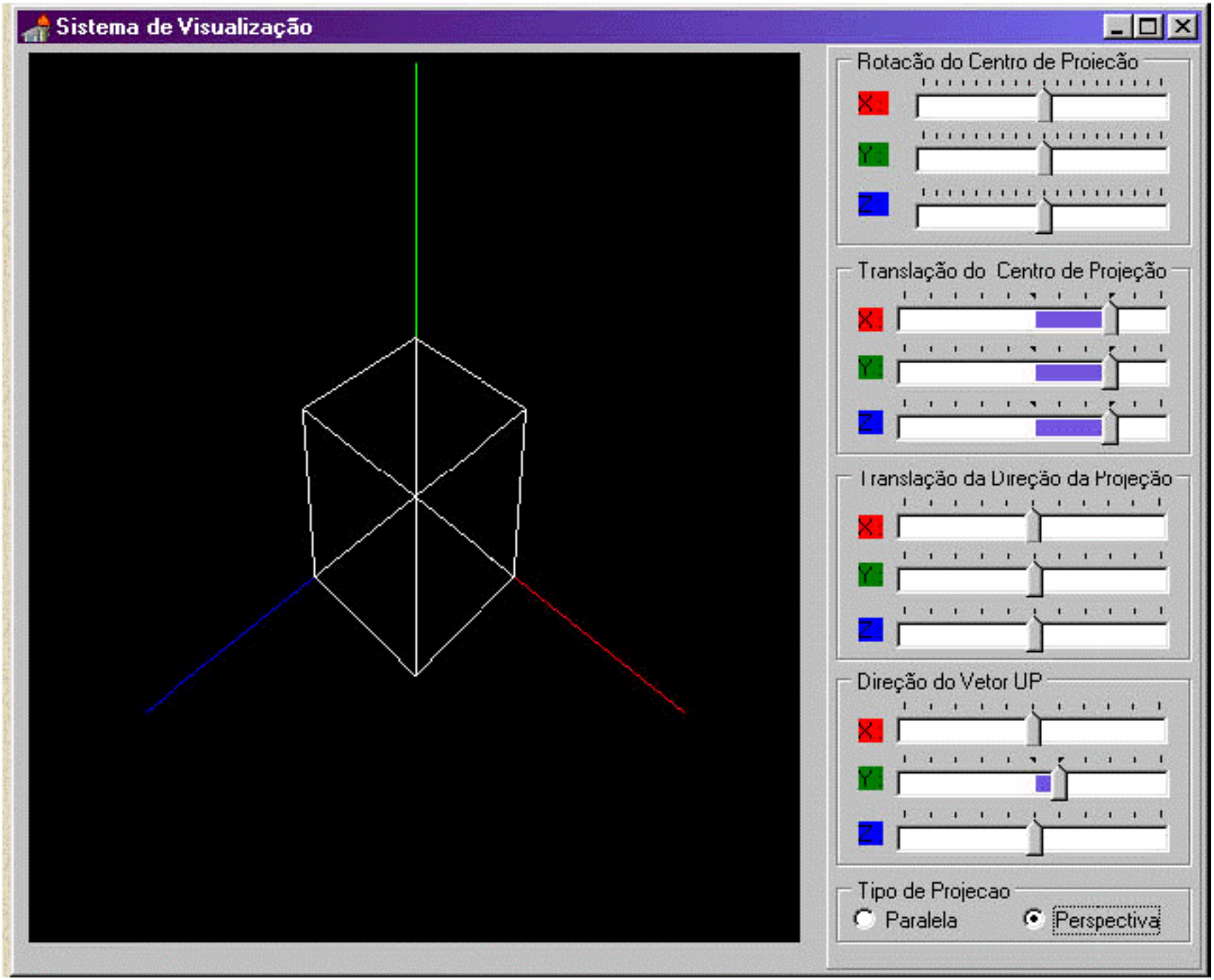

Width angle
Up
vector
Position
Look vector
θ_W
Height
angle
θ_H
Near distance
Far distance

Resumo


Width
Far
distance
Height
Look
vector
Near
distance
Up
vector
Position

Aplicação no OpenGL

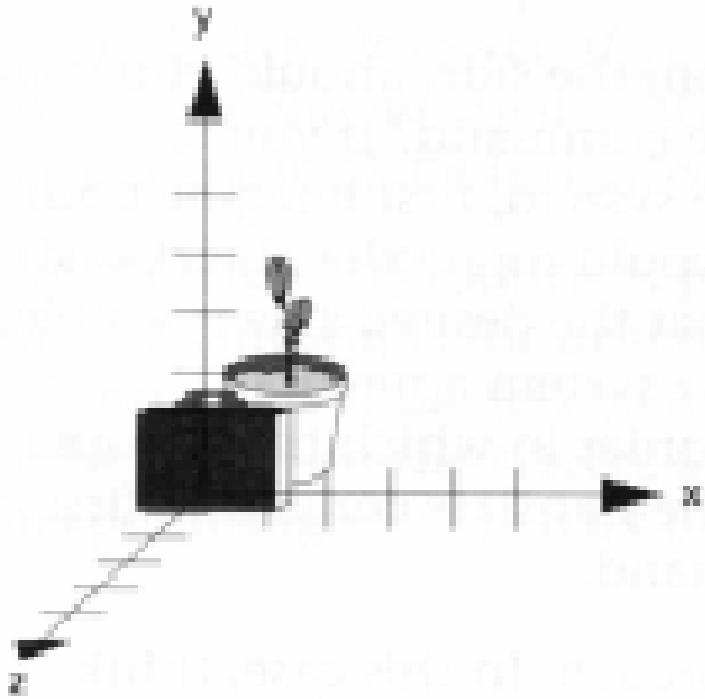
- Apresentaremos agora uma aplicação mostrando como a ***OpenGL*** implementa os conceitos de sistemas de visualização e projeções.
- O exemplo é bastante simples e apresenta a visualização de um cubo.
- Através da interface é possível não só alterar todos os parâmetros de posicionamento da câmera, como também definir o tipo de projeção que será utilizado (perspectiva ou paralela). A interface dessa aplicação pode ser vista na figura a seguir.


Sistema de Visualização
Rotacão do Centro de Proiecão
X:
Y:
Z:
Translação do Centro de Projeção
X:
Y:
Z:
Translação da Direção da Projeção
X:
Y:
Z:
Direção do Vetor UP
X:
Y:
Z:
Tipo de Projecao
Paralela
Perspectiva

Aplicação no OpenGL

- A camera virtual da ***OpenGL*** é posicionada, por *default,* na origem dos sistema de coordenadas global, com a direção de observação coincidente com o eixo ***z***, na direção negativa. A orientação da camera é paralela ao eixo ***Y*** positivo (como na figura a seguir).

Aplicação no OpenGL



- Posicionamento *default* da camera virtual na *OpenGL*.

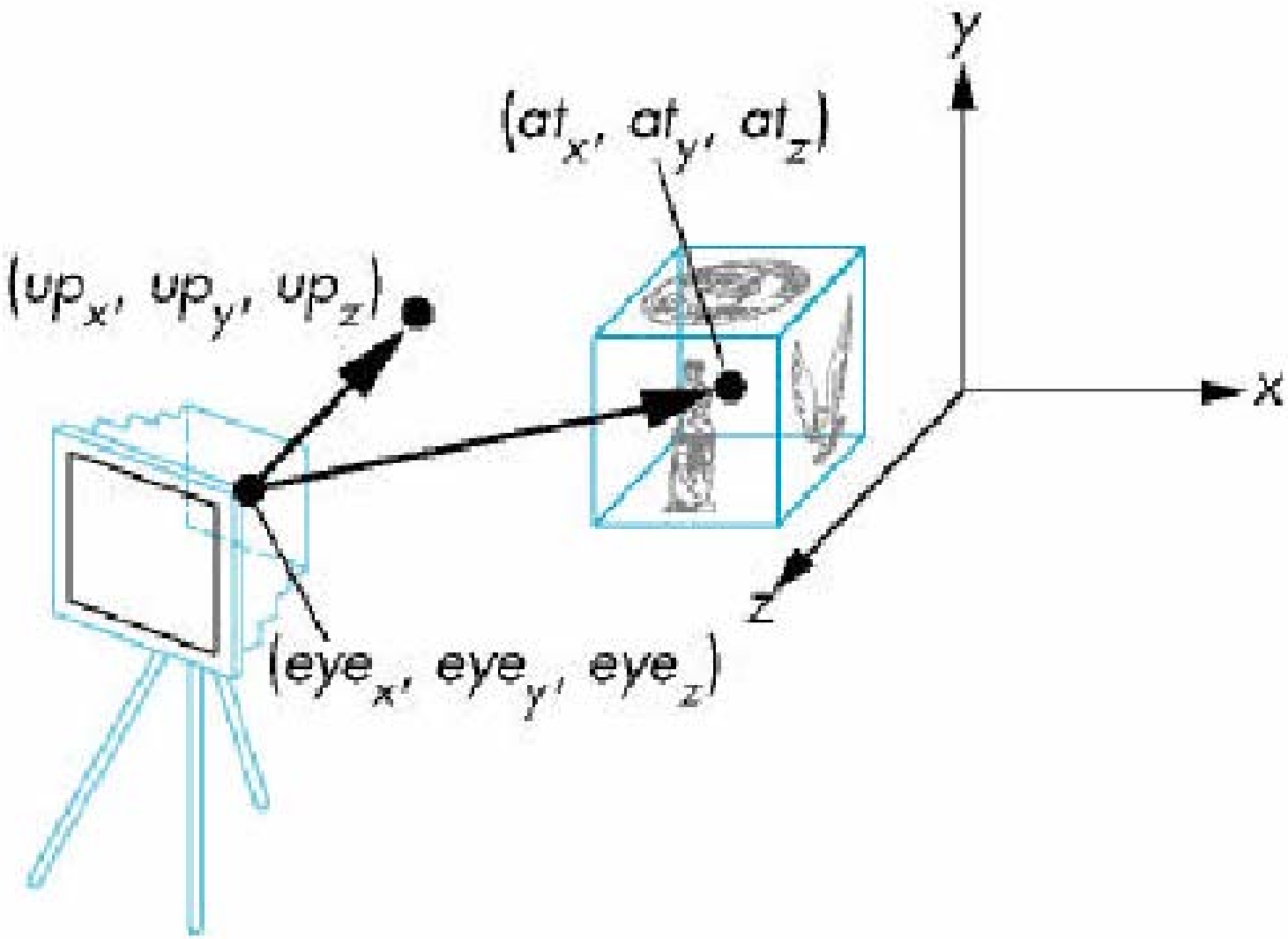
Aplicação no OpenGL

- Essa posição pode ser alterada de forma direta ou via transformações geométricas.
- Na primeira hipótese – utilizada parcialmente dentro do programa– utilizamos a função ***gluLookAt***.
- Ela recebe como parâmetros dois pontos no espaço 3D (respectivamente o centro de projeção (*eye*) e o ponto focal (*at*), que juntos definem o vetor *direção de observação*) e um vetor (*up*) (que determina a orientação da camera, associado ao ângulo de torção da camera). A figura a seguir mostra esses parâmetros.

Controle de Câmera

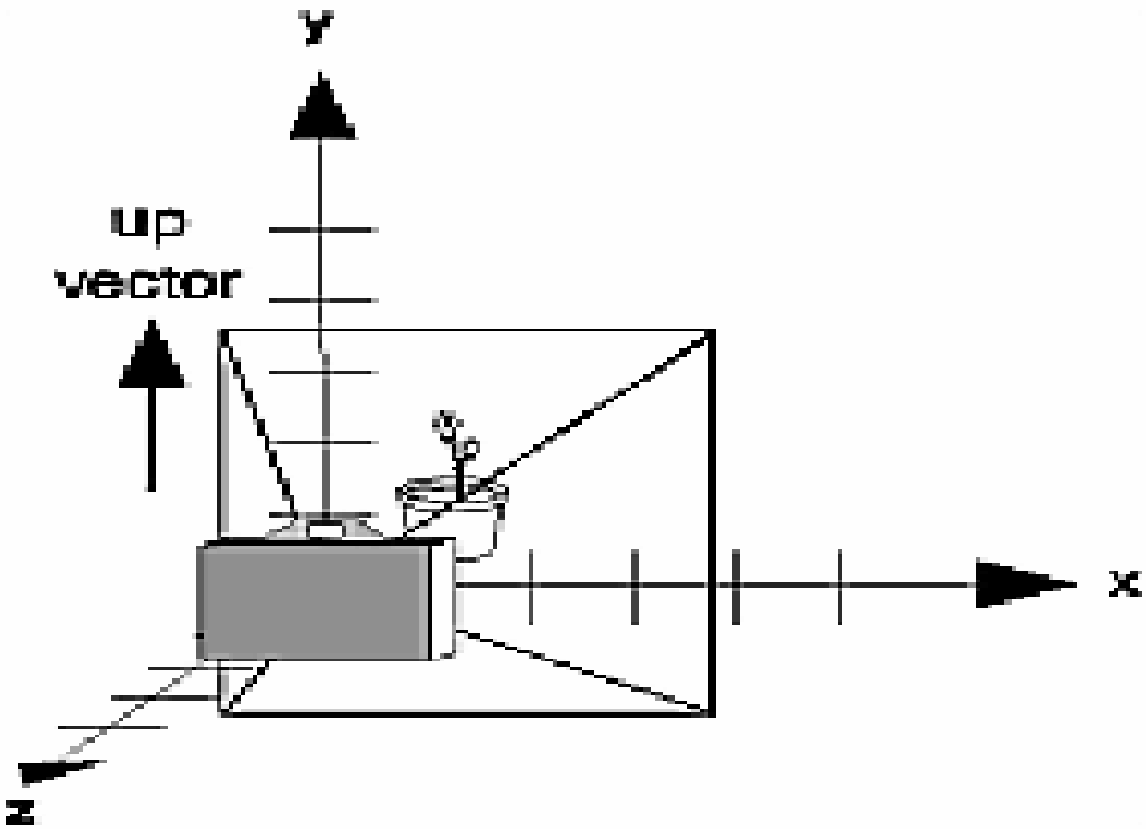
- void **gluLookAt**(GLdouble *eyex*,
  GLdouble eyey,
  GLdouble eyez,
  GLdouble centerx,
  GLdouble centery,
  GLdouble centerz,
  GLdouble upx,
  GLdouble upy,
  GLdouble upz);

Controle de Câmera



Controle de Câmera

- Posição default da Camera



Aplicação no OpenGL

- Na aplicação, a camera é inicialmente posicionada com centro de projeção no ponto (3,3,3) e com foco na origem (0,0,0). O vetor *up* é posicionado paralelo ao eixo ***Y*** positivo (0,1,0).
- Isso é feito através da inicialização das variáveis *PosCPCameraX, PosCPCameraY, PosCPCameraZ, PosDPCameraX, PosDPCameraY, PosDPCameraZ, VetorUpX, VetorUpY, VetorUpZ*, respectivamente, no evento *FormCreate*.
- Outra possibilidade é aplicar transformações geométrica a camera.
- Utilizamos essa abordagem para rodar a camera em torno do ponto focal. Os ângulos de rotação em cada direção, ***x***, ***y***, e ***z***, são definidos, respectivamente, pelas variáveis *RotacaoX*, *RotacaoY* e *RotacaoZ*.

Aplicação no OpenGL

- Relembrando : a ***OpenGL*** trabalha com três pilhas de transformações.
  - Uma delas está vinculada ao sistema de coordenadas global (*GL_MODELVIEW*),
  - Outra pilha está vinculado ao sistema de coordenadas da camera virtual – definido pela constante *GL_PROJECTION*.
- Dessa forma quando queremos aplicar transformações ao sistema de coordenadas da camera devemos direcionar essas transformações para a pilha correta.
- Isso é feito pela função ***glMatrixMode***.

Controle de Projeção

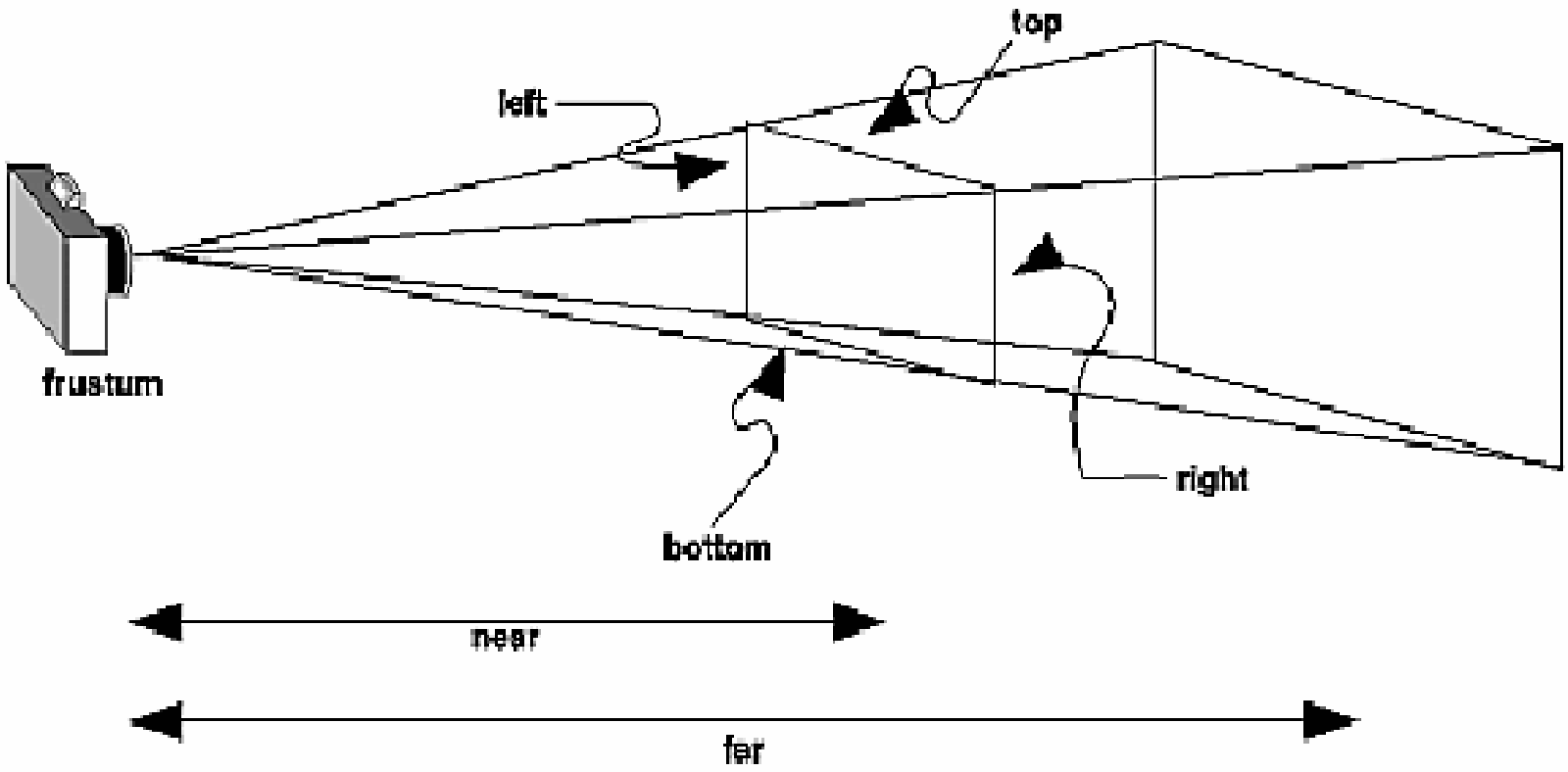
- O tipo de projeção a ser utilizada é definido a partir de duas funções :
- ***glOrtho*** para projeções ortográficas e ***glFrustrum*** para projeções perspectivas.
- No primeiro caso, a função ***glOrtho*** define uma projeção ortogonal e o
- volume de visão associado. Para tanto os parâmetros da função são valores limites para os planos *topo, fundo, esquerda, direita, perto* e *longe*.
- Para as projeções perspectivas a função ***glFrustrum*** é utilizada. Ela
- define, da mesma forma que ***glOrtho***, o volume de visualização associado, através dos parâmetros : *topo, fundo, esquerda, direita, perto* e *longe.*

Controle de Projeção

- Perspectiva
  - void **glFrustum**(GLdouble *left*,
    GLdouble *right*,
    GLdouble *bottom*,
    GLdouble *top*,
    GLdouble *near*,
    GLdouble *far*);

Controle de Projeção

- Perspectiva - glFrustum

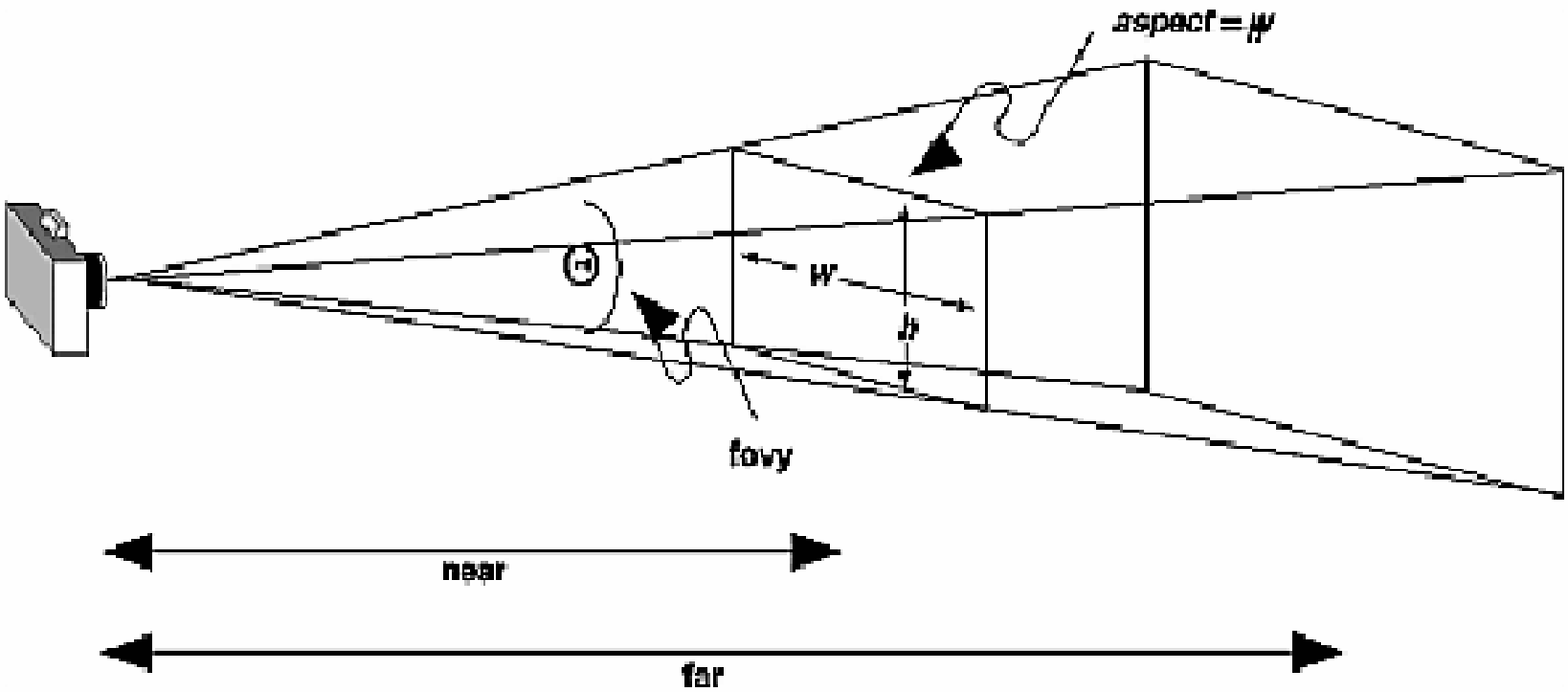


Controle de Projeção

- Perspectiva
  - void **gluPerspective**(GLdouble *fovy*,
    GLdouble *aspect*,
    GLdouble *near*,
    GLdouble *far*);

Controle de Projeção

- Perspectiva - gluPerspective

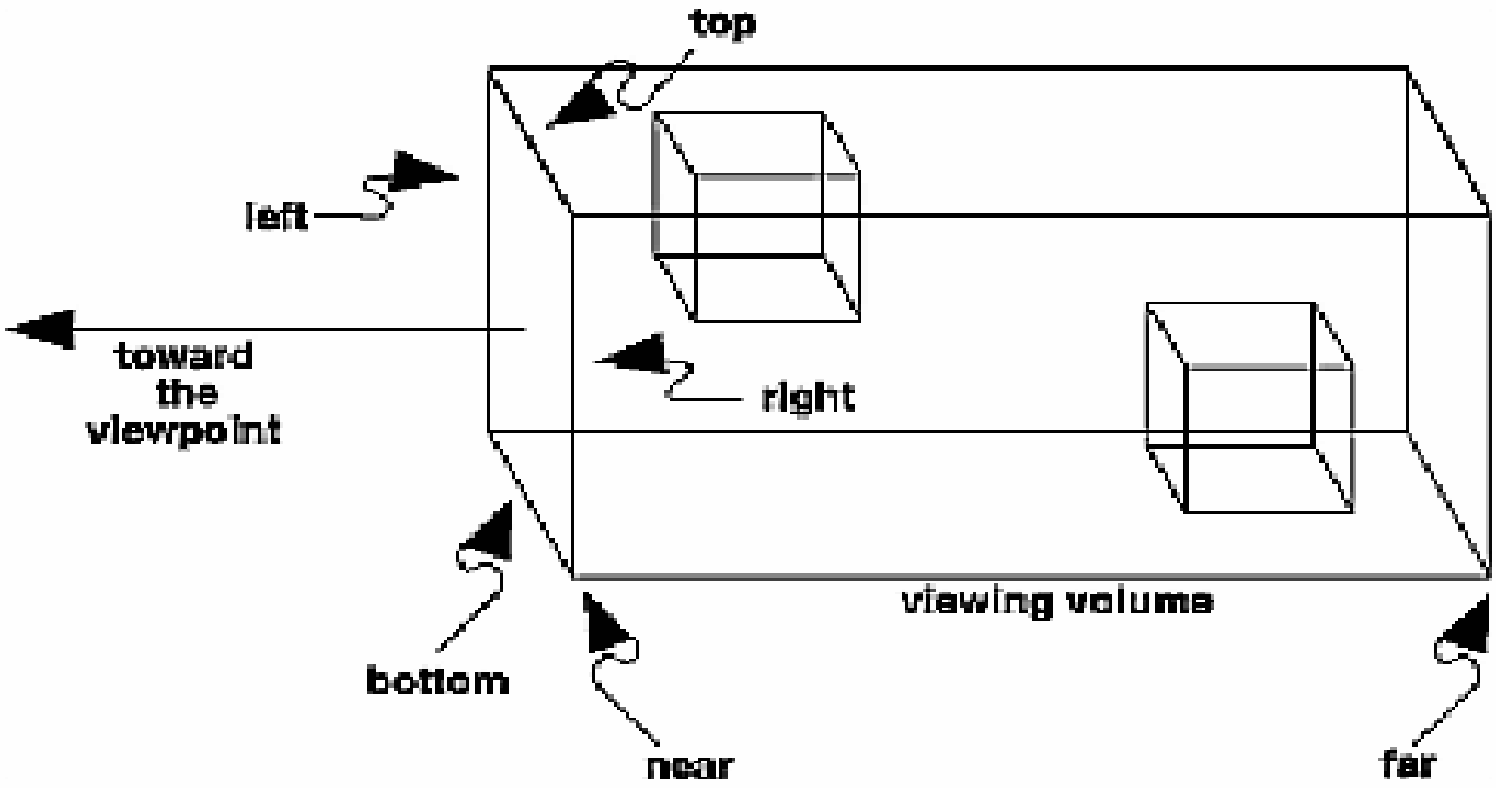


Controle de Projeção

- Paralela
  - void **glOrtho**(GLdouble *left*,
    GLdouble right,
    GLdouble bottom,
    GLdouble top,
    GLdouble near,
    GLdouble far);

Controle de Projeção

- Paralela - glOrtho



Controle de Projeção

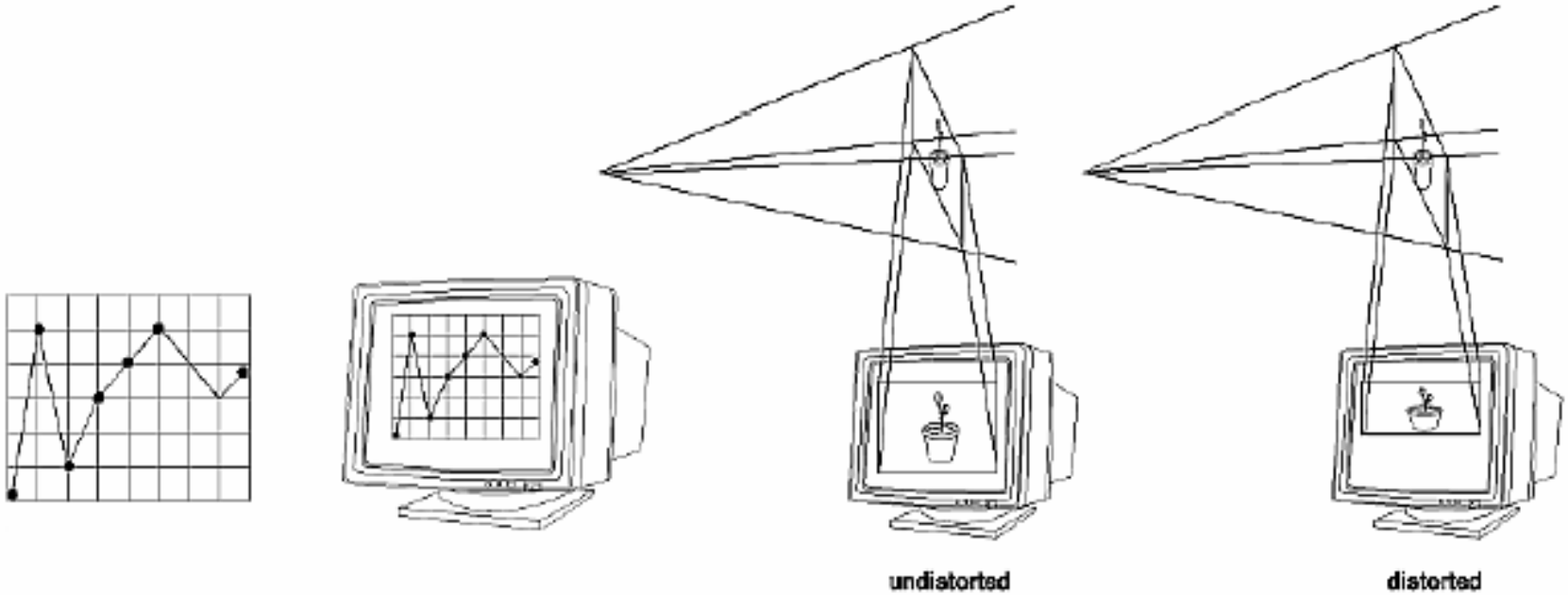
- Paralela 2D
  - void **glOrtho2D**(GLdouble *left*,
    GLdouble *right*,
    GLdouble *bottom*,
    GLdouble *top*);

Controle de Projeção

- Paralela 2D
  - void **glOrtho2D**(GLdouble *left*,
    GLdouble *right*,
    GLdouble *bottom*,
    GLdouble *top*);

Controle de Recorte 2D

- Mapeamento das coordenadas do mundo em coordenadas de Tela
  - Coordenadas Reais -> Coordenadas Inteiras



Definindo o Viewport

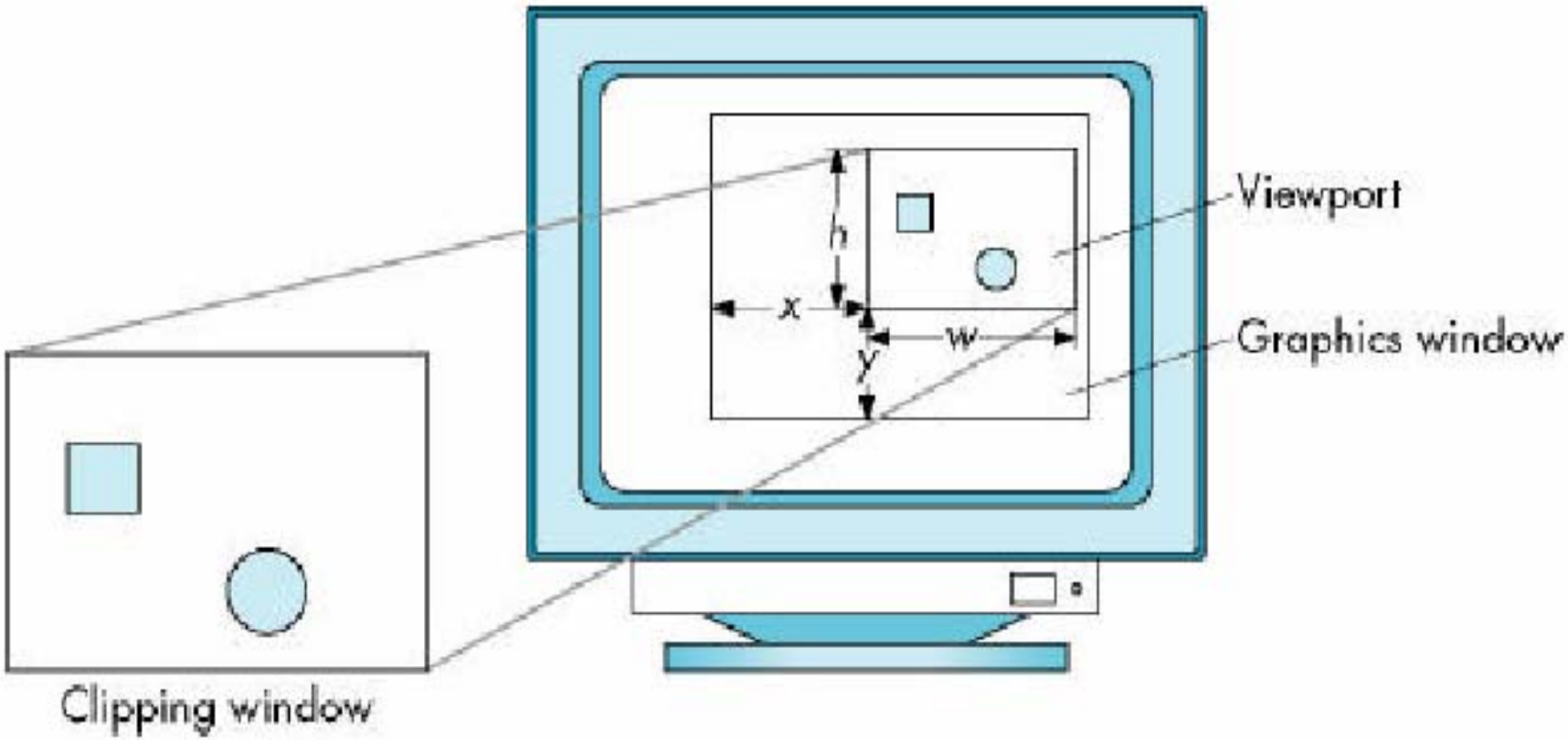
- Para especificar a área da janela na qual será mapeado o plano de projeção, utiliza-se_ *void* ***glViewport****(GLint x, GLint y, GLsizei width, GLsizei height);*
- O parâmetro (*x,y*) especifica o canto inferior esquerdo do *viewport* (janela de visualização), e os parâmetros *width* e *height* indicam o tamanho da *viewport*.
- Por *default*, os valores iniciais da *viewport* são (*0,0,winWidth, winHeiht*), onde o *winWidth* e o *winHeiht* são as dimensões da janela.

Controle de Recorte 2D

- void **glViewport**(GLint *x*,
  GLint *y*,
  GLsizei *width*,
  GLsizei *height*);

Controle de Recorte 2D

- **glViewport**



Exemplo

```
void reshape( int w, int h )
{
        glViewport( 0, 0, (GLsizei) w, (GLsizei) h );
        glMatrixMode( GL_PROJECTION );
        glLoadIdentity();
        gluPerspective( 65.0, (GLdouble) w / h,
                                1.0, 100.0 );
        glMatrixMode( GL_MODELVIEW );
        glLoadIdentity();
        gluLookAt( 0.0, 0.0, 5.0,
                        0.0, 0.0, 0.0,
                         0.0, 1.0, 0.0 );
}
```

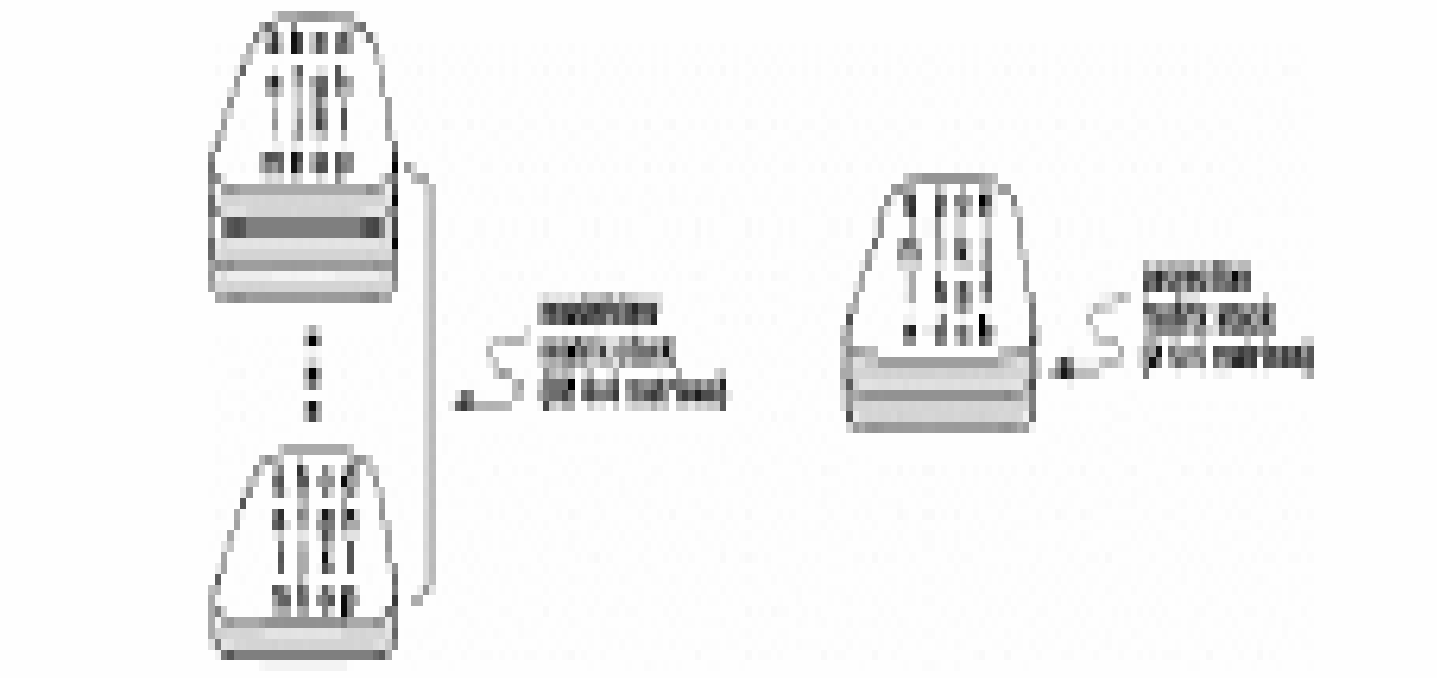
Matriz Identidade

- Para permitir que a transformação atual seja reinicializada há o comando
  - *glLoadIdenty();*
- Este comando inicia a matriz corrente com a matriz identidade.

Pilha de Matrizes

- *O OpenGL utiliza uma estrutura de pilha para armazenar várias matrizes de transformação.*
- *No modo GL_MODELVIEW essa pilha possui 32 matrizes.*
- *No modo GL_PROJECTION a pilha possui somente 2 matrizes*

Pilha de Matrizes



Pilha de Matrizes

- A principal motivação para o uso de uma pilha de matrizes é a possibilidade de "salvar" uma determinada configuração de transformações para posterior reutilização.
- A pilha pode ser manipulada por meio de duas funções básicas :
  - _ ***glPushMatrix();***
  - _ ***glPopMatrix();***

Pilha de Matrizes