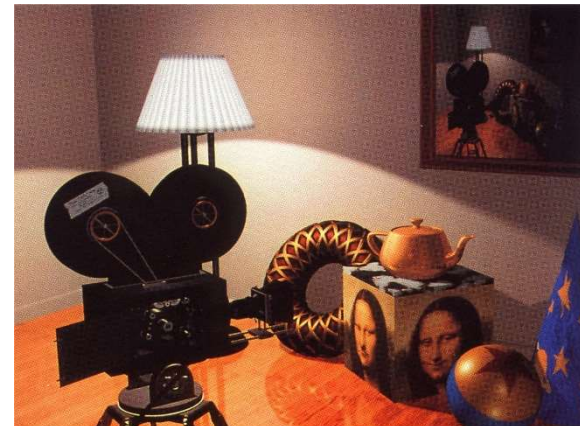
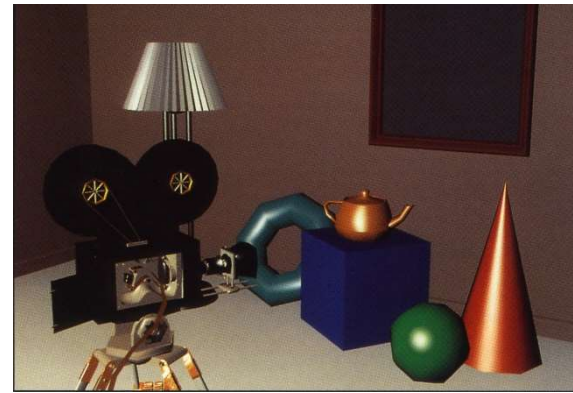
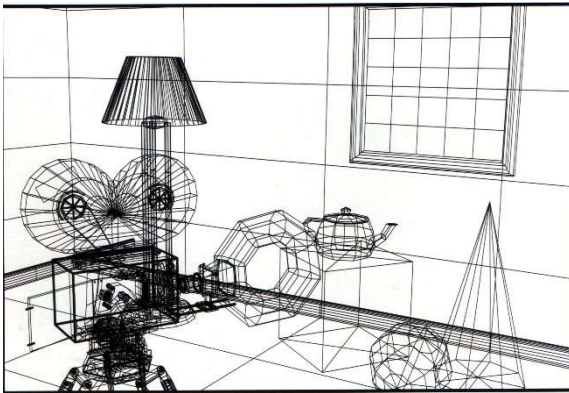


Síntese de Imagem Cálculo de Visibilidade

Sistemas Gráficos/ Computação Gráfica e Interfaces

Síntese de Imagem

A **síntese de imagem** (do inglês **rendering**) consiste na criação de imagens com elevado grau de realismo a partir da descrição dos objectos nela contidos (geometria e interacção com a luz), fontes de luz e posicionamento do observador.



Síntese de Imagem

Índice

1. Modelos de Iluminação

- a) Modelo Elementar
- b) Modelo de Phong
- c) Modelo Melhorado (atenuação com distância, atenuação atmosférica...)

2. Melhorando a imagem

Sombreamento (shading, smooth shading)
Mapeamento de Texturas; “Bump Textures”
“Antialiasing”

3. Projecção de Sombras

4. Cálculo de Visibilidade (este capítulo)

- a) Algoritmos no espaço imagem
- b) Algoritmos no espaço objecto
- c) Algoritmo tipo Lista de Prioridades

5. Algoritmos de Iluminação Global

Síntese de Imagem

- Cálculo de Visibilidade -

Objectivo: a partir de um conjunto de objectos 3D, determinar quais as linhas ou superfícies dos objectos que são visíveis, quer a partir do centro de projecção (para projecção em perspectiva) quer ao longo da direcção de projecção (para projecção paralela), de modo a mostrar apenas as linhas ou superfícies visíveis.

Duas aproximações possíveis:

1. Algoritmos no **ESPAÇO IMAGEM**:

Para cada **pixel** da imagem determinar qual o objecto visível

```
for (cada pixel na imagem)
    determinar o objecto mais perto do observador, atendendo
    aos raios de projecção;
    desenhar o pixel com a cor apropriada;
```

1. Algoritmos no **ESPAÇO OBJETO**:

Comparar os **objectos** entre si de modo a seleccionar a parte visível de cada um

```
for (cada objecto do "mundo")
    determinar as partes do objecto não obstruídas por ele ou
    por outros objectos;
    desenhar as partes visíveis na cor apropriada;
```

Síntese de Imagem

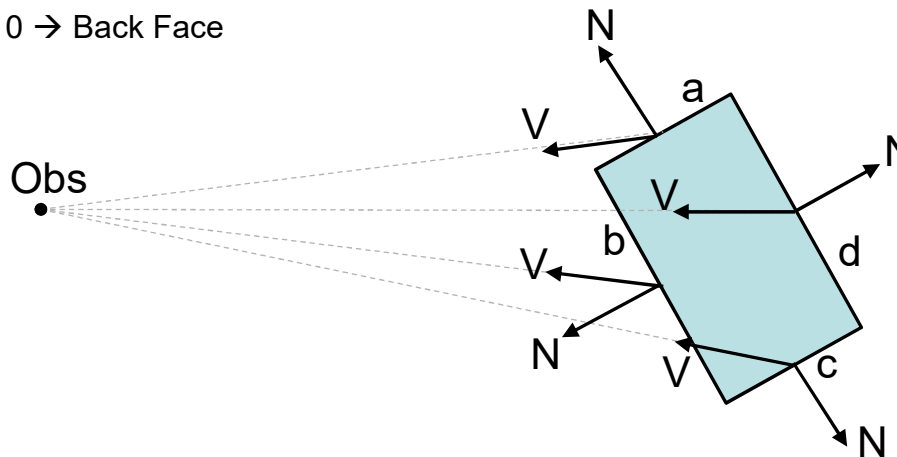
- Cálculo de Visibilidade -

- *Backface culling*

- Técnica usada para reduzir o número de polígonos a processar

- *Front Faces* (ex: a, b): Enviar para cálculo de visibilidade;
- *Back Faces* (ex: c, d): Não são visíveis → desnecessário enviar.

- $N \cdot V = |N| \cdot |V| \cdot \cos(\alpha)$
 - Resultado $> 0 \rightarrow$ Front Face
 - Resultado $< 0 \rightarrow$ Back Face



- Redução:

- $\text{Número de Front Faces} \approx \frac{1}{2} \text{ Número total de Faces}$

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço objecto

Algoritmos no espaço objecto para determinação de linhas visíveis:

Ao volume: Algoritmo de Roberts

À aresta: Algoritmo de Appel, Loutrel, Galimberti e Montanari

Nestes algoritmos todas as arestas são testadas para produzir uma lista com os segmentos visíveis de todas as arestas.

Ao Volume: supõe-se que uma aresta pode ser oculta pelo volume de um objecto.

À Aresta: o teste é efectuado aresta contra aresta observando que a visibilidade de uma aresta goza de coerência, o que permite determinar a invisibilidade de uma aresta a partir da invisibilidade de outra aresta que possua com ela um vértice comum.

Coerência de aresta: uma aresta só altera a sua visibilidade onde se cruza por trás de uma aresta visível.

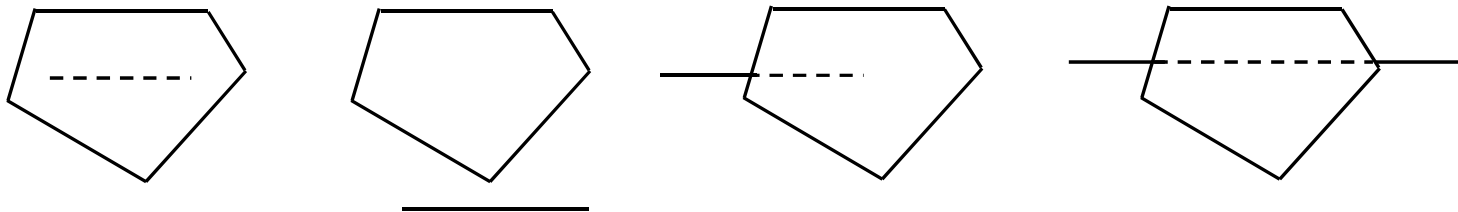
Síntese de Imagem

- Cálculo de Visibilidade – Alg. espaço objecto

Ao Volume - Algoritmo de Roberts (1963)

Requisito: cada aresta deve pertencer a uma face de um **poliedro** convexo. Poliedros côncavos devem ser partidos em vários convexos para poder aplicar o algoritmo.

1. Remover todas as faces posteriores dos objectos (*backface culling*) e correspondentes arestas
2. Comparar as arestas restantes contra cada **volume** (poliedro) da cena; deste teste podem ocorrer 4 situações:



- Aresta completamente oculta pelo volume.
- Aresta não oculta
- Uma parte da aresta não é oculta
- Duas partes da aresta não são ocultas

Síntese de Imagem

- Cálculo de Visibilidade – Alg. espaço objecto

À aresta - Algoritmo de Appel, Loutrel, Galimberti e Montanari (1967/9,1970)

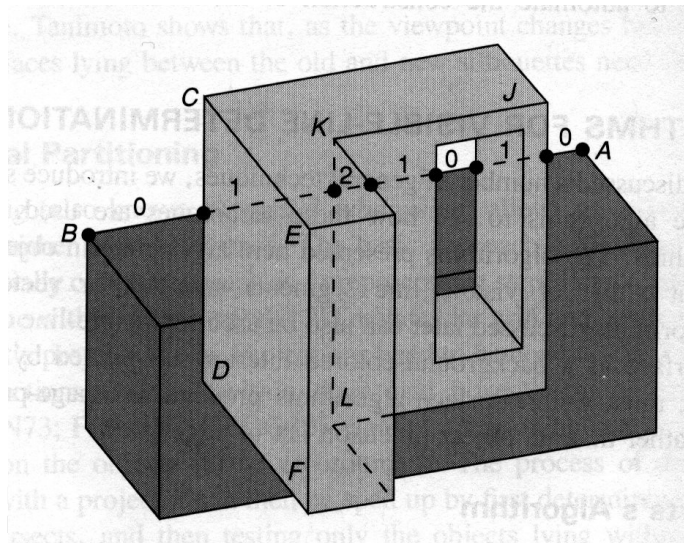
Ao contrario do alg. de Roberts trata ao nível do polígono.

1. Determinar as faces orientadas para o observador (*backface culling*).
2. Calcular a “**Quantitative Invisibility**” de um vértice para cada objecto.

“**Quantitative Invisibility**” **QI** de um ponto: é o número de polígonos entre o observador e o próprio ponto.

Quando uma aresta passa por detrás de um polígono, a sua **QI** é incrementada de 1, e quando deixa de ser ocultada é decrementada de 1.

- Quando se chega ao vértice final de uma aresta, o valor **QI** desse vértice é o valor inicial de **QI** nas as arestas que se iniciam nesse vértice.



Contour line: é definida como uma aresta partilhada por um polígono *back-facing* com outro front-facing, ou um polígono *front-facing* isolado.

Contour lines: AB, CD, DF, KL

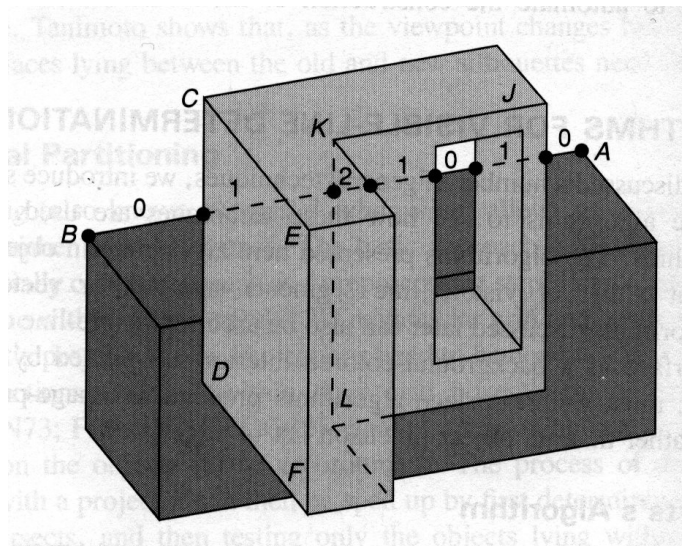
CE, EF, JK não são (porque são partilhadas por polígonos *front-facing*)

Síntese de Imagem

- Cálculo de Visibilidade – Alg. espaço objecto

O **QI** é alterado quando a aresta passa por trás de uma *contour line*.

Na figura os pontos indicam as intersecções da projecção da aresta AB com as projecções das *contour lines*.



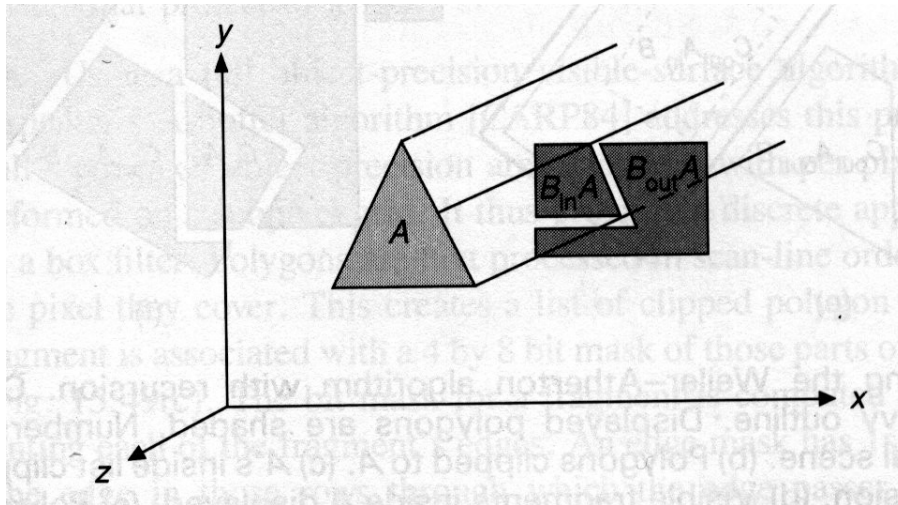
No final apenas os segmentos com valor **QI** igual a **zero** são visíveis.

Síntese de Imagem

- Cálculo de Visibilidade – Alg. espaço objecto

Determinação de faces visíveis: **Atherton & Weiller (1977)**

Requisito: exige a aplicação de um algoritmo sofisticado de *clipping* de polígonos, capaz de efectuar *clipping* de um polígono concavo com buracos contra um outro qualquer.



Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço objecto - **Atherton & Weiller**

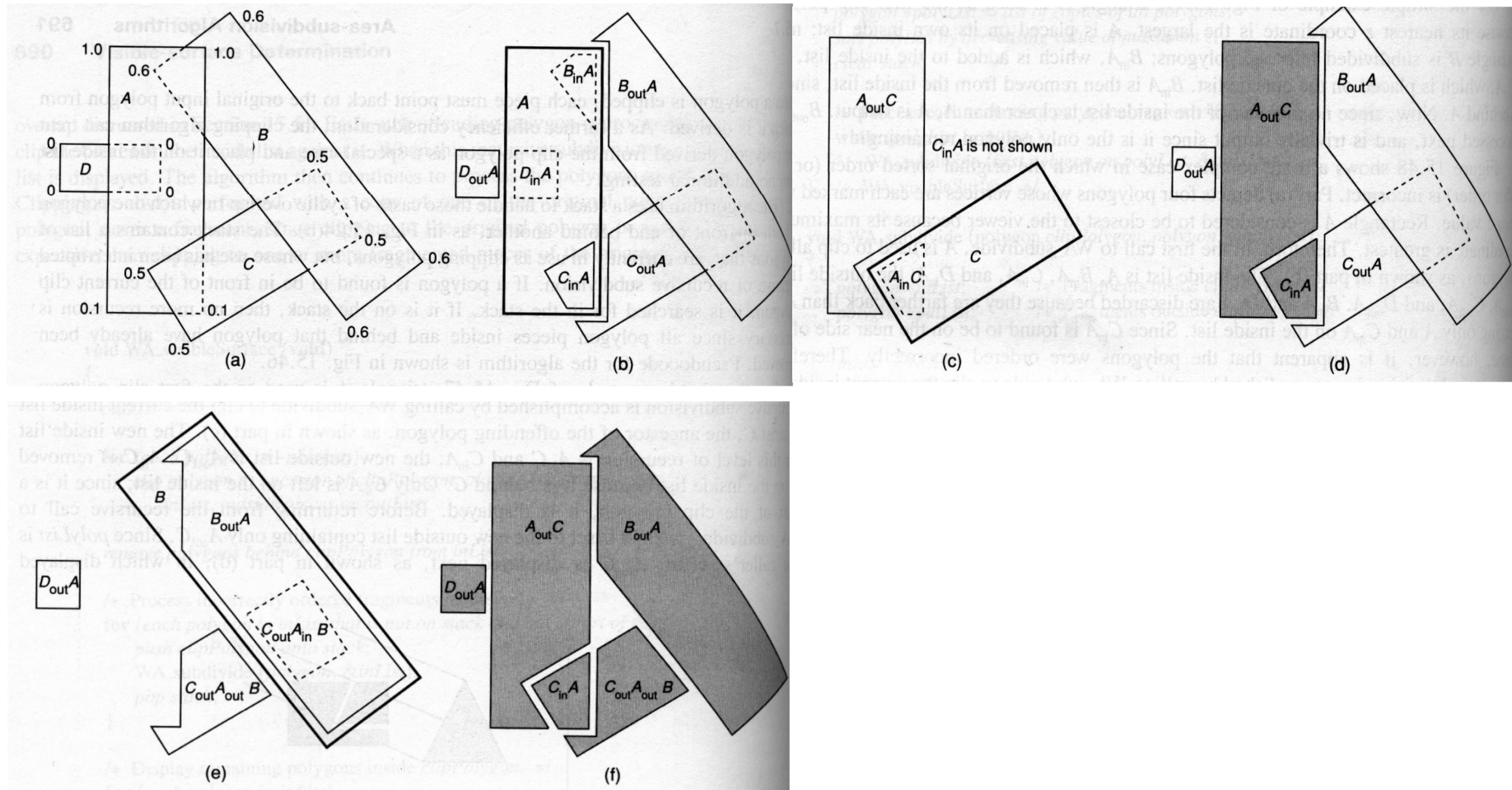
Procedimento:

1. Ordenar os polígonos pela coordenada **Z**; o polígono mais próximo do observador é seleccionado como janela de corte.
2. O polígono é usado para efectuar *clipping* dos restantes, resultando **duas listas** contendo os polígonos (ou parte deles) que são **interiores (invisíveis)** e **exteriores (visíveis)** à região de *clipping*.
3. Os polígonos da lista interior são marcados como invisíveis.
.
4. Passa ao polígono seguinte da lista de exteriores e regressa ao ponto 2

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço objecto - **Atherton & Weiller**

Na figura os valores indicam a coordenada Z de cada vértice.



Síntese de Imagem

- Cálculo de Visibilidade – Alg. espaço imagem

1. Orientada à área: **Algoritmo de Warnock** (1969)
2. Orientado à linha: **Linha de Varrimento / Watkins**
3. Orientado ao pixel: **Z-buffer, Ray Casting**

Algoritmo de Warnock

- O algoritmo divide sucessivamente a imagem projectada em áreas rectangulares.
- Se uma área é considerada coerente,
 - A área é desenhada com os polígonos que contém
 - **SENÃO**, a área é dividida em áreas menores; o procedimento é aplicado recursivamente.
- Quanto menores forem as áreas, menor número de polígonos estarão sobrepostos nessas áreas; é mais fácil se poderá decidir qual o polígono a desenhar.

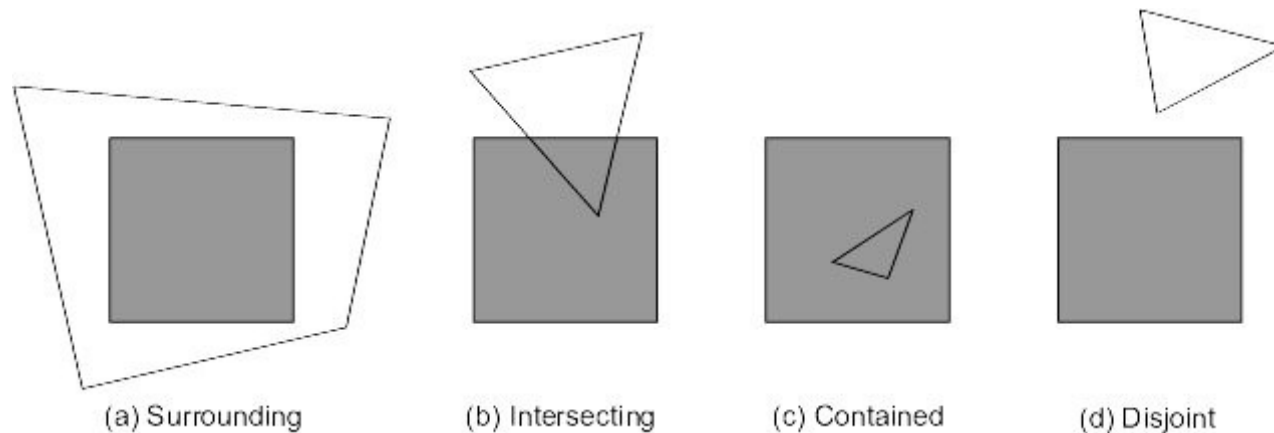
O algoritmo utiliza o conceito de coerência de área: um grupo de pixels adjacentes é habitualmente coberto pela mesma face visível.

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: alg. de Warnock

Procedimento alg. Warnock:

- Divisão da área em 4 blocos iguais. Em cada fase da subdivisão, a projecção de cada polígono estará em uma das 4 situações em relação a cada área:

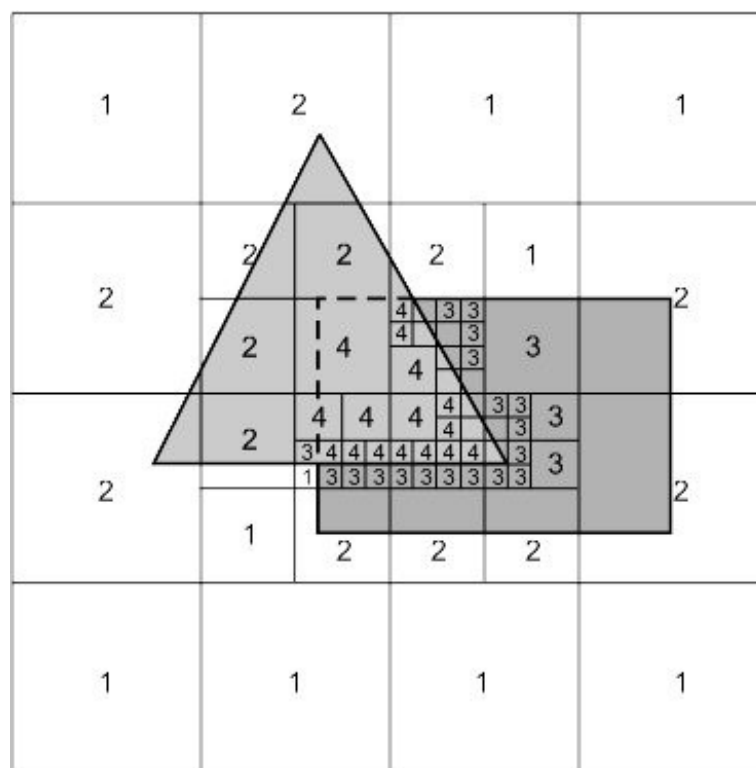


As situações em que a decisão é possível, não havendo mais subdivisão:

- a)**- Apenas um polígono que ocupa toda a área, não havendo mais nenhum projetado nessa área. Pintar a área com a cor desse polígono.
- b, c)**- Apenas um polígono que intersecta ou que está totalmente dentro da área. Preencher a área com a cor de fundo e depois pintar a parte do polígono que se encontra nessa área.
- d)**- Todos os polígonos estão fora da área → Pintar a área à cor de fundo.

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: alg. de Warnock



Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: linha de varrimento

Algoritmo de Linha de Varrimento

A imagem é criada linha a linha, à semelhança do algoritmo de preenchimento de regiões 2D, designado por **algoritmo da lista das arestas activas**.

Conceitos explorados:

- **Coerência vertical:** o conjunto de objetos visíveis determinados para uma linha de varrimento, difere pouco do conjunto correspondente da linha anterior.
- **Coerência de aresta:** uma aresta só altera a sua visibilidade quando se cruza com outra aresta visível ou quando penetra uma face.

Estruturas de dados utilizadas:

AEL - Lista de Arestas Activas

ET - Tabela de (novas) Arestas

PT - Tabela de Polígonos

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: linha de varrimento

(nota: descritores com mais informação do que consta no livro, a vermelho)

Tabela de Arestas (ET): guarda informação de todas as arestas cuja projecção no plano de visualização não é horizontal. As entradas da tabela estão ordenadas de forma crescente pelo menor valor de Y, e contêm inicialmente:

1. Coordenadas (X, **Z**) do vértice com menor Y
2. Altura da aresta (Y1-Y0) ou, em alternativa, Ymax
3. Incrementos $\Delta X/\Delta Y = \partial X/\partial Y$, **$\Delta Z/\Delta Y = \partial Z/\partial Y$** , usados na actualização de X e de Z, na passagem para a linha de varrimento seguinte
4. Identificação do **(s)** polígono **(s)** partilhados pela aresta

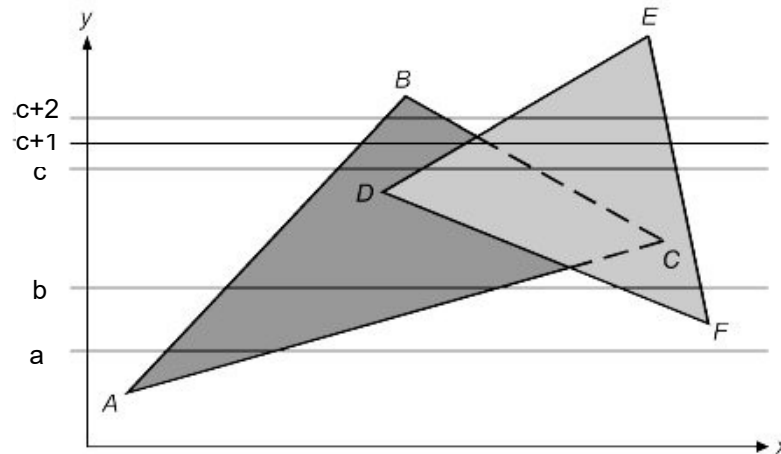
Tabela de Polígonos (PT): informação de todos os polígonos, contendo para cada um:

1. Coeficientes da equação do plano (**no mínimo, $\Delta Z/\Delta X = \partial Z/\partial X$**)
2. Informação da cor
3. Coordenada **Z**, a recalculer a cada pixel
4. Flag de *in-out*, inicializada a *False*, é usada para controlar se o processamento está dentro ou fora do polígono

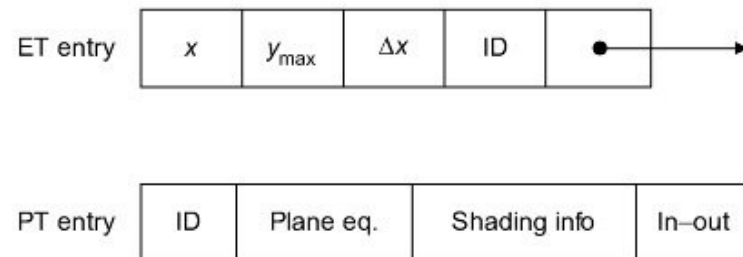
Lista de Arestas Activas (AEL): informação de quais as arestas ativas na linha de varrimento actual.

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: linha de varrimento



Quando se verifica a sobreposição de polígonos, como na linha **c**, mais do que um polígono possuem a flag **in-out** a **true**. Utilizando a equação do plano de cada polígono, determina-se a coordenada **Z** de cada um para saber qual deles está mais próximo do observador.



AET contents				
Scan line	Entries			
a	AB	AC		
b	AB	AC	FD	FE
c, c+1	AB	DE	CB	FE
c+2	AB	CB	DE	FE

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: linha de varrimento

- Actualização incremental de uma aresta à mudança de uma linha de varrimento:

$$\left. \begin{aligned} x &= x + \Delta y \cdot \left(\frac{\partial x}{\partial y} \right) = x + \left(\frac{\partial x}{\partial y} \right) \\ z &= z + \Delta y \cdot \left(\frac{\partial z}{\partial y} \right) = z + \left(\frac{\partial z}{\partial y} \right) \end{aligned} \right\}, \text{ visto } \Delta y = 1$$

- Inicialização da profundidade de um polígono quando *in-out* muda para TRUE:

$$Z_{poly} = Z_{edge}$$

- Actualização de Z_{poly} ao fim de Δx pixels:

$$z = z + \Delta x \cdot \left(\frac{\partial z}{\partial x} \right)$$

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: Z-Buffer

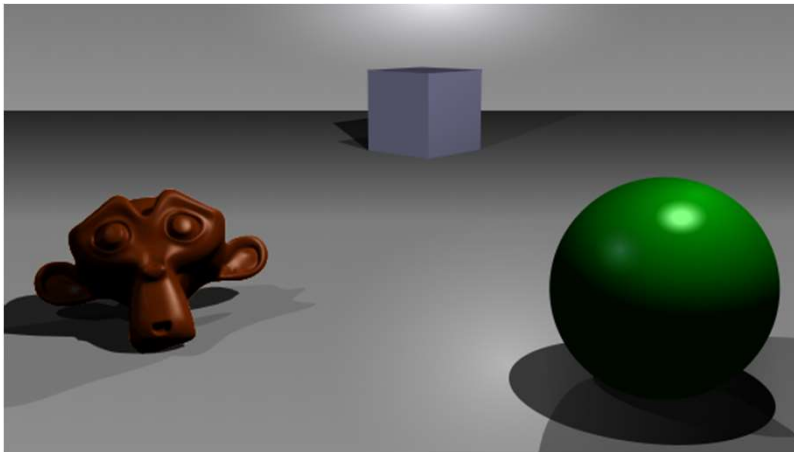
Algoritmo Z-Buffer

Um dos algoritmos mais simples de implementar quer em software quer em hardware.
Não exige qualquer pré-ordenação nem efetua comparações objeto-objeto.

Requisitos: dois buffers

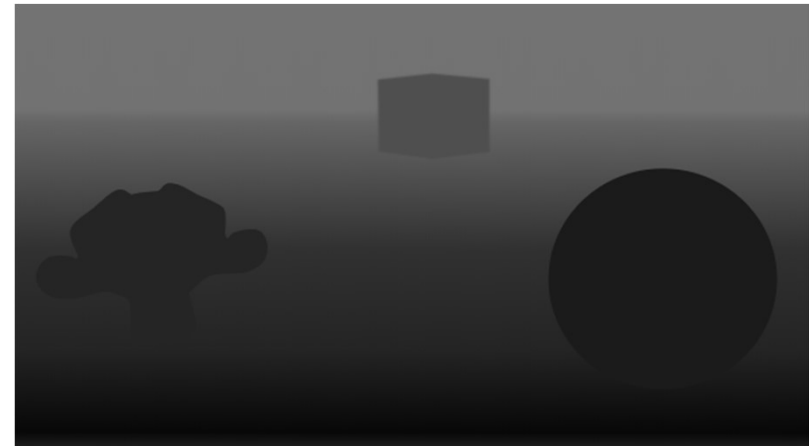
Frame Buffer:

contém a imagem final, pixel a pixel.



Depth Buffer / Z-Buffer:

contém os valores Z, pixel a pixel.



Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: Z-Buffer

Procedimento:

1. Preencher com zeros o *Z-Buffer* e o *frame buffer* com a cor de fundo (background).
O maior valor de Z será o correspondente ao plano frontal de *clipping*.
2. Percorrer cada polígono (Scan-convert), por qualquer ordem.
3. Se o ponto (x,y) do polígono corrente estiver mais próximo do observador do que o ponto actual do Z-Buffer, então o corrente substitui o anterior.

```
for (y=0; y<YMAX; y++)
    for (x=0; x<XMAX; x++)
        ImBuffer(x,y) = BACKGROUND_COLOR;
        ZBuffer(x,y) = 0;
for (cada polígono)
    for (cada pixel na projecção do polígono)
        /* calcula Z para (x,y) no polígono*/
        pz = poligono_Z(x,y);
        if (pz > ZBuffer(x,y))
        {
            ZBuffer(x,y) = pz;
            ImBuffer(x,y) = poligono_cor();
        }
```

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: Z-Buffer

Optimização no processamento:

O valor z do ponto $(x+1, y)$ no polígono pode ser obtido a partir do valor z em (x,y) se atendermos que o polígono é plano.

Tomando $Ax+By+Cz+D=0$:

$$z = \frac{-D - Ax - By}{C}$$

Se em (x,y) a equação tem o valor z_1 então em $(x+1, y)$ o valor de z pode ser calculado:

$$z = z_1 + \Delta x \cdot \frac{\partial z}{\partial x} = z_1 + 1 \cdot \frac{\partial z}{\partial x}$$

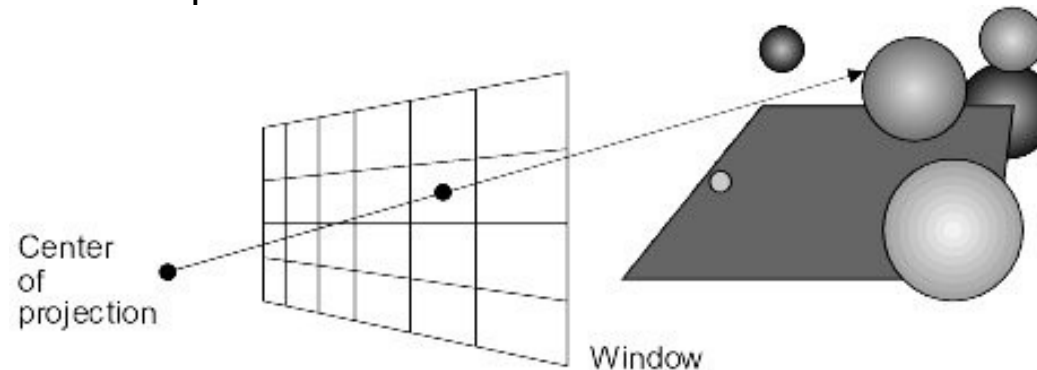
$$z = z_1 - \frac{A}{C}$$

Síntese de Imagem - Cálculo de Visibilidade

Alg. espaço imagem: Ray-casting

Algoritmo Ray-casting

A superfície visível em cada pixel da imagem é determinada traçando um raio de luz imaginário a partir do centro de projeção (observador), passando pelo centro do pixel para a cena 3D. A cor em cada pixel é definido pela cor, no ponto de interseção, do objeto intersetado mais próximo.



Definir Centro de Projeção e window no plano de visualização

```
for (cada linha da imagem)
  for (cada pixel da linha)
  {
    Definir o raio que a partir do centro de projeção passa no pixel
    for (cada objeto na cena)
    {
      if ((objeto intersectado) e (mais próximo do que registro anterior))
        registrar interseção e referência do objeto
    }
    atribuir ao pixel a cor da interseção mais próxima
  }
```

Síntese de Imagem - Cálculo de Visibilidade

Alg. tipo lista de prioridades

Algoritmos tipo Lista de Prioridades

- Alg. Newel, Newel & Sancha
- Binary Space-Partitioning Trees

Objectivo: determinar a ordem de visibilidade para os objectos (polígonos), assegurando assim que a imagem será correctamente criada se os objectos forem desenhados por certa ordem:

1. pintar as faces mais afastadas do observador em primeiro lugar
2. à medida que outras, mais próximas, vão sendo pintadas, ocultam as anteriores.

(Algoritmo do “Pintor”)

Síntese de Imagem - Cálculo de Visibilidade

Alg. tipo lista de prioridades - Alg. Newel, Newel & Sancha

Alg. Newel, Newel & Sancha (Depth-sort algorithm)

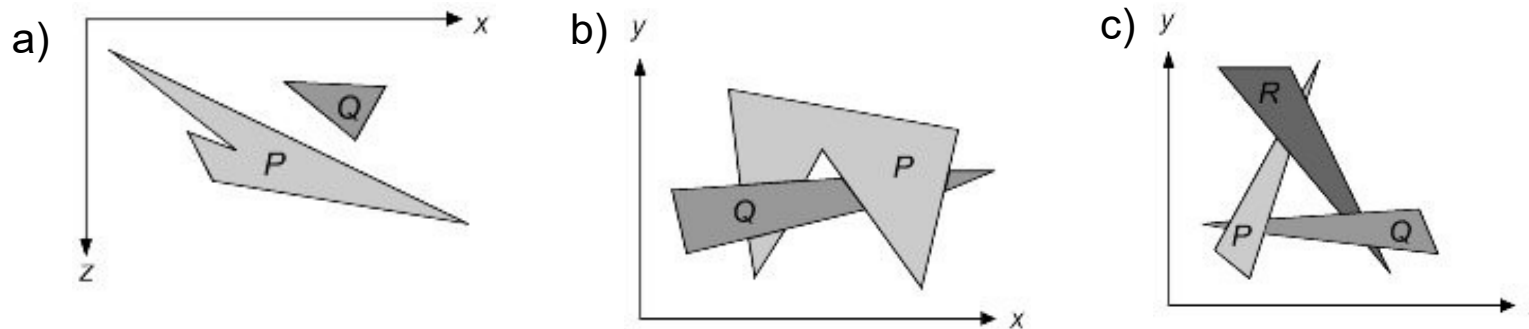
Procedimento: pintar os polígonos por ordem decrescente da distância ao observador. Para isso são realizados 3 passos:

1. Ordenar os polígonos por ordem crescente de **z**
2. Resolver qualquer ambiguidade na ordenação, nomeadamente se houver sobreposição de polígonos na coordenada **z**. Poderá ser necessário dividir polígonos.
3. Pintar os polígonos por ordem do mais afastado para o mais próximo.

Síntese de Imagem - Cálculo de Visibilidade

Alg. tipo lista de prioridades - - Alg. Newel, Newel & Sancha

Tipo de ambiguidades que podem surgir na ordenação dos polígonos:



Pré-processamento:

Ordenar os polígonos pela coordenada **Z do vértice mais afastado**

Processamento:

Para o último polígono **P** da lista, verificar se existe algum polígono **Q** cujo maior **Z** seja mais afastado do que o menor **Z** de **P**, e que esteja a ser obstruído por **P**. Se não estiver, então **P** pode ser desenhado.