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;
```

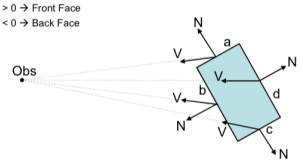
1. Algoritmos no ESPAÇO OBJETO:

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;
```

Backface Culling

- *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.
    - $N.V = |N|.|V|.\cos(\propto)$
      - Resultado > 0 → Front Face
      - Resultado < 0 → Back Face

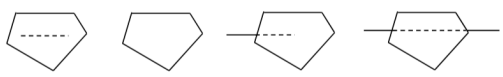


  - Redução:
    - $Número\ de\ Front\ Faces \approx \frac{1}{2} Número\ total\ de\ Faces$

Espaço Objeto

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.

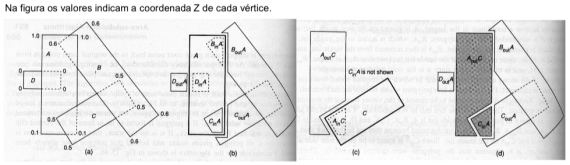
1. Remover todas as faces posteriores dos objectos *(backface culling)* e correspondentes arestas
2. Comparar as arestas restantes contra cada **volume** (poliedro) da cena; deste teste podem ocorrer 4 situações:

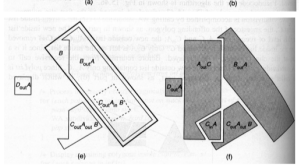


- Aresta completamente oculta pelo volume.
- Aresta não oculta
- Uma parte da aresta não é oculta
- Duas partes da aresta não são ocultas

Atherton & Weiller

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





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

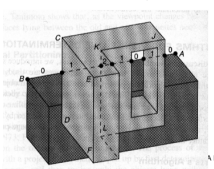
Ao contrario do alg. de Roberts trata ao nível do polígono.

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*)

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**

Warnock

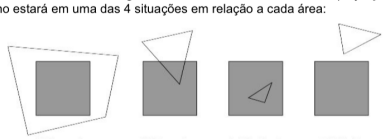
- O algoritmo divide sucessivamente a imagem projectada em áreas rectangulares.
- Se uma área é considerada coerente,
  - A área é desenhada com os polígonos que contém
  - **SENÃO**, a área é dividida em áreas menores; o procedimento é aplicado recursivamente.
- Quanto menores forem as áreas, menor 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.

Warnock

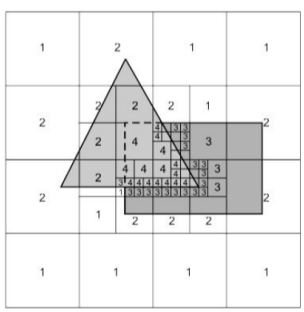
Procedimento alg. Warnock:

- 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.



Alg. 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

Tabela de Arestas (ET): guarda informação de todas as arestas cuja projeção 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 recalcular 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.

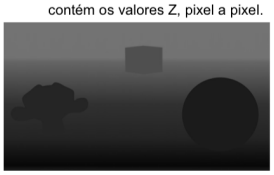
Z-Buffer

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.



Procedimento:

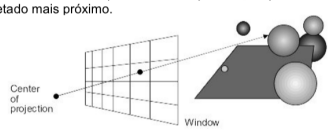
1. Preencher com zeros o *Z-Buffer* e o *frame buffer* com a cor de fundo (background).
   O maior valor de Z será o correspondente ao plano frontal de *clipping*.
2. Percorrer cada polígono (Scan-convert), por qualquer ordem.
3. Se o ponto (x,y) do polígono corrente estiver mais próximo do observador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

→ Otimização nos slides de visibilidade.

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 Projecção 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 registo anterior))
                        registar interceção e referência do objeto
            }
            atribuir ao pixel a cor da interceção mais próxima
      }
```

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.

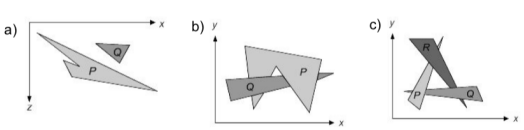
(Algoritmo do "Pintor")

Algoritmo 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.

→ Tipo de Ambiguidades Possíveis



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.