**Visibilidade**

**Sintese de Imagem (rendering)**

Objetivo: a partir de um conjunto de objetos 3D, determinar quais linhas/superficies sao visiveis

Dois calculos possiveis:

1 – algoritmo no espaco imagem:

Text

Description automatically generated with medium confidence

2 – algoritmo no espaco objeto:

Graphical user interface, text

Description automatically generated with medium confidence

Diagram

Description automatically generated**Backface Culling** (tecnica usada para reduzir numero de poligonos a processar)

* front faces : enviar para calculo de visibilidade
* back faces: ignorar

Reducao: numero de front faces = ½ total de faces

**Algoritmos espaço objeto**

* ao volume: algoritmo de roberts
* à aresta: algoritmo de Appel, Loutrel, Galimberti e Montanari

(nestes algoritmos, todas as arestas sao testadas para produzir uma lista com os segmentos visiveis de todas as arestas)

**ao volume:** supoe-se que uma aresta pode ser oculta pelo volume de um objeto

**à aresta:** teste é efetuado aresta contra aresta observando que a visibilidade de uma aresta goza de coerencia, o que permite determinar a visibilidade de uma aresta a partir da invisibilidade de outra aresta que possa com ela um vertice comum

**coerencia da aresta:** uma aresta so altera a sua visibilidade onde se cruza por tras de uma aresta visivel

**Ao volume (algoritmo de roberts):**

Requisito: cada aresta deve pertencer a uma face de um poliedro convexo. poliedros concavos devem ser partidos em varios convexos para poder aplicar o algoritmo

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

Shape, polygon

Description automatically generated

**À aresta (algoritmo de Appel, Loutrel, Galimberti e Montanari):**

Ao contrario de alg. de robrts, trata ao nivel do poligono

1. determinar as faces orientadas para o observador (backface culling)
2. calcular a “quaititative invisibility” de um vertice para cada objeto
   1. “Quantitative Invisibility” QI de um ponto: é o número de polígonos entre o observador e o próprio ponto.”
   2. 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.
   3. 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.

A picture containing graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

**Determinacao de faces visiveis (atherton & weiller):**

Requisito: exige a aplicacao de um algoritmo sofisticado de clipping de poligonos, capaz de efetuar clipping de um poligono concavo com buracos contra um outro qualquer

Procedimento:

1. Ordenar os poligonos pela coordenada “Z”; o poligono mais proximo do observador é selecionado como janela de corte
2. o poligono é usado para efetuar clipping dos restantes, resultado duas listas contendo os poligonos (ou parte deles) que sao interiores (inivisiveis) e exteriores (visiveis) à regiao de clipping
3. os poligonos da lista interior sao marcados como invisiveis
4. passa ao poligono seguinte da lista de exteriores e regressa ao ponto 2.

Diagram, engineering drawing

Description automatically generated

**Algoritmos espaco imagem**

1. Orientada à area: Algoritmo de warnock
2. Orientado à linha: Linha de varrimento/Watkins
3. Orientado ao pixel: z-buffer, ray casting

**Algoritmo de warnock**

* algoritmo divide sucessivamente a imagem projetada em areas retangulares
* se uma area é considerada coerente:
  + area é desenhada com poligonos que contem
  + senao, area é dividida em areas menores; procedimento aplicado recursivamente
* quanto menores forem as areas, menor numero depoligonos estarao sobrepostos nessas areas; é mais facil se podera decidir qual o poligono a desenhar

coerencia de area: um grupo de pixels adjacentes é habitualmente coberto pela mesma face visivel

Procedimento:

1. divisao da area em 4 blocos iguais. em cada fase da subdivisao, a projecao de cada poligono estará em uma das 4 situacoes em relacao a cada area:

Shape

Description automatically generated

As situacoes em que a decisao é possivel, nao havendo mais subdivisao:

1. apenas um poligoono que ocupa toda a area, nao havendo mais nenhum projetado nessa area. pinta a area com a cor desse poligono

b, c. apenas um poligono que interseta ou que está totalmente dentro da area. preencher a area com a cor de fundo e depois pintar a parte do poligono que se encontra nessa area

* 1. todos os poligonos estao fora da area -> pinta a area à cor de fundo

Diagram, engineering drawing

Description automatically generated

**Algoritmo de linha de varrimento**

A imagem é criada linha a linha, à semelhanca do algoritmo de preenchimento de regioes 2D, designado por algoritmo da lista das arestas ativas.

Conceitos:

* **coerencia vertical:** o conjunto de objetos visiveis determinados para uma linha de varrimento, difere pouco do conjunto correspondente da linha anterior
* **coerencia de aresta:** uma aresta so altera a sua visibilidade quando se cruza com outra aresta visivel ou quando penetra uma face

Estruturas de dados:

* **AEL:** lista de arestas ativas
* **ET:** tabela de (novas) arestas
* **PT:** tabela de poligonos

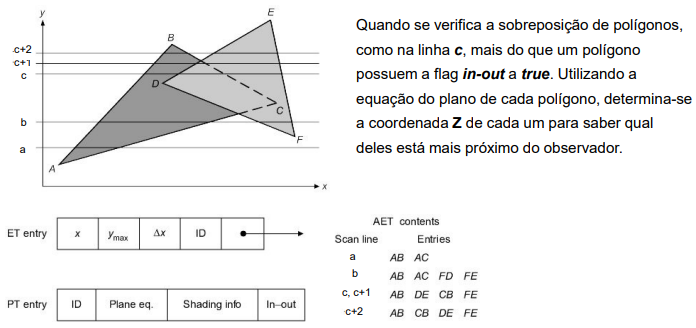
**Tabela de arestas (ET):** guarda informacao de todas as arestas cuja projecao no plano de visualizacao nao é horizontal. As entradas da tabela estao ordenadas de forma crescente pelo menor valor de Y, e conteem inicialmente:

1. corrdenadas (X,Z) do vertice com menor Y
2. altura das arestas (Y1-Y0) ou, em alternativa, Ymax
3. incrementos  usados na atualizacao de X e de Z, na passagem para a linha de varrimento seguinte
4. identificacao do(s) poligono(s) partilhados pela aresta

**Tabela de poligonos (PT):** informacao de todos os poligonos, contendo para cada um:

1. coeficientes da equacao do plano
2. informacao da cor
3. coordenada Z, a recalcular a cada pixel
4. flag de in-out, inicializada a False, é usada para controlar se o processamenteo está dentro ou fora do poligono

**Lista de arestas ativas (AEL):** informacao de quais as arestas ativas na linha de varrimento atual



Graphical user interface, text, application

Description automatically generated

**algoritmo z-buffer**

um dos algoritmos mais simples de implementar

nao exige qualquer pre-ordenacao nem efetua comparacoes objeto-objeto

Requisitos: dois buffers

* frame buffer: conte a imagem final, pixel a pixel
* depth buffer / z-buffer: contem os valor Z, pixel a pixel

Procedimento:

1. preencher com zeros o z-buffer e o frame buffer com a cor de fundo
   1. o maior valor de Z sera o correspondente ao plano frontal de clipping
2. percorrer cada poligono (scan-convert) por qualquer ordem
3. se o ponto (x,y) do poligono corrente estiver mais proximo do observado do que o ponto atual do z-buffer, entao o corrente substitui o anterior

Text, letter

Description automatically generated

Otimizacao do processamento:

Text

Description automatically generated with medium confidenceO valor z do ponto (x+1, y) no poligono pode ser obtido a partir do valor z em (x,y) se atendermos que o poligono é plano

Tomando Ax + By + Cz + D = 0:

Se em (x,y) a equacao tem o valor z1 entao em (x+1,y) o valor de z pode ser calculado:

Text

Description automatically generated

**Algoritmo ray-casting:**

superficie visivel em cad apizel da imagem é determinada traçando um raio de luz imaginario a partir do centro de projecao (observado), passando pelo contro do pixel para a cena 3D. a cor em cada pixel é definido pela cor, no ponto de intersecao, do objeto intersetado mais proximo

Diagram

Description automatically generated

**Algoritmo tipo lista de prioridades**

* alg. newel, newel & sancha
* binary space-partitioning trees

Objetivo: determinar a ordem de visibilidade para os objetos assegurando assim que a imagem será corretamente criada se os objetos forem desenhados por certa ordem:

1. pintar as faces mais afastadas do observado em primeiro lugar
2. à medida que outras, mais proximas, vao sendo pintadas, ocultam as anteriores

**algoritmo newel, newel & ssancha**

Procedimento: pintar os poligonos por ordem decrescente da distancia ao observador. para isso sao realizados 3 apssos:

1. ordenadar os poligonos por ordem crescente de Z
2. resolver qualquer ambiguidade na ordenacao, nomeadamente se houver sobreposicao de poligonos na coordenadas Z. podera ser necessario dividir poligonos.
3. pintar os poligonos por ordem do mais afastado para o mais proximo

Shape

Description automatically generated

**Algoritmos de Iluminacao Global**

* Ray tracing
* radiosity

**Ray Tracing**

Algoritmo depende da posicao do observador

* plano de visualizacao é discretizado em pontos de amostragem (pixels ou ...)
* faz-se passar, por cada ponto de amostragem, um raio luminoso que parte do observador em direcao ao interior da cena
* Diagram

  Description automatically generatedDiagram

  Description automatically generatedo rasto (tracing) de cada raio vai permitir somar as contribuicoes de reflexao entre faces proximasDiagram

  Description automatically generatedDiagram, engineering drawing

  Description automatically generated

O algoritmo de ray tracing é vantajoso porque:

* sombras, reflexoes e refracoes sao facilmente incorporadas
* simula razoavelmente bem os efeitos especulares

O algoritmo de ray tracing tem custos computacionais elevados porque:

* o custo de calculo das intersecoes é elevado
* nao simula bem os efeitos de iluminacao difusa (necessita de outras variantes)

A otimizacao faz-se em duas areas:

* diminuicao do numero de raios a processar
* diminuicao do numero de intersecoes a testar

Diminuicao do numero de raios a processar:

* “item buffers”: determinam-se quais as areas do ecra onde se situam os objetos
* “adaptive tree-depth control”: nao é necessario levar todos os ramos da arvore de shading à sua profundidade máxima: usa a importancia de um raio luminoso sobre o pixel a que pertence; esta importancia diminui a cada reflexao ou transmissao
* “light-buffers”: a cada font de luz associam-se listas com os objetos que a rodeiam; os shadow feelers, uma vez definida a sua direcao, soa primariamente testados com os objetos que se encontram na lista respetiva.

Diminuicao de numero de intersecoes a testar:

* volumes envolventes: antes de efeutar o teste de intersecao de um raio com um objeto, tenta-se a sua intersecao com um volume simpes (normalmente numa caixa) envolvente do objeto. este teste previo é muito rápido (a caixa tem as faces alinhadas com os tres eixos) e exclui imediatamente muitos testes de intersecao mais complexos
* organizacao hierarquica dos volumes envolventes: a utilizacao de volumes envolventes de outros volumes envolventes permite economizar muitos testes de intersecao: se um raio nao interseta um volme, entao tambem nao interseta os volumes nele contidos.
* divisao espacial em grelhas tridimensionais: cada celula resultante desta divisao conhece os objeto que contem, total ou parcialmente. de acordo com a posicao e a direcao do raio em questao, so determinadas celulas sao visitadas e, deste modo, so os objetos nelas contido sao testados. dado que a ordem de progressao nas celulas é difinida pelo sentido do raio, a primeira celelula onde se detete uma intersecao termina o processo de visita do raio às celulas

**Radiosity**

Algoritmo independente do ponto de observacao

Algoritmo so efetua o calculo de iluminacao; trabalho no espaco objeto.

Fases de processamento:

1. modela as interacoes de luz entre objetos e fontes de luz, sem considerar a posicao do observador
2. cria a imagem considerando o observador, efetua o calculo de visibilidade

Nos modelos de iluminacao anteriores, as fontes de luz foram tratadas de forma diferente das superficies que iluminam. Pelo contrario, os metodos de radiosidade consideram que todas as superficies podem produzir (emitir) luz. Assim, as fontes de luz sao modeladas como superficies normais, com uma dada area.

O metodo assume que os processos de emissao e reflexao sao “difusos ideais”. Necessita das faces discretizadas em “patches” de forma a garantir que a radiosidade se mantem constante na area correspondente a um “patch”.

Text

Description automatically generatedText, letter

Description automatically generated

Criacao da imagem:

1. Resolvendo o sistema de equacoes, por eliminiacao gaussiana, obtem-se a radiosidade para cada “patch”.
2. Definir a posicao do observador
3. aplicar um algoritmo de visibilidade (ex. z-buffer)
4. calcular a radiosidade dos vertices de cada poligono
5. aplicar a interpolacao de cor (gouraud)

* a mesma solucao do sistema é usada para qualquer posicao do observador
* é necessario resolver novamente o sistema de equacoes se houver alteracoes de coeficientes de reflexao “ρ” ou de valores de emissao “E”
* é necessario recualcular os fatores de forma se a geometria da cena for alterada (mudanca de posicao de algum objeto)

**Fatores de forma**: complexidade do metodo esta no calculo dos fatores de forma

O fator de forma **Fij** representa a percentagem da energia total expelida pelo patch “i” que atinge o patch “j”, tomando em consideracao a forma, orientacao relativa e distancia entre ambos os patches, bem como os obstaculos que obstruam o caminho

Diagram

Description automatically generated

**Simplificacao de Cohen e Greenberg:** metodo do hemicubo

Em vez de usar a projecao num hemisferio, projeta na parte superior de um cubo centrado em **dAi**, sendo a parte superior do cubo paralela com a superficie

A picture containing diagram

Description automatically generatedA picture containing text, antenna

Description automatically generated

A picture containing text

Description automatically generated

Diagram

Description automatically generated with medium confidence**Representacao de curvas e superficies**

**Malha poligonal**

colecao de arestas, vertices e poligonos interligados de modo que cada aresta é apenas ligada por um ou dois poligonos

Caracteristicas:

* uma aresta liga 2 vertices
* um poligono é definido por uma sequencia fechada de arestas
* uma aresta é ligada a um ou dois poligonos (adjacentes)
* um vertice é partilado pelo menos por 2 arestas
* todas as arestas fazem parte de algum poligono

A estrutura de dados para representar a malha poligonal pode ter varias configuracoes, que sao avaliadas pelo espaco de memoria e tempo de processamente necessario para obter resposta, por exemplo, a:

* obter todas as arestas que se unem num dado vertice
* determinar os poligonos que partilham uma aresta ou um vertice
* etc

Representacoes:

1. Diagram

   Description automatically generatedrepresentacao explicita: cada poligono é representado por uma lista de coordenadas dos vertices que o constituem

uma aresta é definida por dois vertices consecutivos, fechando o poligono

P = ((x1,y1,z1),(x2,y2,z2), ..., (xn,yn,zn))

Avaliacao da estrutura de dados:

* vertices repetidos a quantidade de arestas que há nesse vertice
* nao ha representacao explicita das arestas e vertices partilhados

1. representacao por apontadores para lista de vertices: cada poligono é representado por uma lista de indices (ou apontadores) para uma lista de vertices

Text, letter

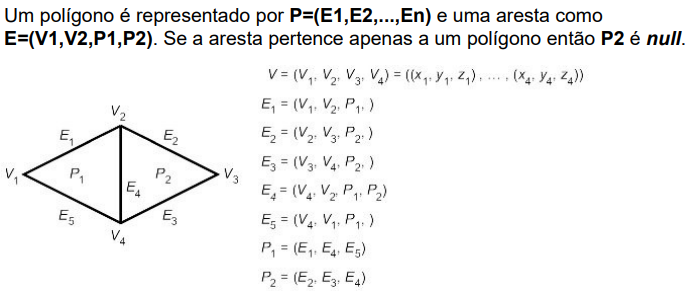
Description automatically generatedVantagens:

* cada vertice da malha é guardado uma vez
* coordenada de um vertice facilmente alterada

Desvantagens:

* dificil obter poligonos que partilham uma aresta
* arestas continuam a ser desenhadas mais que uma vez

1. representacao por apontadores para lista de arestas: cada poligono é representado por uma lista de apontadores para uma lista de arestas, na qual cada aresta aparece uma unica vez. por sua vez, cada aresta aponta para os dois vertices que a definem e guarda tambem quais os poligonos a que pertence.

Vantagens:

* desenha grafico é facilmente obtido percorrendo a lista de arestas. nao ocorre a repeticao de desenho de arestas
* para o preenchimento dos poligonos trabalha-se com a lista de poligonos. facil efetuar a operacao de clipping sobre os poligonos

Desvantagens:

* continua a nao ser imediato determinar quais as arestas que incidem sobre o mesmo vertice

**Solucao de baumgart**

* cada vertice tem um apontador para uma das arestas (aleatorio) que incide nesse vertice
* cada aresta apresenta um apontador para a proxima aresta que incide nesse vertice

**Curvas cubicas**

A representacao por malha poligonal é uma aproximacao de primeira ordem:

* a curva é aproimada por uma sequencia de segmentos lineares
* grande quantidade de dados (vertices) para obter curva com precisao
* dificil manipulacao para mudar a forma da curva (ex. necessario posicionar varios pontos com precisao)

Geralmente utilizam-se polinomios de grau 3, sendo a curva completa formada por um conjunto de curvas cubicas

* grau < 3 : pequena flexibilidade no controlo da forma das curvas e nao permitem interpolacao entre dois pontos com a definicao da derivada nos pontos extremos. um polinomio de grau 2 é especificado por 3 pontos que definem o pano onde a curva toma lugar
* grau > 3: podem introduzir oscilacoes indesejaveis e exigir maior calculo computacional

Representacao:

* forma explicita -> y = f(x) (ex. y = x^3 + 2x^2)
  + nao podemos ter varios valor de y para o mesmo x
  + nao podemos descrever curvas com tangentes verticais
* forma implicita -> f(x,y) = 0 (ex. x^2 + y^2 – r^2 = 0)
  + necessita de restricoes para poder modelar apenas uma parte da curva
  + dificil juntar duas curvas de forma suave

Text, letter

Description automatically generated

**Curvas cubicas parametricas**

Tipos de continuidade:

* G0: continuidade geometrica zero: curvas so se juntam num ponto
* G1: continuidade geometrica um: direcao dos vetores tangentes é igual
* C1: continuidade parametrica 1: tangentes no ponto de juncao teem a mesma direcao e amplitude
* Cn: continuidade parametrica n: curvas teem no ponto de juncao todas as derivadas iguais até ordem n

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generated**Tipos de curvas**

1. Curvas de Hermite
   1. continuidade G1 nos pontos de juncao
   2. vetor geometrico
      1. 2 pontos extremos
      2. os vetores tangentes nesses pontos
2. Diagram

   Description automatically generated with medium confidenceCurvas de Bezier
   1. continuidade G1 nos pontos de juncao
   2. vetor geometrico
      1. 2 pontos extremos
      2. 2 pontos que controlam os vetores tangentes nesses extremos
3. Curvas Splines
   1. familia de curvas muito alargada
   2. maior controlo de continuidade nos pontos de juncao (continuidade C1 e C2)

Table

Description automatically generated**Notacao comum**

Text, letter

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated**Curvas de Hermite**

Diagram

Description automatically generated

**Curvas de Bezier**

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

**Desenhos de curvas cubicas**

Dois algoritmos:

1. avaliacao de x(t), y(t) e z(t) para valores incrementais de t entre 0 e 1.
2. Subdivisao da curva: Algoritmo de Casteljau
3. Avaliacao de x(t), y(t) e z(t)

É possivel reduzir o numero de operacoes de 11 multiplicacoes e 10 adicoes para 9 e 10, respetivamente.



1. Algoritmo de Casteljau

Efetua a subdivisao recursiva da curva, parando apenas quando a curva em questao é suficientemente “plana” para poder ser aproximada por um segmento de reta.

Criterios possiveis de paragem:

* curva em questao é suficientemente “plana” para poder ser aproximada pelo seu convex hull
* Text, letter

  Description automatically generatedos 4 pontos de controlo estao no mesmo pixel

Diagram

Description automatically generated

**Superficies cubicas**

as superficies cubicas sao uma generalizacao das curvas cubicas. a equacao da superficie é obtida a partir da equacao da curva:

Q(t) = T . M . G, send G constante.

Mudar para a variavel s: Q(s) = S . M . G

Fazendo variar os ponts do vetor geometrico em 3D ao longo de um percurso parametrizado por t obtem-se:

A picture containing text

Description automatically generated

Diagram

Description automatically generatedA picture containing table

Description automatically generated**Superficie de Hermite**

Superficie de Bezier

Diagram, engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generated