

Apontamentos CGI

(Estes apontamentos têm apenas conceitos teóricos, pois a parte mais prática está presente no outro pdf também com apontamentos)

PRIMITIVAS GRÁFICAS:

Pontos
Linhas
Polígonos

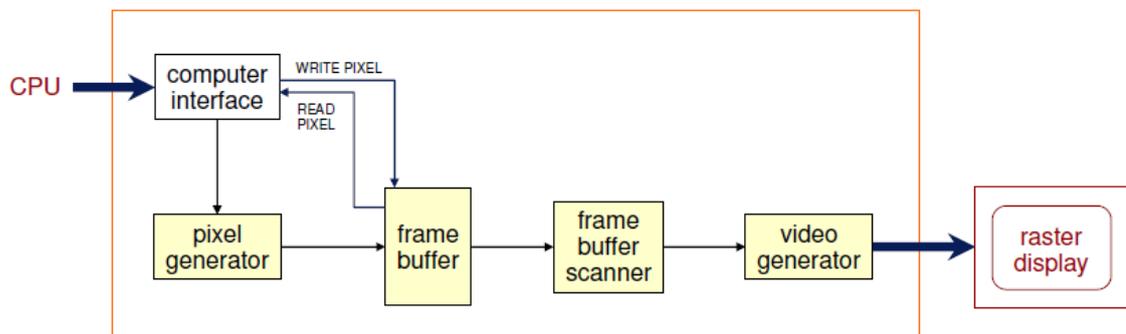
Características do uso da Posição Corrente Posição da caneta virtual nesse instante

(ISO/ANSI não usam este conceito)

- + Reduz o nº de argumentos das primitivas gráficas
- Aumenta o nº de primitivas
- Não tem independência da ordem de execução
- Implica valores arbitrários em certas transições de estado do sistema gráfico.

Coordenadas absolutas vs relativas

Com relativas é necessário a noção da posição corrente.



PIXEL GENERATOR: Interpretador das instruções do CPU, mas escrevendo no frame buffer só quando o scanner não o estiver a utilizar.

FRAME BUFFER: Memória de refrescamento, cujas dimensões em x e em y tanto podem ser potências de 2 como os valores correspondentes às dimensões do sistema de visualização

FRAME BUFFER SCANNER: Módulo de varrimento da memória de refrescamento (ESQUERDA PARA A DIREITA E DE CIMA PARA BAIXO) com modulação do sinal de saída proporcionalmente aos valores nessa memória.

VIDEO GENERATOR: Produz o sinal de vídeo para o sistema de visualização.

Tecnologia RASTER: Imagens de boa qualidade
Algoritmos de execução rápida

PIXEL

WRITE_PIXEL(round(x), round(y), value)

LINHA

Atributos: cor, grossura, tipo de traço

Existem modos de escrita no frame buffer que podem originar efeitos diferentes quando se repete a escrita de um mesmo pixel (XOR)

Rubber Banding

ALGORITMO INCREMENTAL:

Quando $m > 1$ ficam pixels por intensificar.

+ Eliminação do produto $m * x$

- Acumulação de erros

POLÍGONO

FILL AREA (pintar)

RASTERIZAÇÃO DE LINHAS CURVAS GENÉRICAS

CURVA

Especificada por uma ou mais equações com **uma** só variável independente.

◆ Descrição não-paramétrica

- Forma explícita

$$Q = (x , y=f(x) , z=g(x))$$

- Forma implícita

$$Q = (F(x,y,z)=0 , G(x,y,z)=0)$$

◆ Descrição paramétrica

$$Q = (x=f(t) , y=g(t) , z=h(t))$$

$$a \leq t \leq b$$

SUPERFÍCIE

Especificada por uma ou mais equações com **duas** variáveis independentes.

◆ Descrição não-paramétrica

- Forma explícita

$$Q = (x , y , z=f(x,y))$$

- Forma implícita

$$Q = (F(x,y,z)=0)$$

◆ Descrição paramétrica

$$Q = (x=f(s,t) , y=g(s,t) , z=h(s,t))$$

$$a \leq t \leq b$$

$$c \leq s \leq d$$

Descrição não paramétrica:

Forma Explícita:

- ☺ Facilidade de cálculo
- ☹ Não representa não funções
- ☹ Não se aplica transformações por operadores matriciais

Forma Implícita:

- ☺ Representa não funções
- ☹ Difícil determinar raízes
- ☹ Não se aplica transformações por operadores matriciais

Descrição paramétrica:

- ☺ Representa não funções
- ☺ Aplicam-se transformações por operadores matriciais

CONCLUSÃO: Usar descrições paramétricas

SPLINES NATURAIS

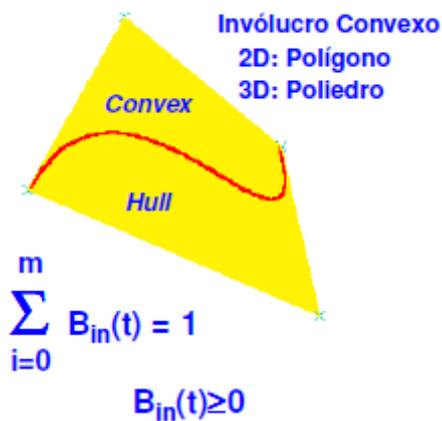
Não é boa solução para design interativo

- Sendo o sistema de equações global, o controlo da curva é **global**.
- O ajuste da curva depende inteiramente da qualidade dos pontos, não se garantindo a ausência de oscilações indesejáveis.

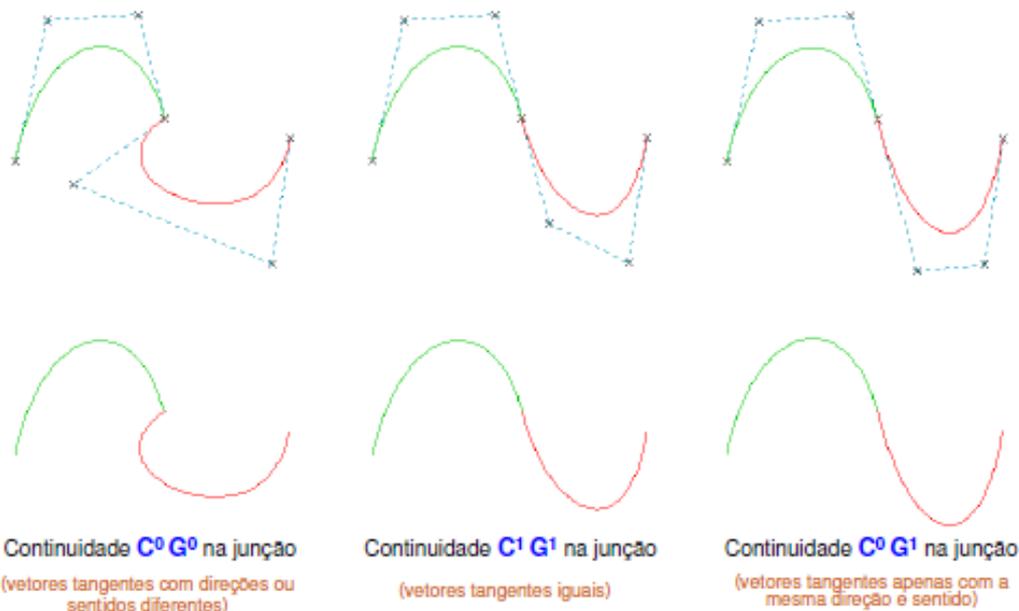
Bezier, catmull e b-spline têm controlo local, ou seja, pode-se mover um ponto e parte da curva mantém-se inalterada.

BEZIER:

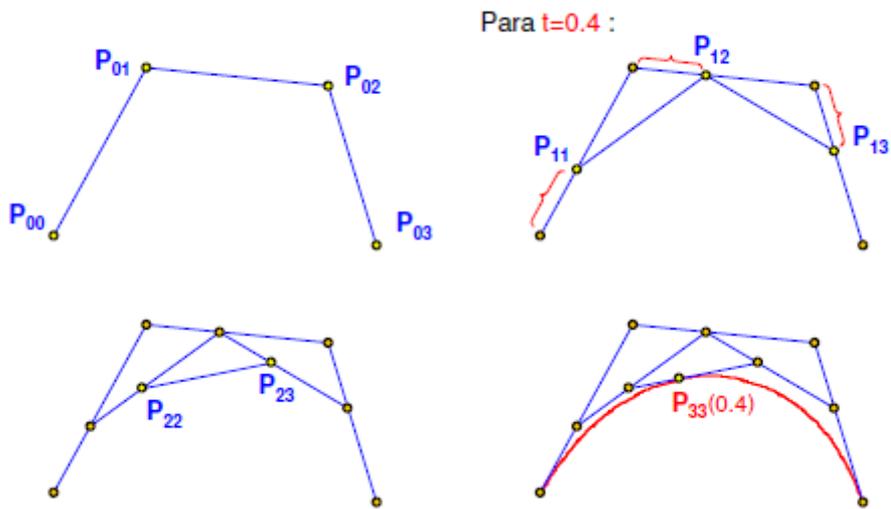
Interpolam sempre dois pontos



CONTINUIDADE PARAMETRICA E GEOMETRICA

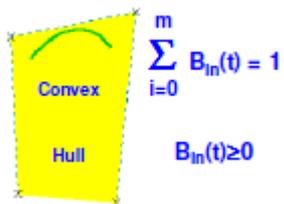


DE CASTELJAU



B-SPLINE

Sempre C2



Continuidade Geométrica vê-se nos pontos em que os troços se juntam, ou seja uma curva pode ter multiplicidade 3, no entanto se esta multiplicidade estiver presente somente no ponto inicial ou final não influencia na continuidade geométrica.

Multiplicidade 1: G2

Multiplicidade 2: G1

Multiplicidade 3: G0

CATMULL-ROM

Sempre C1G1, só é G0 quanto revela um ponto anguloso

ENQUADRAMENTO

$a_{\text{visor}} \leq a_{\text{janela}} \rightarrow$ escalar pela largura

Caso contrário escala-se pela altura

Full screen: $a_{\text{visor}} = a_{\text{janela}}$

TRANSFORMAÇÕES GEOMÉTRICAS

SRT

Garantir Comutatividade:

- $R(a).R(b)$

- $S(K1,K2).S(K3,K4)$

- $T(D1,D2).T(D3,D4)$

- $S(K,K).R(a)$

PROJECCÕES

PERSPECTIVA:

→ Não é paralela

→ Pontos de fuga

OBLÍQUA

→ Alçado principal sempre de frente

Cavaleira: $l = 1$ $B = 45^\circ$

Gabinete: $l = 0.5$ $B = 63.4^\circ$

Ortogonal: $l = 0$ $B = 90$

AXONOMÉTRICA

- Isometria: $r1 = r2 = r3 = 1$

- Dimetria

- Trimetria

Paralelismo é preservado

SUPERFÍCIES

Continuidade Geométrica e Continuidade Paramétrica

Superfície formada por 2 retalhos
Gh: matriz de geometria de Hermite

- Para garantir G0: $Gh[2,i]=Gh[1,i]$
- Para garantir G1: $Gh[4,i]=k.Gh[3,i]$

Primeira derivada: vector tangente
Segunda derivada: vector torção

Produto externo entre as derivadas: vector normal

TEXTURAS

COMPARAÇÃO DOS MÉTODOS RELATIVOS À ORDEM DE EXECUÇÃO

$T=f^{-1}(P')$ INVERSE SCANNING

- :(Exige o **cálculo** das transformações de visualização **inversas**
- :) Evita a subdivisão de pixels e permite o uso de filtros para **antialiasing**

$P'=f(T)$ TEXTURE SCANNING

- :(Em geral a textura **não** coincide com um **nºinteiro de pixels**, implicando **cálculos** para a subdivisão da área de um pixel.
- :) Transformações de visualização seguem-se no sentido mais natural.

CORRESPONDÊNCIA ENTRE PIXELS E TEXTELS

Um pixel tem uma determinada área.

Quando se faz a correspondência com a textura, a área aí obtida poderá ser diferente da área de um texel.

pintado por AMPLIAÇÃO
ou REDUÇÃO

Soluções:

- usar texel mais próximo do pto calculado pela correspondência do centro do pixel (GL_NEAREST)
- média ponderada com os 2x2 texels mais próximos do pto calculado (GL_LINEAR) → Mais lento, melhor resultado

TIPOS DE MAPEAMENTO

Ortogonal

Cilíndrico

Esférica

DOUBLE BUFFERING (usado em animação)

instante t: Escrita do frame i+1 no 1º frame buffer
Visualizar frame i do 2º frame buffer

instante t+1: Visualizar frame i+1 do 1º frame buffer
Escrita do frame i+2 no 2º frame buffer

HLHDR → Remover partes ocultas (Culling)

- polígono convexo: Aquele em que uma linha unindo dois quaisquer pontos interiores ao polígono esteja nele totalmente contida.

Determinação da posição de um ponto via **produto externo** de dois vectores:

MÉTODO DO PRODUTO INTERNO

(Test for **CULLING** Polygon Faces) → Tem de ser convexo

A superfície não é visível $90^\circ \leq \theta \leq 180^\circ$

A superfície é visível $0^\circ \leq \theta < 90^\circ$

Z-BUFFER

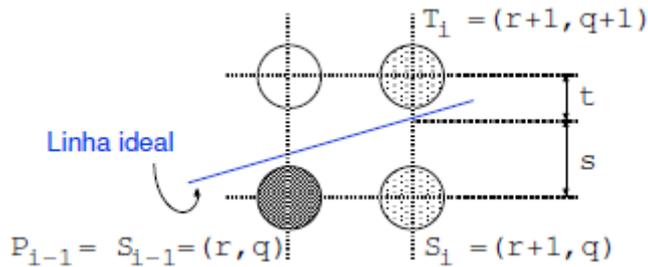
Cada pixel de um polígono só é escrito se não houver sobreposição com nenhum outro ou, caso contrário, se a sua **distância ao observador for menor que a distância associada ao pixel, já escrito**, pertencente a polígono tratado em fase anterior.

Justificação pelas características de X3D/VRML:

A ordem de execução das transformações, em cada nó, é S – R – T (independentemente da ordem de escrita no nó Transform)

LINES+FILL:

ALGORITMO DE BRESENHAM



```
procedure LINE_BRESENHAM (x1,y1,x2,y2,value: integer);
```

```
{ Condição de aplicabilidade:  $0 \leq m \leq 1$ 
```

```
Para resolver nos outros octantes faz-se uma conversão para este! }
```

```
var dx,dy,ks,kt,d,x,y,x_end: integer;
```

```
begin
```

```
dx := abs(x2-x1); dy := abs(y2-y1); d := 2*dy - dx;
```

```
ks := 2*dy; kt := 2*(dy - dx);
```

```
if x1 > x2 then begin x := x2; y := y2; x_end := x1 end
```

```
else begin x := x1; y := y1; x_end := x2 end;
```

```
WRITE_PIXEL(x,y,value);
```

```
while x < x_end do
```

```
begin
```

```
x := x + 1;
```

```
if d < 0 then d := d + ks
```

```
else begin y := y + 1; d := d + kt end;
```

```
WRITE_PIXEL(x,y,value)
```

```
end
```

```
end;
```

ALGORITMO DO PONTO MÉDIO

Efeito escada, mesmo aumentando a resolução da imagem este problema não é minorado.

POLÍGONOS+FILL:

FILL AREA

O algoritmo de Sutherland-Hodgman pode gerar arestas estranhas (como F'H'' ou H''H') aos polígonos não convexos, com possibilidade de serem eliminadas apenas em pós-processamento.

No entanto, não haverá problemas desse gênero se se recortar um qualquer polígono convexo.

SCAN CONVERSION PROBLEMS

Aliasing: Plotting a point in a location other than its true location.
round(x), round(y)

Devido a este arredondamento, pode acontecer o EFEITO ESCADA

global aliasing: Equal distances between pickets but different overall length.

local aliasing: Equal overall length but different distances between pickets.

ANTIALIASING PELO METODO DE FILTRAGEM:

Aplica-se (por pós-processamento) a uma imagem já existente.

Método: o valor de cada pixel contribui, por soma ponderada, para os valores dos pixels vizinhos e na relação inversa da distância.

Este cálculo não é cumulativo, dando resultados bastante aceitáveis tanto para linhas como para polígonos.

CORES

ORIENTADO PARA O EQUIPAMENTO

→ Especificação precisa das cores numa certa gama, para determinada classe de equipamentos.

Sistema aditivo: **RGB**

Luz emitida por 3 focos de cores primárias

Sistema subtrativo: **CMY** (mistura de pigmentos no papel)

Luz reflectida por 3 cores primárias, precisa de luz branca incidente

CMYK: para melhorar a qualidade adicionou-se o pigmento BLACK

$K = \min(C, M, Y)$

Atributos psicológicos da cor:

- tom, tonalidade HUE
- saturação (da cor pura ao cinzento)
- claridade, brilho, intensidade

ORIENTADO PARA O SER HUMANO

→ Especificação precisa das cores e de acordo com os respectivos atributos psicológicos.

Interpolar: Soma dos valores/2

HSV/HSB Hue, Saturation, Brightness/Value

H ângulo

V varia na altura

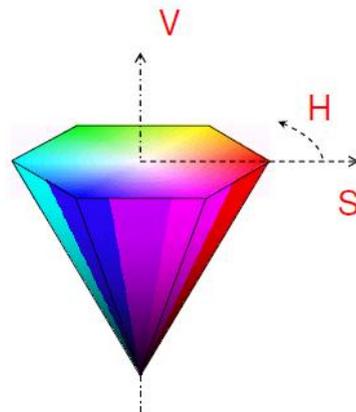
S distancia ao centro

RGB para HSV

$V = \max(R, G, B)$

$S = V - \min(R, G, B) / V$

$S = 0 \rightarrow H$ undefined



HLS Hue, Lightness, Saturation

O branco é considerado a cor mais luminosa

H ângulo

L varia na altura

S distancia ao centro

RGB para HLS

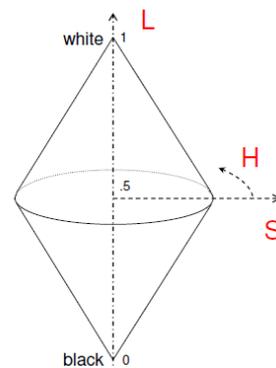
$L = (\max(R, G, B) + \min(R, G, B)) / 2$

if ($\max(R, G, B) == \min(R, G, B)$) $S = 0$; // H undefined

else if ($L < .5$) $S = (\max(R, G, B) - \min(R, G, B)) / (2 * L)$;

else $S = (\max(R, G, B) - \min(R, G, B)) / (2 - 2 * L)$;

$S = 0 \rightarrow H$ undefined



MODELO DE ILUMINAÇÃO

Reflexão difusa:

$$I = I_p \cdot K_d \cdot \cos(\text{ang})$$

I – intensidade luminosa num ponto P da superfície reflectora S

I_p – intensidade do foco (cor incidente no obj)

k_d – coeficiente de reflexão difusa (cor reflectida pelo obj)

ang – angulo de incidência da luz

LOGICAL INPUT DEVICES

Locator: crosshairs (devolve uma po

Stroke: tablet (devolve sequencia de pontos)

Valuator: devolve um número real

Choice: devolve uma selecção de um conjunto de opções

Pick: Identifica um objecto seleccionado

String: input de chars