

P2 CG 2013.2



Computação Gráfica 1
Prof. Rodrigo de Toledo

Data: 4/12/2013
P2 2013.2

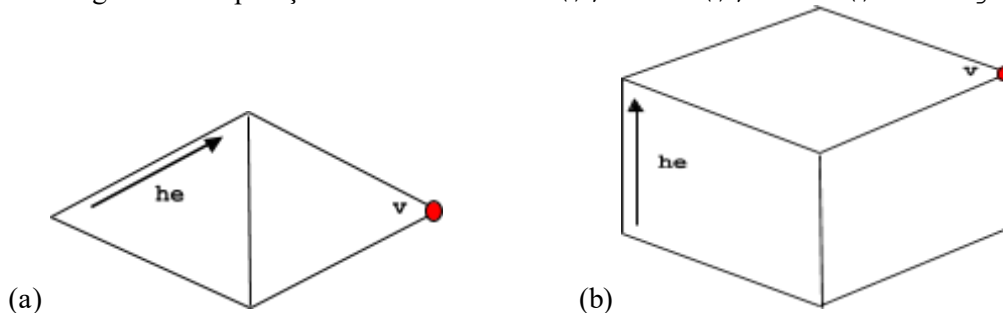
1) (2 pontos) Primitivas implícitas em GPU

Vimos que objetos como cilindros, cones, esferas e tori podem ser visualizados em tempo real nas placas de vídeo com o z-buffer usando duas técnicas diferentes: discretizadas em polígonos (usualmente triângulos) ou através de primitivas implícitas em GPU, renderizadas por traçado de raios (como na tese do Rodrigo). Compare essas duas técnicas, julgando para cada um dos quesitos qual é a melhor, ou se “tanto faz”, ou se depende. Justifique cada item.

(a) Memória (b) Performance (c) Qualidade visual (d) Dificuldade de implementação

2) (1 ponto) Half-edge

Dada a meia-aresta he do desenho abaixo, encontre uma expressão para o vértice v usando a estrutura de dados half-edge com as operações básicas de `face()`, `twin()`, `next()` e `origin()`.



3) (2 pontos) Fórmula de Euler, provas

(a) Prove que não existe uma malha poligonal de uma esfera com 5 vértices e 10 arestas.

(b) Prove que em toda malha poligonal sem triângulos de uma esfera existe um vértice v tal que grau de v é no máximo 3.

4) (2 pontos) Orientação dos vértices

Existe uma convenção de que a orientação dos vértices de um polígono deveria ser anti-horária (CCW).

(a) Para que serve essa convenção? (b) Cite duas vantagens (em modelagem ou em visualização) se os polígonos respeitarem essa convenção.

5) (1.5 pontos) (P3 2013.1)

Dada a função `bool = intersection(vec3 ray_origin, vec3 ray_direction, float radius, vec3 center, float &t1, float &t2)` que retorna verdadeiro/falso se houve interseção entre uma esfera e um raio, assim como guarda os parâmetros dos pontos ordenados de interseção em $t1$ e $t2$.

Implemente a função `bool = SpheresDifferenceIntersection (vec3 ray_origin, vec3 ray_direction, float radiusA, vec3 centerA, float radiusB, vec3 centerB, float &t1)`; que retorna a interseção entre o raio e a diferença de duas esferas (CSG, $A - B$). A função também deve calcular o parâmetro do ponto de entrada do raio.

6) (1.5 pontos) (P1 2012.1)

Dados os seguintes pontos de controle de duas curvas bidimensionais de Bezier, $B1$ e $B2$:

$B1: (0,5) (5,10) (10,0) (15,5)$

$B2: (15,5) (25,15) (30,10) (35,0)$

Considerando a sequência $B1$ e $B2$ como uma única curva, diga:

(a) Em quais pontos a curva cruza os pontos de controle?

(b) Qual é a continuidade da curva final ($C^0, C^1, C^2, C^3 \dots$) e por quê?

Gabarito

1)

(a) Memória

As primitivas implícitas usam menos memória (são melhores) pois com poucos parâmetros descrevem o objeto, enquanto na discretização, precisa-se de uma lista de vértices, adjacências, normais etc...

(b) Performance

Depende. Se forem muitas primitivas pequenas na tela, as implícitas performam melhor. Se forem poucas primitivas ocupando um grande espaço do canvas, as poligonais serão mais eficientes. Isso acontece porque o custo do traçado de raios está no número de pixels.

(c) Qualidade visual

Implícitas são melhores pois tem qualidade com precisão de pixel.

(d) Dificuldade de implementação

A dificuldade de implementação da discretização é baixíssima pois usa o básico do OpenGL para a visualização. A implementação das primitivas implícitas requer algoritmos programados em placa de vídeo (vertex e fragment shaders).

2)

a) $v = \text{Origin}(\text{Next}(\text{Next}(\text{Twin}(\text{Next}(\text{he}))))$

b)

3)(a) Temos que $V = \text{número de vértices} = 5$, $A = \text{número de arestas} = 10$, e temos $HE = \text{número de half-edges} = 2 \cdot A = 20$. Em uma esfera sabemos que $V - A + F = 2$, então $F = 7$. Em uma malha poligonal o número de half-edges por face é maior ou igual a três ($HE/F \geq 3$). Então $F \leq 20/3 = 6.666\dots$. Temos uma contradição porque $F = 7$ e $F \leq 6.666\dots$, então não existe uma malha poligonal com $V=5$ e $A=10$.

(b) Prova por contradição: Assume que todos os vértices tem grau maior ou igual a 4 ($HE/V \geq 4$). Numa malha poligonal sem triângulos temos que o número de half-edges por face é maior ou igual a 4 ($HE/F \geq 4$). Então $2 = V - E + F = V - HE/2 + F \leq HE/4 - HE/2 + HE/4 = 0$, e temos uma contradição.

4) R: Determina a frente e o verso de um polígono. Tratar diferentemente os lados dos polígonos; otimizar a renderização fazendo o back-face culling; cálculo correto da normal; percorrer corretamente a estrutura HE.

5)

```
bool SpheresDifferenceIntersection (vec3 ray_origin, vec3 ray_direction, float radiusA,
vec3 centerA, float radiusB, vec3 centerB, float &t1)
{
    float t1A, t2A, t1B, t2B;

    if ( ! intersection(ray_origin, ray_direction, radiusA, centerA, &t1A,
&t2A) )
        return false;

    if ( ! intersection(ray_origin, ray_direction, radiusB, centerB, &t1B,
&t2B) )
    {
        t1 = t1A;
        return true;
    }

    //se chegou aqui, houve interseção do raio com as duas esferas

    if (t2B < t1A || t1A < t1B)
    {
        t1 = t1A;
        return true;
    }

    // B subtrai inteiramente A
    if (t2B > t2A)
        return false;

    // B subtrai parcialmente A
    t1 = t2B;
    return true;
}
```

}

6)

(a) (0,5) (15,5) (35,0)

(b) C^1 , pois os pontos (10,0) (15,5) (25,15) estão alinhados, mas não são equidistantes (nesse caso seria C^2)

Publicado por [Google Drive](#) – [Denunciar abuso](#) – 5Atualizado automaticamente a cada minutos
