Relatório Trabalho 2 Geometria Computacional

Matheus T. Batista[1], **Davi G. Lazzarin**[1]

[1]Universidade Federal do Paraná,

{mtb21, dgl20}@inf.ufpr.br

1. Introdução

O trabalho consiste em fazer uma implementação da leitura de uma lista de regiões que compõe uma subdivisão planar (malha), testar se é topologicamente bem definida e gerar uma DCEL como saída.

1.1. Entrada

Uma malha que é descrita por um conjunto de N pontos/vértices identificados por um par de coordenadas (x, y). Um conjunto de F faces descritas como a sequência dos vértices em sentido anti-horário.

1.2. Saída

Caso a entrada seja uma subdivisão completa do plano, a saída de dados deve ser um texto com a descrição da estrutura de dados DCEL que representa o poliedro. Caso contrário, será impresso à qual categoria a malha pertence, sendo essas as seguintes:

`Aberta` Caso alguma Semi-Aresta não tenha uma aresta par correspondente (uma *twin*).
`Não Subdivisão Planar` Alguma aresta é fronteira de mais de uma face.
`Superposta` Alguma face tem auto-intersecção ou intersecta outras faces.

2. O que foi feito

2.1. DCEL

É lido os vértices da malha e os mesmos são inseridos em um vetor w. A partir disso é criado um `unordered_map mapa_sa`, onde a chave é um par de índices inteiros [1], sendo esses os índices de w representando o inicio e fim da aresta. Cada semi-aresta em `mapa_sa` é definida como:

```
struct semi_aresta{
    int id;    // id unico da semi-aresta
    int index;    // indice do ponto inicial
    semi_aresta* prox;
    semi_aresta* ante;
    semi_aresta* par;
    face* face_incidente;
};
```

Cada face possui essa estrutura:

[1]Vale notar que feito uma função hash para pares de inteiros, de tal forma que $(a, b) \neq (b, a)$.

```
struct face{
    int id;  // id unico da instancia
    semi_aresta* semi_aresta_inicial;
    u_int32_t quant_lados;
};
```

É criado a semi-aresta (a, b) caso não haja (a, b) dentro de `mapa_sa`. Caso haja, a malha é uma **não subdivisão planar**, portanto o programa para e não é continuado a construção da *DCEL*. Ao receber (a, b), é verificado se já há um (b, a), a fim de inicializar `par`$((a, b))$. Se `face`$((a, b))$ = `face`(`par`$((a, b))$), é indicado que há uma **superposição** — uma face não pode possuir uma semi-aresta s e o seu respectivo par correspondente simultaneamente.

Cada face está contida em uma lista de faces, que depois são apresentadas caso tudo esteja correto.

2.2. Caso Seja Aberto

Para verificar se é aberto, é verificado se todas as semi-arestas da malha face possuem um par correspondente.

2.3. Caso Não Seja Subdivisão Planar

Como já explicado anteriormente durante a construção da DCEL, o fato de haver 2 semi-arestas que começam em v_i e terminam em v_j em faces distintas implica em na existência de uma aresta que é fronteira de 2 faces.

2.4. Caso Seja Superposto

Como já explicado anteriormente durante a construção da DCEL, o fato de haver 2 semi-arestas pares/gêmeas na mesma face implica na existência de uma aresta que intersecciona a mesma face — Um caso superposto.

Por fim, é testado caso alguma semi-aresta cruze outra semi-aresta da malha (usando as funções do trabalho 1) e se alguma face está ”dentro” de outra.

É necessário verificar apenas as faces descritas no sentido anti-horário para testar se uma face está contida no interior de outra. Para isso, é criada uma função que realiza essa verificação com base na orientação dos vértices da face.

Tendo a descrição de duas faces internas, gera-se o centroide de cada face e se caso o algoritmo de *ray-casting* detecte que um ponto está dentro das duas faces, sabe-se que há **superposição**.

2.4.1. Sentido das Faces

A partir da fórmula de *Shoelace*, ou fórmula da área de Gauss definimos se uma face etá descrita no sentido horário ou anti-horário. Ela é definida como:

$$A = \frac{1}{2} \sum_{i=1}^{n} (x_i y_{i+1} - x_{i+1} y_i) \tag{1}$$

$$\text{T.Q } x_{n+1} = x_1 \text{ e } y_{n+1} = y_1$$

2.5. Notas

Foi reutilizado as funções do trabalho 1 como a de *ray-casting* e a que verifica se há interseção entre 2 arestas/segmentos.