

Verificação de Malha

Aginaldo dos Santos Prieto Junior
GRR20221259

André Felipe de Almeida Pontes
GRR20196474

Departamento de Informática
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
aspj22@inf.ufpr.br
afap19@inf.ufpr.br

I. INTRODUÇÃO

O trabalho consiste em implementar a leitura de uma lista de regiões que compõem uma subdivisão planar (malha), testar se ela é topologicamente bem definida e gerar uma DCEL como saída. Para verificar se uma malha é bem definida, primeiro verificamos se alguma aresta é fronteira de menos de duas faces e se alguma face apresenta auto-intersecção ou intersecciona outras faces neste caso, a malha é chamada de sobreposta. Caso a malha passe por todos esses testes, ela é considerada uma malha bem definida.

II. VERIFICAÇÃO DE ARESTAS

A verificação das arestas é realizada por um algoritmo de complexidade $O(N)$. Todas as arestas são percorridas e registradas em um vetor que contabiliza o número de incidências de cada uma. Se uma aresta tiver menos de duas incidências, a malha é considerada aberta. Se tiver mais de duas, a malha é não planar.

III. INTERSECÇÃO DE FACES

A intersecção entre faces é verificada por um algoritmo de complexidade $O(N^2)$, comparando todas as faces do polígono duas a duas. Para cada segmento de uma face, verifica-se se há intersecção com cada segmento da outra face. A detecção de intersecção entre segmentos é feita por meio do cálculo do produto vetorial, representado por um determinante. Se o determinante for maior ou igual a 1, considera-se que os segmentos se intersectam, indicando intersecção entre as faces, sendo assim uma malha superposta. Porém existem algoritmos mais otimizados como o sweep line com complexidade $O(N \log n)$.

IV. DCEL

A DCEL é uma estrutura de dados que representa uma subdivisão planar por meio de seus elementos básicos: vértices, arestas e faces. O diferencial da DCEL é a decomposição das arestas em semi-arestas orientadas (também chamadas de half-edges), que permitem o acesso eficiente às relações de adjacência entre os elementos.

- Vértice de origem;
- Semi-aresta oposta;

- Próxima semi-aresta;
- Semi-aresta anterior;
- Face associada.

Para a implementação da DCEL foram criadas três structs: Vertex, que contém o número do vértice, suas posições x e y , e um ponteiro para a semi-aresta incidente; a struct HalfEdge, que contém o número da semi-aresta, o vértice de origem, um ponteiro para a próxima semi-aresta, um ponteiro para a semi-aresta oposta, um ponteiro para a semi-aresta anterior e um ponteiro para a face associada; e a struct Face, que contém o número da face e um ponteiro para a semi-aresta que contorna essa face.

Assim, para cada vértice de cada face, criamos uma semi-aresta com vértice de origem igual ao vértice atual e face associada à face atual. Para definir a próxima e a semi-aresta anterior, o vértice de índice i tem a próxima semi-aresta definida como $(i+1) \bmod \text{número de vértices}$, e a semi-aresta anterior da posição $(i+1) \bmod \text{número de vértices}$ é definida como i . A semi-aresta oposta de uma semi-aresta é aquela que possui a mesma origem e destino, mas com a direção invertida. A face associada a uma semi-aresta é a face que está à sua esquerda.