Relatório Trabalho 1 Geometria Computacional

Matheus T. Batista[1], **Davi G. Lazzarin**[2]

[1]Universidade Federal do Paraná

{mtb21, dgl20}@inf.ufpr.br

1. Introdução

Esse trabalho consiste em responder o problema *classificação de polígonos e pontos interiores*, que é definido como:

Instância: um conjunto de linhas poligonais fechadas $\mathcal{S}$ e um conjunto de pontos $\mathcal{P}$.

Resposta: uma classificação de cada um dos polígonos em ”simples”, ”não simples”, ”simples e convexo” e ”simples e não convexo” e em seguida uma lista dos polígonos em $\mathcal{S}$ que contém o ponto p, para todo $p \in P$.

Pode-se dividir a resolução do problema em 3 outros subproblemas.

- Dado um polígono, responder se o mesmo é simples ou não.
- Dado um polígono simples, dizer se o mesmo é convexo ou não.
- Dado um ponto e um polígono, dizer se o polígono contém o ponto.

Foi combinado 3 algoritmos para responder o problema principal e cada algoritmo resolve um dos 3 subproblemas.

2. O que foi feito

2.1. Polígonos

Primeiro, para cada polígono $s \in S$ verificamos a sua simplicidade.

2.1.1. É simples?

Para responder isso, foi feito um algoritmo de força bruta que é de ordem $\mathcal{O}(n^2)$. Segue o passo a passo: É recebido um polígono A de n vértices. Para cada aresta $a \in A$ é testado se há interseção com as outras $n-1$ arestas existentes no polígono e caso haja, já sabe-se que o polígono não é simples. É trivial perceber que a quantidade de testes realizados no pior caso é de ordem quadrática.[1]

Para testar isso, foi usado um array de arestas e passa-se por cada aresta $a \in A$ e verifica com o restante das arestas. Se já foi testado que não há interseção entre as arestas a_i e a_j, não é preciso testar se a aresta a_j tem interseção com a_i (diminui cálculos atoa mas não diminui a limitação quadrática do algorítmo).

É olhado apenas se se há interseção em um ponto em arestas não conseguintes, caso sejam conseguintes verifica se há interseção em um intervalo de pontos.

Após isso, verifica-se se o polígono é convexo ou não.

[1]Caso não seja trivial, note-se que é feito primeiro $n-1$ verificações de interseção, depois $n-2, n-3$... até 1, como o teste de interseção é $\mathcal{O}(1)$, o que temos é $\sum_{i=1}^{n-1} i = \frac{((n-1)^2+(n-1))}{2}$.

2.1.2. É convexo?

Para testar a convexidade do polígono, analisa-se o sinal de cada ângulo formado por cada par de arestas:

1. calcula-se o produto vetorial de cada conjunto de arestas vizinhas
2. se houver pelo menos um produto vetorial com o sinal diferente do restante dos pares de arestas, então polígono não pode ser convexo.

2.2. Pontos

Agora, para cada ponto $p \in \mathcal{P}$ enumeramos os polígonos simples que contém p.

Foi utilizado o método de *ray casting* para verificar se o ponto reside no polígono.

2.2.1. Ray Casting

O algoritmo de Ray Casting consiste em contar quantas vezes uma semirreta que parte do ponto p cruza as arestas do polígono, a fim de determinar se o ponto está dentro ou fora da região. Caso a quantidade de vezes for **ímpar**, então o ponto está **dentro** do polígono, caso seja **par** então o ponto está **fora**. Vale descrever algumas convenções adotadas no método:

Tratamento das arestas:

- Cada aresta é representada como um par ordenado $< p_{inicio}, p_{fim} >$, com os pontos organizados de forma que $p_{inicio.y} \leq p_{fim.y}$;
- A aresta é considerada apenas se não for horizontal, ou seja, se $p_{inicio.y} \neq p_{fim.y}$. São explicitamente ignoradas para evitar contagens ambíguas;
- Para cada aresta válida, verifica-se se a coordenada y do ponto p está dentro do intervalo aberto $(p_{inicio.y}, p_{fim.y})$. Caso não esteja, a aresta é desconsiderada;
- Se p_y está dentro do intervalo da aresta, é verificado se o ponto está à esquerda da aresta — apenas nesse caso ela é contabilizada como interseção.

Tratamento de vértices sobre a semirreta:

- Quando o ponto p está exatamente sobre um vértice v, esse caso é tratado separadamente para evitar ambiguidade na contagem.
- Consideram-se os vértices vizinhos de v, denotados por a (antecessor) e c (sucessor).
- Se $a_y < v_y < c_y$, considera-se que há uma interseção.
- Se v_y for simultaneamente maior ou menor que ambos os vizinhos ($v_y < a_y$ e $v_y < c_y$, ou $v_y > a_y$ e $v_y > c_y$), não é considerada interseção.
- Se $v_y = c_y$, o vértice é ignorado e passa-se para o próximo.
- Se $v_y = a_y$, percorre-se os antecessores de a até encontrar um vértice com coordenada y distinta de v_y, e então retoma-se a análise do caso com os novos vizinhos;
- Caso o ponto esteja sobre a aresta, então conta-se mais uma interseção.