

# Relatório Trabalho 1 - Triangulação de Polígono Simples

Gabriel d A. S. Evaristo<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)

almeidasalesgabriel@gmail.com

## 1. Introdução

O trabalho consistia na implementação de um algoritmo para triangular um polígono dado, para tal foram seguidas as instruções do capítulo 3 do livro-texto da disciplina [De Berg et al. 2008]. Os passos do algoritmo proposto no livro eram: decompor o polígono em partes monótonas e triangular cada uma das partes. Logo, para resolver o problema proposto pelo professor utilizando a sugestão do livro e devolver a saída como especificada, era necessário: ler o polígono da entrada padrão e codificá-lo de maneira interessante, decompor o polígono em partes monótonas, triangular cada parte separadamente, combinar as partes trianguladas e gerar a saída especificada.

## 2. Implementação

Seguindo o processo explicado pelo professor, o objetivo inicial era simplificar o problema e obter uma solução para essa versão, atualizando0a depois com os casos degenerados. O primeiro passo escolhido foi implementar a estrutura de dados para representar os polígonos, foi escolhida a Doubly Connected Edge List descrita no livro-texto [De Berg et al. 2008], com algumas modificações para se adequar a especificação, após implementá-la o passo da leitura e instanciação da estrutura foram os seguintes. Com isso completo o próximo passo foi implementar a triangulação de um polígono y-monótono (para triangular cada pedaço decomposto), essa implementação tomou mais tempo mas uma vez terminada era possível entrar com um polígono estritamente y-monótono e obter o conjunto de diagonais que deveriam ser adicionadas para triangulá-lo. A partir desse ponto foram tomadas algumas escolhas de implementação que se mostraram negativas no fim do trabalho, o que acarretou em sua incompletude. Era necessário a partir da adição dessas diagonais, definir os triângulos para poder representá-los na saída especificada, tomando como referência a estrutura de triangulação sugerida na especificação e as faces descritas no livro-texto [De Berg et al. 2008] foi implementado uma estrutura Triângulo, criada logo depois da adição de uma diagonal na triangulação do polígono y-monótono. Essa estrutura facilitava a impressão da triangulação da maneira como foi especificada, porém, a instanciação dos triângulos não era correta nos casos degenerados, i.e, em algumas instâncias de polígonos y-monótonos com arestas horizontais. A adição dos triângulos foi implementada usando o fato de que a cada diagonal adicionada um triângulo é cortado da área do polígono. No algoritmo descrito no livro existem dois casos onde diagonais são adicionadas, diferenciados pelas posições dos vertices origem e de destino em relação a fronteira esquerda e direita do polígono. A falhas tentativas de implementação da adição de triângulos nos casos degenerados consumiu o tempo de execução do trabalho e impossibilitou a implementação da decomposição do polígono simples em partes y-monótonas.

### 3. Resultado

O algoritmo é capaz de gerar a saída especificada e sendo assim, uma visualização correta no algoritmo de desenhar a triangulação, para polígonos estritamente y-monótonos. Em alguns casos y-monótonos como o da Figura 1 a visualização apresenta uma diagonal fora do polígono (o que se deve a lógica de adição de triângulos, pois as únicas diagonais adicionadas pelo algoritmo foram  $[2,1] \rightarrow [5,2]$  e  $[1,0] \rightarrow [5,2]$ ).

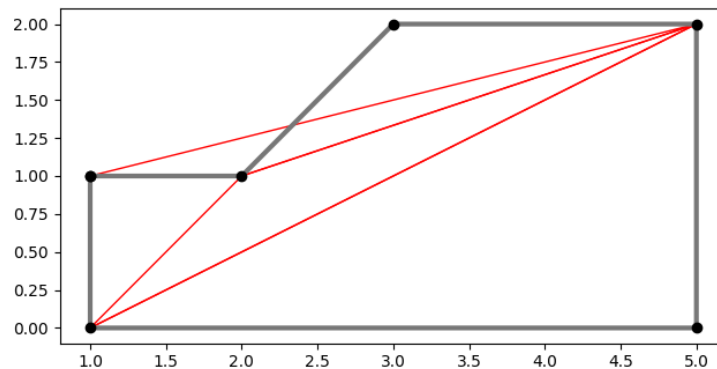


Figura 1. Exemplo 1 de caso degenerado.

No exemplo da Figura 2 a visualização está correta, mas isso também se deve a lógica errônea de adição de triângulos, pois as diagonais adicionadas pelo algoritmo são:  $[1, 1] \rightarrow [3,2]$ ,  $[3,1] \rightarrow [1,1]$  e  $[1,0] \rightarrow [3,1]$ .

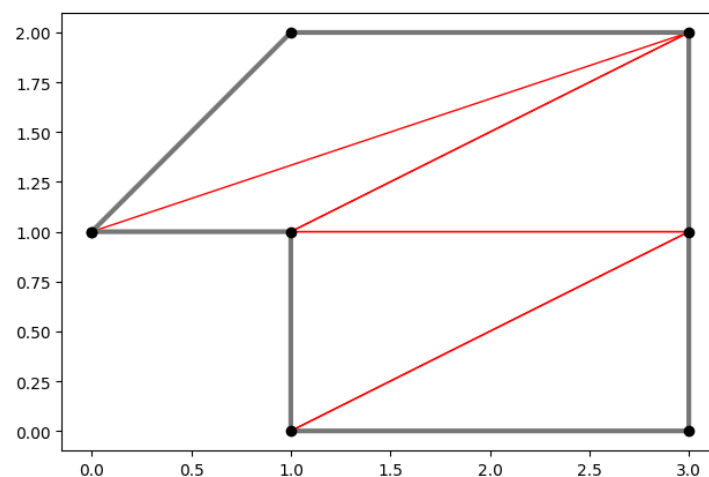
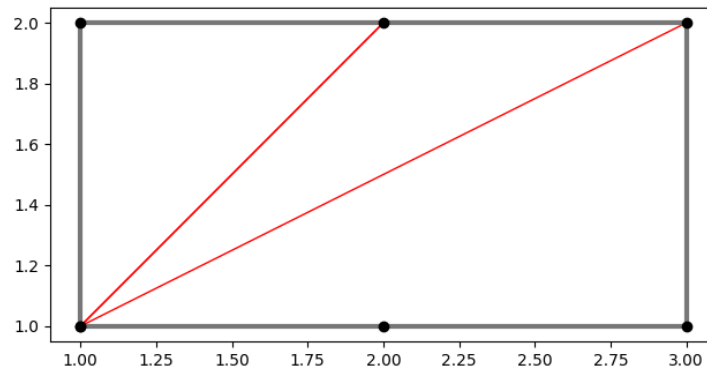


Figura 2. Exemplo 2 de caso degenerado.

O exemplo da Figura 3 mostra um caso onde tanto a visualização quanto as diagonais adicionadas ( $[1,1] \rightarrow [3,2]$ ,  $[1,1] \rightarrow [2,2]$  e  $[3,1] \rightarrow [1,1]$ ) estão erradas:



**Figura 3. Exemplo 3 de caso degenerado.**

#### **4. Próximos passos**

O que julga-se necessário para resolver o problema do algoritmo é mudar a maneira como a Doubly Connected Edge List foi implementada, criando funções mais claras de adição de vértices e arestas, tratando os diversos casos possíveis e adicionando faces novas quando a adição de uma aresta resultar em um ciclo.

Um próximo projeto interessante seria integrar a geração da visualização do polígono e suas diagonais com o algoritmo de triangulação, plotando cada passo do algoritmo e usando um script em bash para unir esses frames em um vídeo, obtendo assim uma animação da adição das diagonais passo a passo.

#### **Referências**

De Berg, M., Cheong, O., Van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition.