Trabalho Prático 1 - Algoritmos II

Augusto Maillo Queiroga de Figueiredo

[1]Universidade Federal de Minas Gerais

[2]Belo Horizonte - MG

augusto.maillo@gmail.com

1. Introdução

Este trabalho consiste em solucionar o problema da galeria de arte utilizando o algoritmo Ear Clipping, além de provar uma visualização interativa dos passos envolvidos na solução.

O problema da galeria de arte consiste em instalar o menor número de câmeras possível em uma galeria de arte, de forma que todos os cômodos possam ser vigiados por elas. Como a galeria pode ser modelada como um polígono, este problema pode ser resolvido utilizando algoritmos geométricos, porém a solução utilizando o algoritmo Ear Clipping não é capaz de produzir a resposta ótima para todos os casos.

A solução que será construída neste trabalho consiste em três passos principais:

- Triangular o polígono: dividiremos o polígono que representa a galeria em diversos triângulos.
- Montar um grafo com os triângulos: os triângulos obtidos irão compor um grafo onde eles são os vértices e as arestas deste grafo representam se dois triângulos compartilham uma aresta.
- Realizar uma 3-coloração do grafo obtido. Com isso, os vértices coloridos com a cor que aparecer o mínimo neste grafo podem ser interpretados como as câmeras a serem posicionadas a fim de monitorar todo a galeria.

2. Implementação

Todos os passos necessários para obter a solução e gerar visualizações interativas foram encapsulados em classes.

Inicialmente, foram implementadas uma classe e funções auxiliares. Esta classe, Point, tem o objetivo de facilitar as operações com pontos e consequentemente, vetores. Foram implementadas operações matemáticas, operadores relacionais e outros métodos auxiliares para essa classe. Já as funções auxiliares tem o objetivo de converter estruturas dados para maior conveniência (*e.g.* converter listas) e deixar o código modular (*e.g.* checar se um ponto pertence a um triângulo).

Em seguida foi implementada a classe que efetivamente gera a triangulação do polígono utilizando Ear Clipping. Esta classe realiza a triangulação no próprio construtor, fazendo uso de métodos auxiliares para decidir se um ponto é uma ponta de orelha e controlar os pontos que já foram analisados. Além disso, essa classe possui um método que reconstrói as informações de passo a passo da execução do algoritmo a fim de gerar uma visualização interativa posteriormente. Este método retorna três listas, uma contendo os segmentos a serem plotados em um passo i, outra contendo os pontos e uma

lista contendo mensagens informando o que está acontecendo a cada passo. Junto com isso, os pontos e segmentos possuem informações de cor, com o objetivo de gerar uma visualização sobretudo intuitiva.

Para o próximo passo foi implementada uma classe que representa uma árvore gerada com os triângulos obtidos no passo anterior. Esta classe, além de performar uma busca em profundidade nesta árvore para obter a coloração, gera uma lista com as cores de cada vértice a cada passo da busca. Com isso, conseguimos visualizar como este algoritmo percorre na árvore.

Para o último passo, da visualização, foram implementadas duas classes, uma que controla qual passo da solução deve ser exibido, Index, e outra que consiste no controlador efetivo da visualização interativa, Plotter. Esta última define os botões, as configurações de layout, conecta callbacks para cada ação e de fato gera a visualização do polígono na tela. A classe Index foi utilizada para permitir que o usuária consiga navegar livremente pelos passos da solução.

Tendo todas essas classes e funcionalidades, uma função final combina todos os passos. Primeira os pontos, recebidos como uma lista de tuplas, são verificadas a fim de decidir se já estão na ordem horária, necessária para que o algoritmo funcione corretamente. Em seguida, o algoritmo Ear Clipping é executado e posteriormente a coloração. As soluções de ambos são unidas em listas de pontos, polígonos e mensagens para cada passo. O menor número de vértices de uma mesma cor, solução do problema, é então computado de forma simples, e os passos são informados para a classe Plotter que gera a visualização.

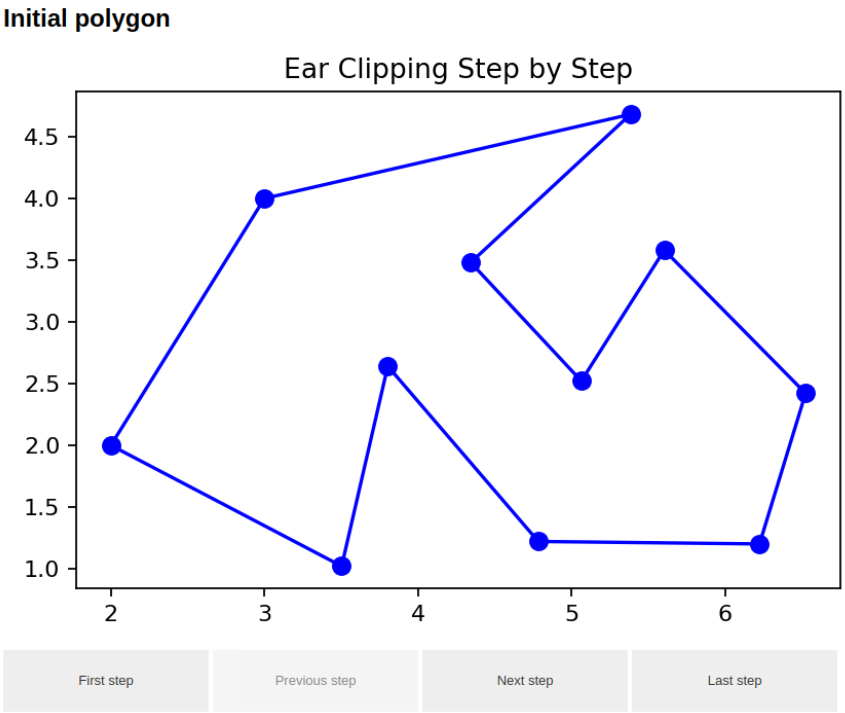

Figura 1. Passo inicial da solução.

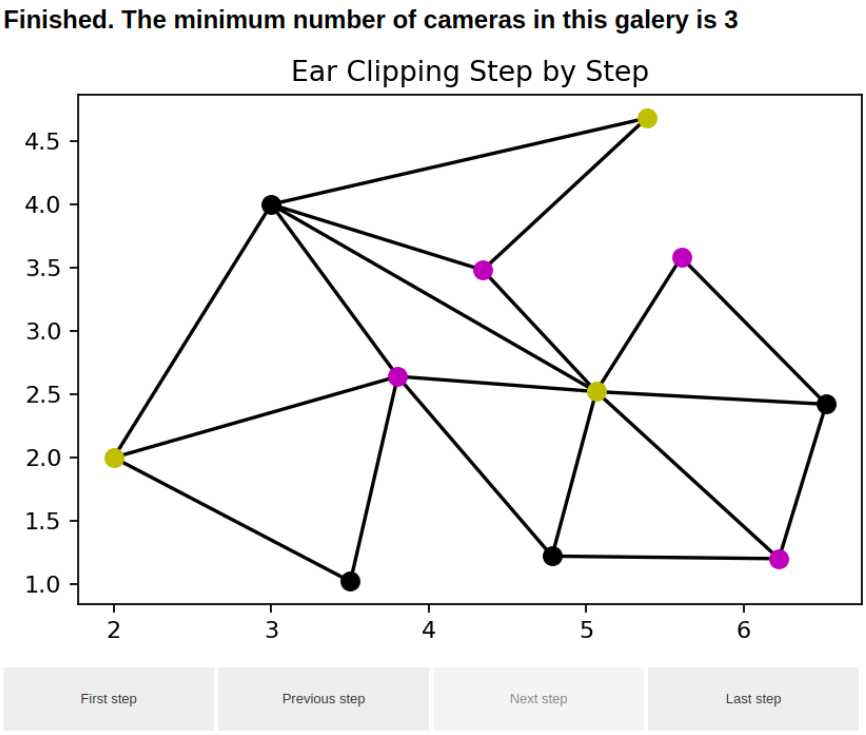

Figura 2. Passo final da solução.

3. Testes

Para testar a implementação foram submetidos diversos polígonos. Nesta seção serão mostrados 3 deles, representando um caso geral, um polígono convexo e um polígono não convexo com uma grande quantidade de pontos.

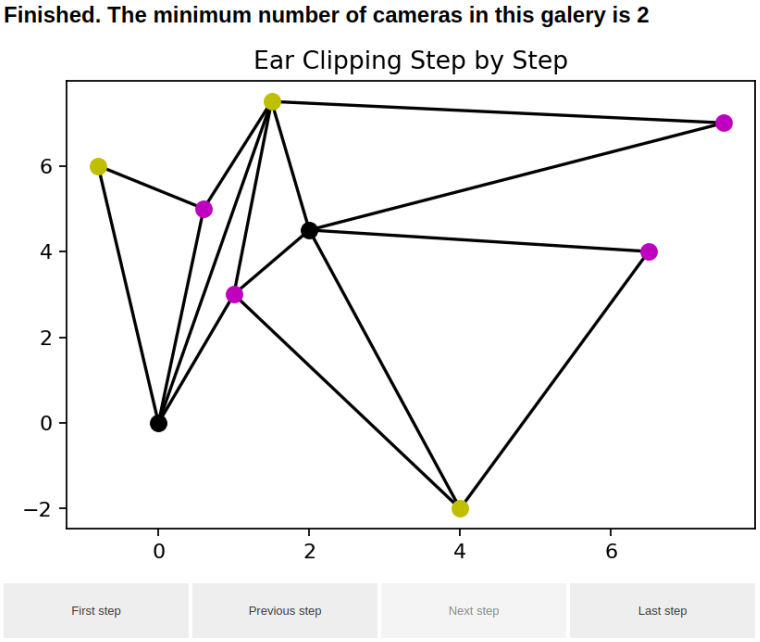

Figura 3. Polígono geral

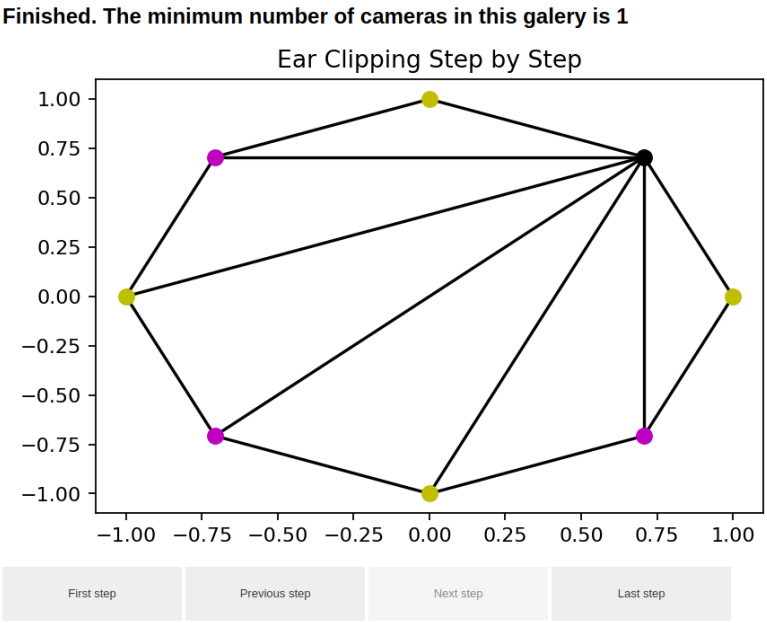

Figura 4. Polígono convexo.

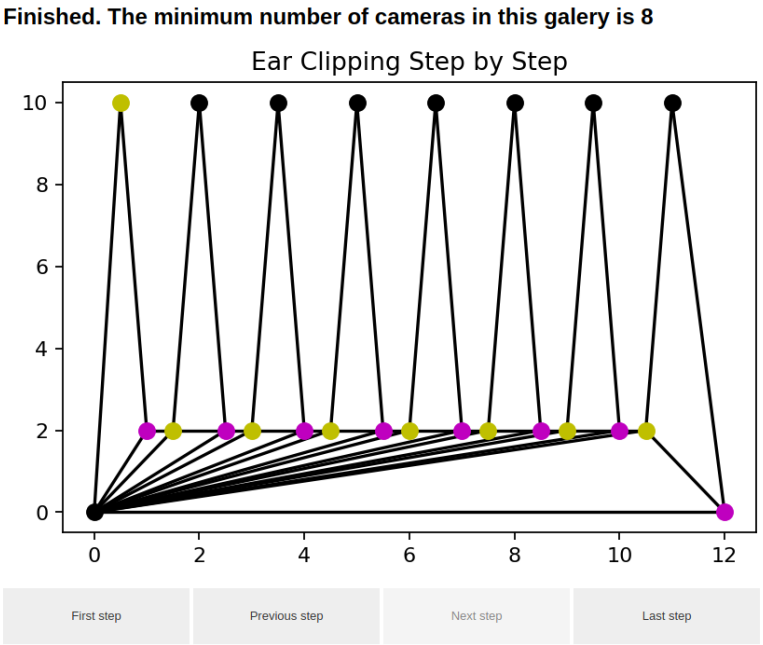

Figura 5. Polígono não convexo com muitos pontos.

4. Conclusão

Neste trabalho foi implementada uma solução para o problema da galeria de arte, de forma que a execução do algoritmo pudesse ser visualizada de forma interativa. O desenvolvimento do trabalho permitiu consolidar os conceitos e métodos vistos ao longo das aulas, como por exemplo, verificar se um ponto pertence a um triângulo ou qual a mudança de direção em dois segmentos consecutivos.

Além disso, construir uma visualização interativa permitiu entender melhor o passo a passo da execução do algoritmo, visto que foi necessário compreender cada um deles de forma a gerar uma exibição intuitiva para o usuário.