

Trabalho Prático 01

Algoritmos I

Otávio Maciel Munaier Zucheratto
Nº: 2019041612

Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brasil

ommz21@ufmg.br

1. Relatório

1.1. Implementação

O problema proposto para esse trabalho prático consiste resolver o problema da galeria de arte em duas dimensões, isto é, dado um polígono simples no plano devemos encontrar o número ótimo de câmeras que consiga cobrir todo o interior do polígono. Vimos em aula que o algoritmo ear-clipping resolve esse problema com complexidade de tempo $O(n^2)$.

Foi implementado uma versão do ear-clipping na linguagem Python3, o funcionamento do algoritmo pode ser descrito da seguinte maneira:

- Deve ser fornecido como entrada do algoritmo um conjunto de pontos que correspondem aos vértices do polígono, de maneira que os pontos serão armazenados em uma lista de tuplas correspondente as coordenadas do vértice em questão.
- A lista de pontos deve ser tal que se dois pontos são consecutivos, existe uma aresta entre eles e deve existir uma aresta entre o primeiro e o último ponto. Além disso, os pontos devem ser fornecidos em ordem anti-horária, como no exemplo abaixo:

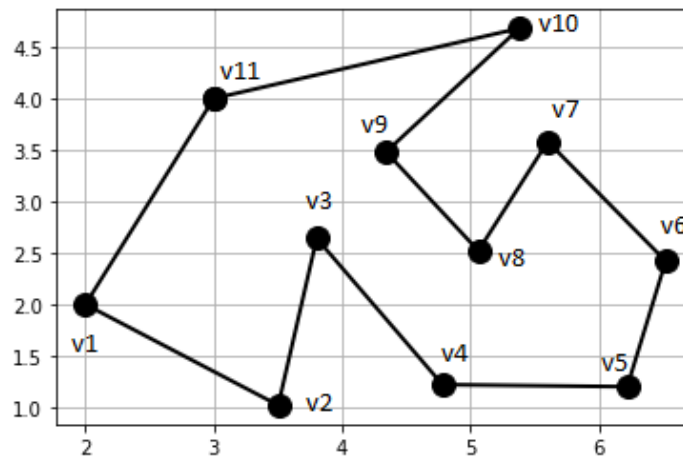


Figura. 1. Exemplo de um polígono válido

- Após a entrada de um polígono de n vértices, o algoritmo realiza o processo de triângulação, que consiste em dividir o polígono em $n - 2$ triângulos. Para tal será necessário percorrer todos os vértices do polígono aparando as "orelhas" do mesmo.
- As "orelhas" ocorrerão quando houver uma guinada para a esquerda a partir de um vértice em relação ao próximo vértice no sentido anti-horário. A orelha só poderá ser aparada e delimitar uma diagonal no polígono, se uma vez verificado que todos os demais vértices não se encontram dentro do triângulo formado pelos vértices v_{n-1}, v_n e v_{n+1} . Caso contrário deve seguir para o próximo vértice
- Quando restarem apenas três vértices, o algoritmo se encerra com a triângulação desejada, conforme o exemplo abaixo:

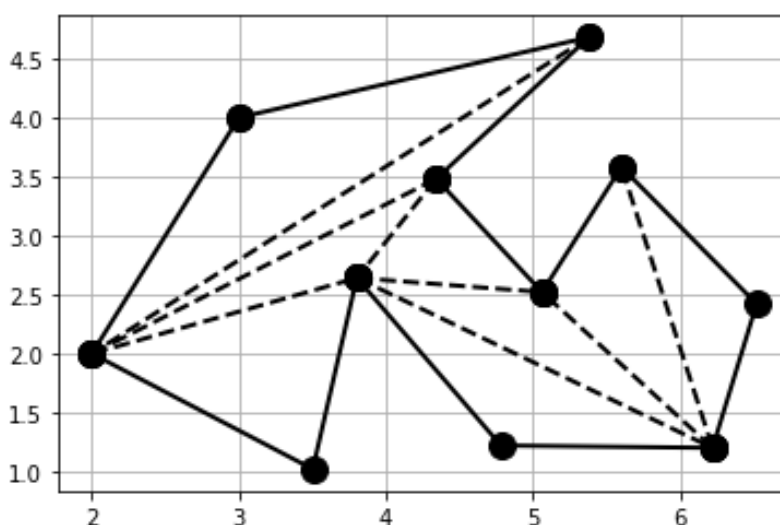


Figura. 2. Resultado do Ear-Clipping

Uma vez que a triângulação esteja completa, deve-se construir um grafo dual $G(V,E)$ de tal forma que os vértices serão os triângulos construídos pelo ear-clipping, e existirá uma aresta entre dois vértices caso exista uma diagonal em comum entre os triângulos que os vértices representam.

Para construir o grafo dual verificamos para cada um dos $n - 2$ triângulos, quais são os triângulos que compartilham exatamente dois vértices e então uma aresta deve ser criada. Logo a complexidade da criação do grafo dual terá um custo $O(n^2)$.

Então faremos uma busca em profundidade partindo de qualquer vértice do grafo dual, de maneira que o caminho percorrido pela busca é guardada. Portanto para resolver o problema, basta colorir os vértices do grafo dual a partir dos caminho obtido pela busca, sendo que dois triângulos adjacentes possuem dois vértices em comum. Logo basta colorir o vértice que diferencia os triângulos com uma terceira cor, resultando em um grafo colorido com três cores.

Vimos em aula que ao tomar qualquer uma dentre as três cores, com câmeras dispostas nos vértices que possuem essa tal cor, teremos uma cobertura completa do polígono com a quantidade ótima de câmeras. Abaixo se encontra a coloração produzida pelo algoritmo com o polígono visto anteriormente:

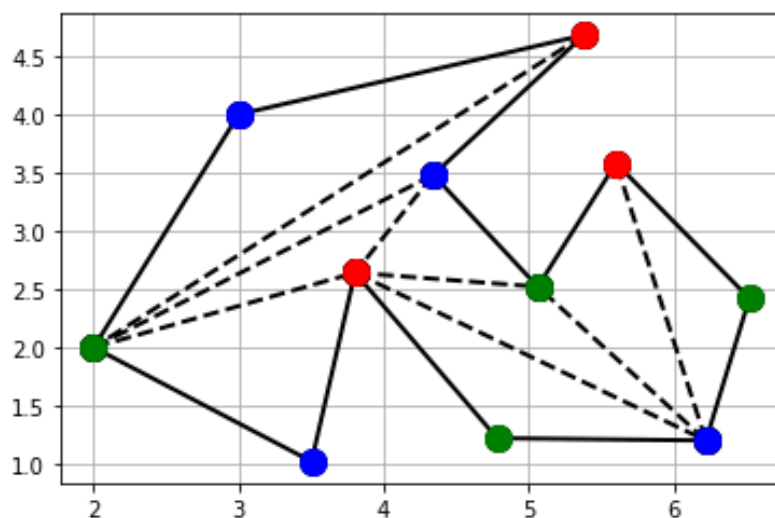


Figura. 3. Resultado do Ear-Clipping

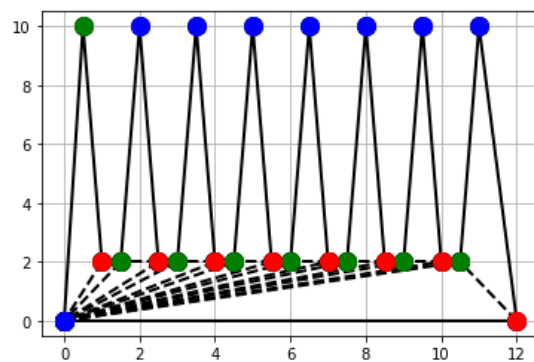
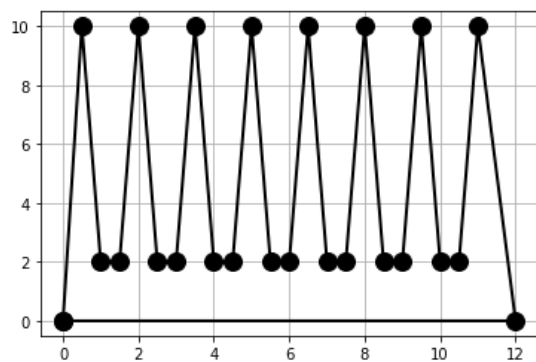
1.2. Estrutura de Dados

As principais estruturas utilizadas ao longo desse trabalho estão descritas abaixo:

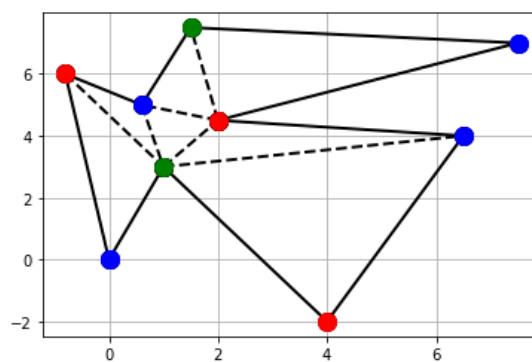
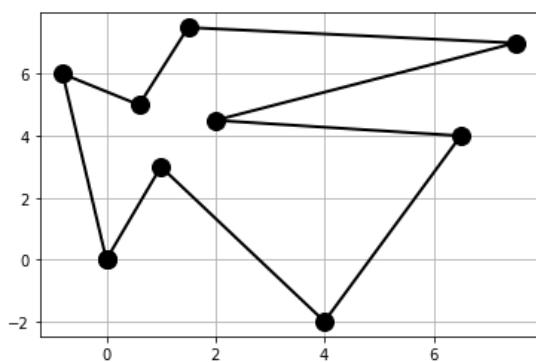
- **Pontos:** Os pontos fornecidos são armazenados em uma lista conforme especificado anteriormente, sendo que a ordem diz respeito a como o polígono é formado.
- **Cor:** Corresponde a um dicionário que possui como chave os respectivos vértices do polígono, a cor é fundamental para a visualização da etapa de coloração ao final do algoritmo.
- **Grafo:** Foi criada uma classe que representa o grafo dual para permitir uma busca em profundidade entre os triângulos que particionam o polígono, nessa classe as arestas entre vértices estão armazenadas em listas de adjascências.
- **Triangulos:** É uma lista de listas de tuplas, essa estrutura auxilia a criação do grafo dual, possibilitando uma maneira de verificar quais triangulos são adjacentes e devem possuir conexão no grafo dual.
- **Diagonais:** Estrutura auxiliar para armazenar os vértices que possuem uma diagonal entre eles, é utilizada pela função de visualização da triangulação do polígono.

1.3. Testes

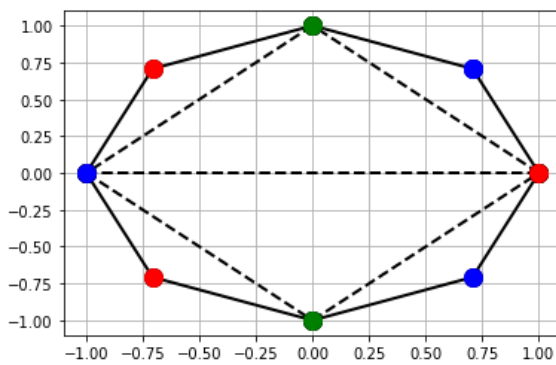
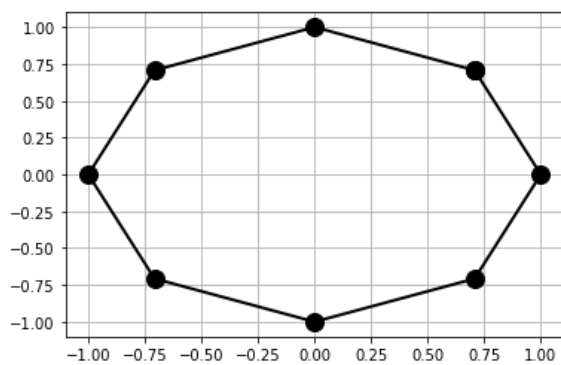
Foram disponibilizados quatro testes para verificar a corretude do código implementado nesse trabalho prático, algumas adaptações se fizeram necessárias quanto a ordem da entrada dos pontos conforme as restrições explicadas anteriormente. O algoritmo executou conforme o esperado, abaixo se encontram as saídas fornecidas pelo programa:



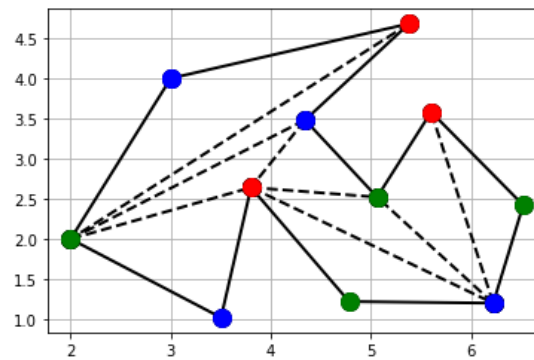
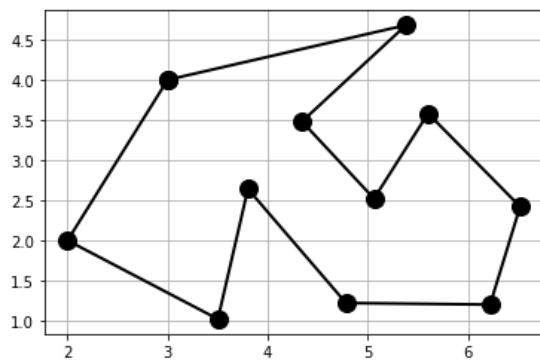
Teste 1



Teste 2



Teste 3



Teste 4

2. Bibliografia

[R. Vimieiro](2021). Slides da disciplina Algoritmos II. DCC. Universidade Federal de Minas Gerais (UFMG).