

## Algoritmo de construção do eixo medial de polígonos retilineares simples

Márcio Peixoto<sup>1</sup>

PUC-Rio, Rio de Janeiro, RJ

Felipe de Oliveira<sup>2</sup>

PUC-Rio, Rio de Janeiro, RJ

Um polígono é chamado *retilinear* quando dois lados adjacentes são sempre perpendiculares. Neste trabalho, consideramos apenas polígonos retilineares simples, isto é, polígonos retilineares tais que dois lados se intersectam se, e somente se, são adjacentes. Para este tipo de polígono, o Teorema da Curva de Jordan garante que podemos definir o interior e o exterior do polígono, e que estas regiões são desconexas (Ver Figura 1 (esquerda)).

Dizemos que um quadrado  $Q$  é *bitangente* a um polígono retilinear simples  $P$  quando dois ou mais lados de  $Q$  intersectam  $P$  em pontos que não são vértices do quadrado, e  $Q$  não intersecta o exterior de  $P$  (Ver Figura 1 (direita)).

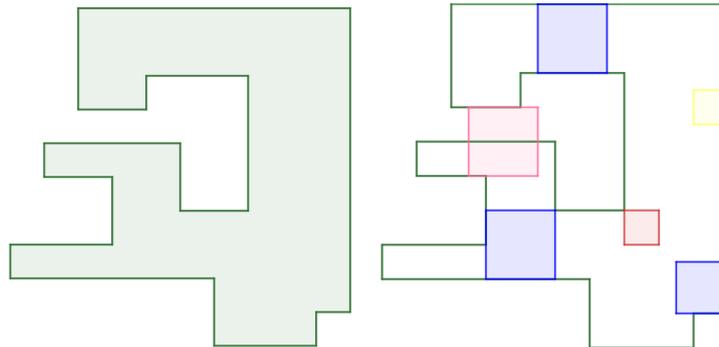


Figura 1: *Esquerda: Exemplo de polígono retilinear simples, em verde escuro, com interior em verde claro. Direita: Exemplos de quadrados bitangentes (azul) a um polígono retilinear simples (verde). Os demais quadrados (rosa, vermelho e amarelo) não são quadrados bitangentes.*

Definimos o *eixo medial* (MA) de um polígono retilinear simples  $P$  como o fecho do lugar geométrico dos centros dos quadrados bitangentes a  $P$  (Ver Figura 2, esquerda). O eixo medial de um polígono simples é uma árvore cujas arestas são segmentos de reta.

Neste trabalho, desenvolvemos um algoritmo, em C++, que constrói o MA de qualquer polígono retilinear simples que não possui duas arestas colineares.

Seja  $P$  um polígono retilinear simples que satisfaz a condição acima. O algoritmo constrói o MA de  $P$ , começando no vértice  $V_1$  do polígono que está mais ao sul, dentre os mais à esquerda (isto é, no vértice com menor coordenada  $y$ , dentre os com menor coordenada  $x$ ). Deste vértice, sempre parte

<sup>1</sup>marciodosgpeixoto@gmail.com

<sup>2</sup>felipedeoliveira1407@gmail.com

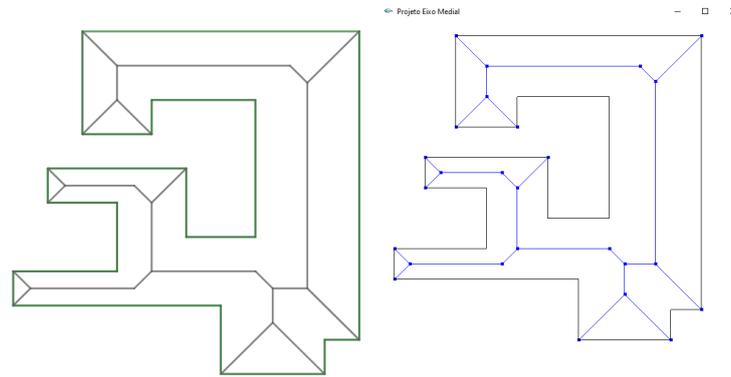


Figura 2: *Esquerda: Eixo medial (em preto) do polígono da Figura 1, construído através do GeoGebra. Direita: Eixo medial (em azul) do polígono da Figura 1, construído através do algoritmo.*

uma única aresta  $a_1$  do MA. Para determinar  $a_1$ , basta localizar sua outra extremidade  $V_2$ . Para isto, realiza-se um teste, chamado *Teste da Bissetriz Crescente* (que no algoritmo, corresponde a três funções recursivas). No final desse teste,  $V_2$  estará determinado, assim como as direções das arestas do MA que partem de  $V_2$ . Escolhe-se arbitrariamente uma delas, e então repete-se o processo para localizar a outra extremidade  $V_3$  dessa aresta escolhida.

Eventualmente, chega-se num vértice do MA do qual não parte nenhuma aresta ainda não determinada (são aqueles que também são vértices de  $P$ ). Neste momento, o algoritmo volta para o último vértice do MA do qual partem arestas ainda não determinadas, e repete o processo descrito acima para uma dessas arestas desconhecidas. Esta parte do algoritmo é uma aplicação do algoritmo *Deep First Search* (DFS) [1]. Ao final da execução do algoritmo, o MA de  $P$  estará completamente determinado. Seu plot é feito, aresta por aresta, ao longo de toda sua construção (Ver Figura 2, direita).

Comparado a outros algoritmos de construção do eixo medial já existentes (tais como [2]), este não é o que possui menor complexidade computacional. Porém, ele é determinístico, no sentido de que cada polígono tem seu MA bem definido pelo algoritmo. Além disso, acreditamos que o algoritmo ainda pode ser aprimorado.

## Agradecimentos

Agradecemos ao professor Marcos Craizer (PUC-Rio), nosso orientador, pela liderança e atenção ao nos guiar neste projeto; e à doutoranda Tamires Pereira (PUC-Rio) pela participação ativa e ajuda técnica que foram essenciais para a criação do algoritmo.

## Referências

- [1] Even, Shimon (2011), Graph Algorithms, ISBN 978-0-521-73653-4 2nd ed., Cambridge University Press, pp. 46–48.
- [2] Chin, F., Snoeyink, J. Wang, C. Finding the Medial Axis of a Simple Polygon in Linear Time . Discrete Comput Geom 21, 405–420 (1999). <https://doi.org/10.1007/PL00009429>