

**CLASSIFICAÇÃO DE PONTOS EM RELAÇÃO A POLÍGONOS E POLIEDROS.** Rafael Xavier de Lima, Marco Antônio Piteri. – Exatas - Ciência da Computação – Departamento de Matemática, Estatística e Computação – Faculdade de Ciências e Tecnologia – Campus de Presidente Prudente.

Ao utilizar qualquer aplicativo orientado por uma interface gráfica, o usuário interage com o sistema através de *ícones* e das opções existentes na *barra de menus* objetivando selecionar ações a serem realizadas. Na verdade este processo de seleção se dá através do ato de pressionar um botão do mouse numa área retangular associada à opção desejada. Neste caso em particular, todas as regiões poligonais são retangulares e o algoritmo de classificação está implementado no nível de *hardware* e se reduz a fazer comparações entre as coordenadas de um *ponto teste* com relação aos limites horizontais e verticais do retângulo desejado.

A situação descrita acima é um dos exemplos clássicos de classificação de pontos em relação a outros objetos geométricos, entretanto, ele ocorre ainda em aplicações mais complexas ligadas às áreas de Computação Gráfica, Cartografia, Modelagem de Sólidos, Geometria Computacional e de CAD/CAM, onde as regiões são mais arbitrárias do que simples retângulos. Nesse sentido, nosso objetivo no contexto deste trabalho é estudar as diferentes abordagens algorítmicas existentes para resolver o problema de classificação de pontos em relação a conjunto de pontos com uma estrutura combinatória bem definida. Mais formalmente, dado um ponto  $p$  e um polígono ( $\mathbb{R}^2$ ) ou um poliedro ( $\mathbb{R}^3$ )  $P$  arbitrário, nosso propósito é verificar se  $p$  pertence ao interior (inclusive a fronteira) ou ao exterior de  $P$ .

É importante salientar que estamos em busca de princípios algorítmicos gerais, independentes de eventuais características relativas à  $P$ . Por exemplo, existem algoritmos específicos para tratar somente o caso em que  $P$  é convexo. A convexidade associada a conjunto de pontos é uma propriedade importante, pode e deve ser explorada, mas infelizmente o mundo real não é feito de objetos convexos.

Os algoritmos existentes para resolver o problema no caso de  $P$  ser convexo, aproveitam-se do fato de que, uma reta em  $\mathbb{R}^2$  ou um plano em  $\mathbb{R}^3$  induzem uma subdivisão do respectivo espaço em três subconjuntos. Logo, a reta suporte (caso do polígono) ou o plano suporte (caso do poliedro) associado a quaisquer dos lados de  $P$ , faz com que todos os outros pontos de  $P$  estejam de um mesmo lado. Assim, uma primitiva geométrica que seja capaz de classificar o lado em que um ponto se localiza em relação a uma reta, e, analogamente, em relação a um plano, é suficiente para resolver o problema. Em  $\mathbb{R}^2$ , a primitiva  $Ccw(p_1, p_2, p_3)$  dada pela equação (1), nos diz se o ponto  $p_3$  está à esquerda ou a direita do segmento orientado  $\overrightarrow{p_1 p_2}$ , ou ainda, se os três pontos são colineares.

Devemos observar que na verdade não estamos interessados na magnitude do determinante da equação (1), mas apenas em seu sinal. A generalização desta primitiva para o  $\mathbb{R}^3$ , para classificar pontos em relação a planos, é imediata.

$$Ccw(p_1, p_2, p_3) = \begin{vmatrix} p_{1x} & p_{1y} & 1 \\ p_{2x} & p_{2y} & 1 \\ p_{3x} & p_{3y} & 1 \end{vmatrix} = \begin{vmatrix} p_{1x} - p_{3x} & p_{1y} - p_{3y} \\ p_{2x} - p_{3x} & p_{2y} - p_{3y} \end{vmatrix} \quad (1)$$

Entre as abordagens mais genéricas e capazes de trabalhar com polígonos simples arbitrários, encontram-se os algoritmos de *intersecção ímpar-par*, *índice de rotação* (*winding number*) e *leque de triângulos*. Vamos descrever as duas últimas inicialmente e detalhar em maior

profundidade a primeira, já que esta pode ser generalizada para o caso tridimensional, sendo capaz portanto de classificar pontos em relação a poliedros arbitrários.

Para todos os efeitos, vamos admitir que um polígono simples  $P$  qualquer seja dado por uma seqüência de vértices  $\{p_1, p_2, p_3, \dots, p_{n-1}, p_n\}$  módulo  $n$ , orientada no sentido anti-horário, ou seja,  $p_{n+1} = p_1$ ,  $p_{n+2} = p_2$ , e assim por diante.

O algoritmo do *índice de rotação* se baseia no fato de que, se o *ponto teste* pertence ao interior do polígono, então a soma dos ângulos entre vértices consecutivos do polígono com  $p$  resulta em  $360^\circ$ . Por outro lado, se o *ponto teste* pertence ao exterior do polígono, então a soma resultante é igual à zero. A equação (2) sintetiza essa discussão que pode ser visualizada com o apoio da Figura 1. Nesta figura, o uso do quadrado é apenas para facilitar a compreensão do princípio subjacente à técnica. O ângulo no *ponto teste*  $p$  é considerado positivo se  $Ccw(p, p_i, p_{i+1}) \geq 0$ , e negativo, caso contrário.

$$\sum_{i=1}^n \hat{p_i p p_{i+1}} = \begin{cases} 360^\circ & \rightarrow p \in P \\ 0^\circ & \rightarrow p \notin P \end{cases} \quad (2)$$

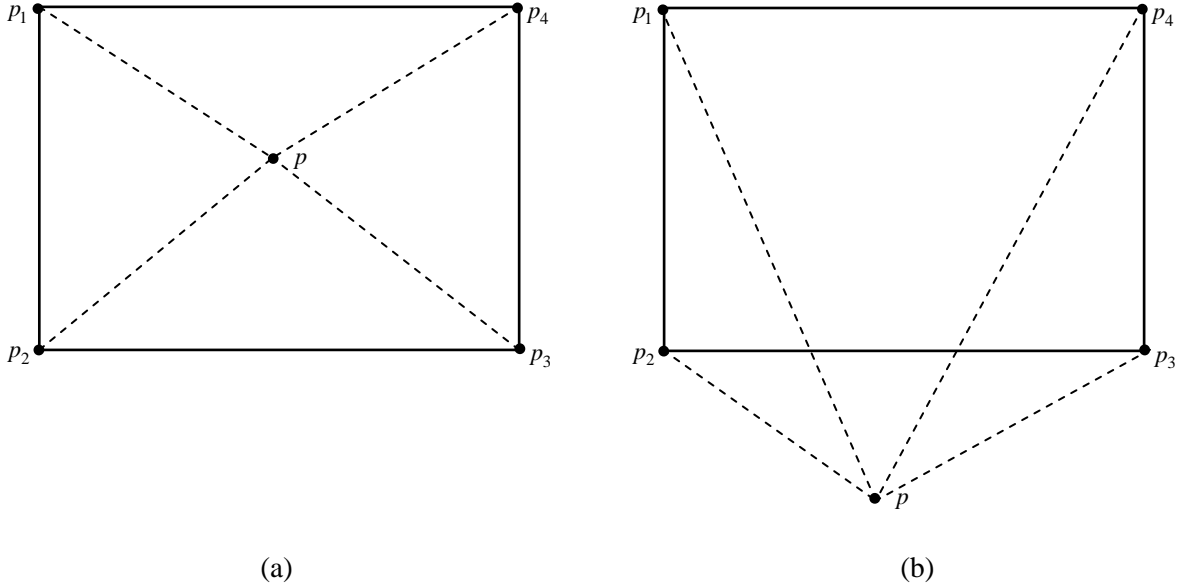


Figura 1: Princípio associado ao algoritmo do *índice de rotação*. (a) *ponto teste* no interior de  $P$ . Observem que  $Ccw(p, p_i, p_{i+1}) \geq 0, \forall i$ , logo a soma dos ângulos é  $360^\circ$ ; (b) *ponto teste* no exterior de  $P$ . Observem que  $Ccw(p, p_2, p_3) < 0$ , enquanto  $Ccw(p, p_1, p_2)$ ,  $Ccw(p, p_3, p_4)$  e  $Ccw(p, p_4, p_1)$  são positivos, logo, a soma dos ângulos se anulam.

O algoritmo baseado no *leque de triângulos* pressupõe um pré-processamento no polígono original de modo que este passe a ser descrito por um conjunto de triângulos com origem num vértice em comum, por exemplo, o vértice inicial. Desta concepção resulta o nome da técnica, já que o polígono poderia se assemelhar a uma dobradura em papel semelhante a um leque, ver Figura 2.

Com as subdivisões concluídas, identificam-se todos os triângulos que se sobrepõem ao ponto teste  $p$ . Caso o número de triângulos que o contêm seja ímpar, então  $p \in P$ , caso contrário, localiza-se no exterior. A figura 3 ilustra o princípio desta técnica.

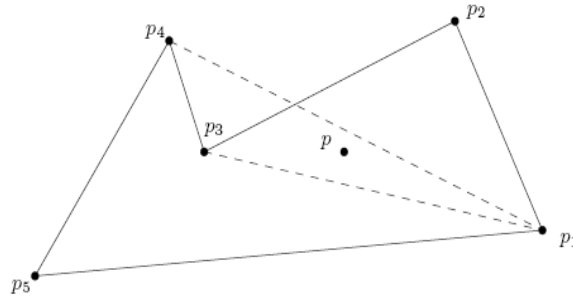


Figura 2: Observem que os triângulos formados pelas triplas de vértices  $(p_1, p_2, p_3)$ ,  $(p_1, p_3, p_4)$  e  $(p_1, p_4, p_5)$  contêm  $p$  e este número é igual a 3 (ímpar), logo,  $p \in P$ .

A seguir, vamos descrever o algoritmo de intersecção *ímpar-par*. Conforme mencionado anteriormente, este princípio pode ser generalizado para o espaço euclidiano de dimensão 3. A idéia subjacente ao método pode ser usada ainda para o preenchimento de regiões poligonais arbitrárias.

A concepção do algoritmo é simples e consiste em estender a partir do *ponto teste*  $p$  uma semi-reta  $L$  em qualquer direção. Por simplicidade, vamos assumir que  $L$  seja paralela ao eixo- $x$ . Em seguida, computamos o número de intersecções de  $L$  com os segmentos que constituem a fronteira de  $P$ . Se o número de intersecções for ímpar, então  $p \in P$ , caso contrário o ponto  $p \notin P$ .

Observem que a idéia descrita acima é muito intuitiva. Vamos imaginar que  $p \notin P$ , então, ao “caminharmos” pela semi-reta  $L$  a partir de  $p$  em direção ao infinito e encontrarmos a primeira intersecção com a fronteira de  $P$ , significa que estamos adentrando o polígono; na segunda intersecção, significa que estamos saindo do polígono e assim sucessivamente. Portanto, se o se localiza no exterior, entramos e saímos do polígono um número par de vezes. A Figura 3 oferece uma representação diagramática deste princípio.

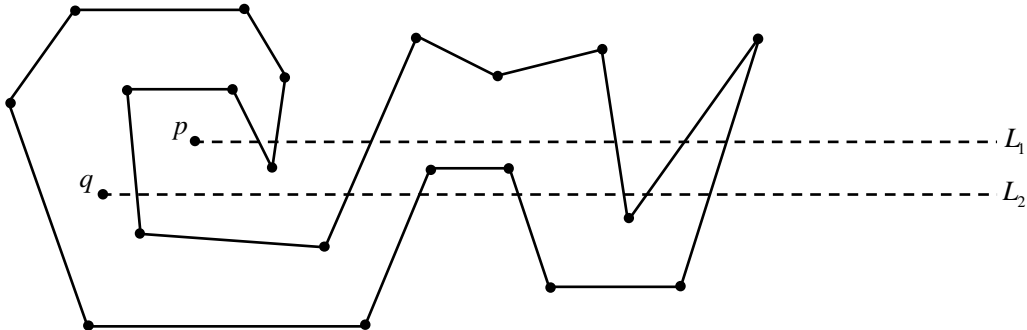


Figura 3: Polígono simples arbitrário, onde  $p \notin P$  e  $q \in P$ . O número de intersecções da semi-reta  $L_1$  com a fronteira de  $P$  é par, enquanto o da semi-reta  $L_2$  é ímpar.

Devemos considerar que podem ocorrer algumas intersecções singulares da semi-reta  $L$  com a fronteira de  $P$ , mais especificamente no caso em que  $L$  coincide com um dos extremos do segmento. Neste caso, o ponto de intersecção pertence a dois segmentos e as situações existentes, bem como o tratamento dado a cada uma delas, estão ilustradas na Figura 4. Nesta figura podemos observar que os segmentos horizontais podem ser ignorados, assim como, intersecções com os vértices de mínimo e máximo local.

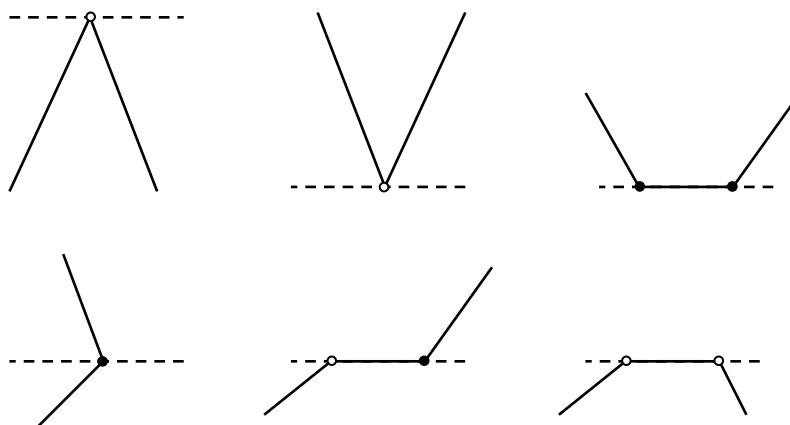


Figura 4: Singularidades geométricas que podem ocorrer no algoritmo de *intersecção ímpar-par* em relação ao cálculo das intersecções. Os pontos sólidos (círculos) indicam que a intersecção deve ser computada uma vez, enquanto as demais intersecções não são computadas.

Para finalizar, realçamos que as três abordagens algorítmicas aqui apresentadas foram devidamente implementadas usando-se o ambiente de programação *DevC++*, enquanto a parte gráfica relativa a interface, edição de polígonos simples arbitrários e visualização dos resultados foi feita com base na biblioteca gráfica *allegro*. O sistema permite ainda gerar de forma completamente automática polígonos estrelados (*star-shaped*). Esta funcionalidade está baseada numa ordenação polar de um conjunto pontos gerados randomicamente no plano.

Testes envolvendo polígonos arbitrários entre 3 e 100 lados e conjunto de *pontos testes* da ordem de  $10^2$  a  $10^6$ , tem mostrado a robustez e a consistência das implementações realizadas.

Em relação ao problema tridimensional, já realizamos os estudos teóricos para classificação de pontos em poliedros estritamente convexos. Quanto à classificação em relação a poliedros arbitrários, o princípio algorítmico de *intersecção ímpar-par* aqui descrito para resolver o problema em  $\mathbb{R}^2$ , pode ser generalizado para o respectivo problema em  $\mathbb{R}^3$ .

Uma parte significativa deste trabalho foi elaborada no âmbito de um projeto de iniciação científica que decorreu no ano de 2005 enquanto o aluno era bolsista **PAE**.

## Referências Bibliográficas

- [1] SLOAN, S. W. A Point-in-polygon program. **Advances in Engineering Software**, v. 7, n. 1. Londres: Oxford University Press. 1985, p. 45-46.
- [2] HAINES, E. Polygon and Polyhedra : point in polygon strategies. In: HECKBERT P. **Graphics Gems IV**. Boston: Academic Press, Inc. 1994. p. 24-45.
- [3] WEILER, K. Polygon and Polyhedra: an incremental angle point in polygon test. In: HECKBERT P. **Graphics Gems IV**. Boston: Academic Press, Inc. 1994. p. 16-23.
- [4] LANE, J.; MAGDSON, B.; RARICK, M. An efficient point in polyhedron algorithm. **Computer Vision, Graphics and Image Processing**. Seattle, v. 26. n.1, p. 118-125, abr. 1984.
- [5] O'ROURKE, J. **Computational Geometry in C**. Massachusetts: Cambridge University Press, 1994. 376p.

**Bolsa:** PAE