

1 CARTOGRAFIA - MODELOS VECTORIAIS

Os modelos vectoriais reproduzem o modelo conceptual da cartografia impressa, são bidimensionais e compostos por objecto estáticos e com fronteiras bem definidas (Matos, 2001). Os objectos básicos são:

- Ponto;
- Segmento de recta;
- Linha poligonal;
- Polígono.

1.1 OBJECTOS BÁSICOS

1.1.1 O PONTO

É o elemento básico de representação vectorial, definido pelas suas coordenadas cartesianas, $p0=(x0,y0)$.

1.1.2 SEGMENTO DE RECTA

É uma linha definido entre dois pontos. A Figura 3, ilustra vários segmentos de recta.

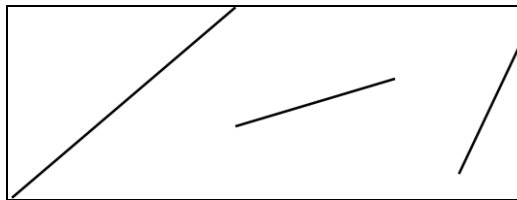


Figura 1- Segmentos de recta.

1.1.3 LINHA POLIGONAL

É gerada a partir de uma sequência de pontos ordenados. A Figura 3, ilustra duas linhas poligonais.

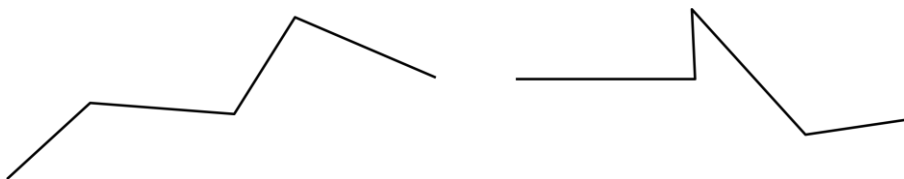


Figura 2- Duas linha poligonais.

1.1.4 POLÍGONO

É uma figura plana que é delimitada por um caminho ou circuito fechado, composto por uma sequência finita de segmentos de linha recta. Estes segmentos são chamados de suas bordas ou faces, e os pontos onde duas bordas encontram-se são os vértices do polígono ou cantos. A Figura 3, ilustra três polígonos.

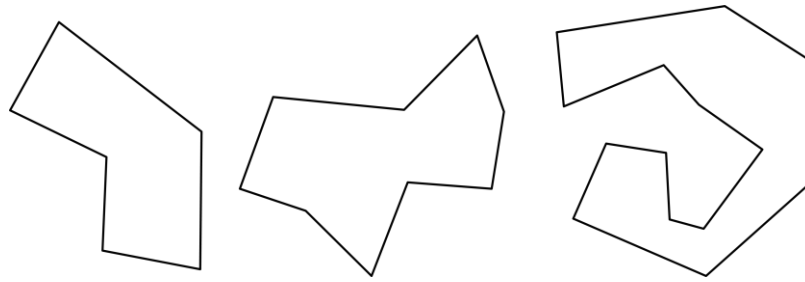


Figura 3 - Exemplo de polígonos.

1.2 OPERAÇÕES ELEMENTARES SOBRE OBJECTOS

1.2.1 COMPRIMENTO DE UMA LINHA

Uma linha é composta por vários segmentos de reta definida por um conjunto ordenado de pontos. O comprimento (C) de uma linha é definido pela soma dos comprimentos dos segmentos que a constituem, dado pela seguinte equação:

$$C = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

n – número de pontos que definem a linha.

A Figura 4, ilustra uma linha poligonal definida pelos pontos p_1 a p_5 .

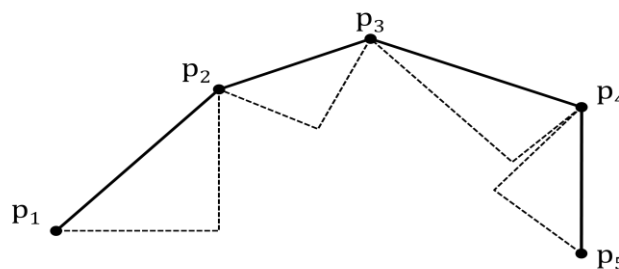


Figura 4 - Linha poligonal, ponto p_1 a p_5 .

Elabore um algoritmo/programa que permita calcular o comprimento de uma linha poligonal representada por uma lista de pontos.

1.2.1.1 ALGORITMO

Algoritmo: comprimento_linha_poligonal

Objectivo: Permite calcular o comprimento de uma linha poligonal representada por uma lista de pontos.

Variáveis:

Entradas:

n (inteiro) – Número de pontos que definem a linha.

P [n][2] (real) – Matriz com n linhas e 2 colunas.
Colunas 1 e 2 representam as coordenadas x e y dos pontos, respectivamente.

Auxiliares:

i (inteiro) – Índice para a matriz pontos/estrutura Para.

Saída:

C (real) – Comprimento da linha poligonal.

Data: 4-4-2010

Versão: 1.0

Autor: Paulo Nunes.

Início:

Escrever "Número de pontos?"

Fazer

Ler n

Enquanto n < 0

Para i

Para i ← 1 Até n Fazer

Escrever "Ponto ", i, " x y ?"

Ler P[i][1], P[i][2]

FimPara

C ← 0

Para i ← 1 Até n-1 Fazer

$C \leftarrow C + \sqrt{(P[i+1][1] - P[i][1])^2 + (P[i+1][2] - P[i][2])^2}$

FimPara

Escrever "Comprimento da linha poligonal: ", C

Fim.

1.2.2 ÁREA, PERÍMETRO E CENTRO DOS POLÍGONOS

A Figura 5, ilustra um polígono definido pelos pontos p₁, p₂, p₃, p₄, p₅, p₆ e p₁.

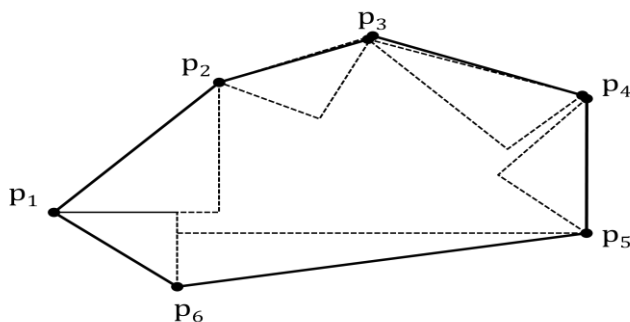


Figura 5 - Polígono, ponto p₁ a p₆.

A área (A) de um polígono é dada pela seguinte equação:

$$A = \frac{1}{2} \sum_{i=1}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

A Figura 6, ilustra a primeira iteração do cálculo da área do polígono, pontos p_1, p_2 . Na figura, o triângulo preenchido representa metade da área do quadrado que resulta da diferença entre as áreas dos dois retângulos definidos por x_1y_2 e x_2y_1 . A área do retângulo definido por x_1y_1 é comum aos dois anteriores.

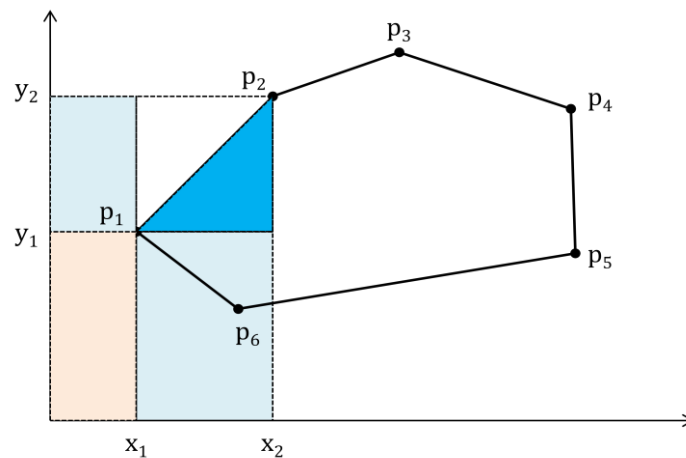


Figura 6 – Cálculo da área do polígono, primeira iteração.

1.2.2.1 ALGORITMO

Algoritmo: Area_poligono($n, P[][]$) (real)	
Objectivo: Permite calcular o comprimento de uma linha poligonal representada por uma lista de pontos.	
Parâmetros:	
Entradas:	
n (inteiro)	– Número de pontos que definem o polígono.
$P[n][2]$ (real)	– Matriz com n linhas e 2 colunas. Colunas 1 e 2 representam as coordenadas x e y dos pontos, respetivamente.
Auxiliares:	
i (inteiro)	– Índice para a matriz pontos/estrutura Para.
Saída:	
A (real)	– Área do polígono.
Data: 7-10-2010	
Versão: 1.0	
Autor: Paulo Nunes.	
Início:	
$A \leftarrow \text{Area_poligono}(n)$	
Para $i \leftarrow 1$ Até $n-1$ Fazer	
$A \leftarrow A + (P[i][1]P[i+1][2] - P[i+1][1]P[i][2])$	
FimPara	
$A \leftarrow A / 2$	
Retorna A	
Fim.	

O perímetro (P) pela seguinte:

$$P = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}$$

E o centro (C_x , C_y) pelas seguintes:

$$c_x = \frac{1}{6A} \sum_{i=1}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$c_y = \frac{1}{6A} \sum_{i=1}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

1.2.2.2 ALGORITMO

<p>Algoritmo: Centro_poligono</p> <p>Objectivo: Permite calcular o comprimento de uma linha poligonal representada por uma lista de pontos.</p> <p>Variáveis:</p> <p>Entradas:</p> <p> n (inteiro) – Número de pontos que definem o polígono.</p> <p> P [n][2] (real) – Matriz com n linhas e 2 colunas. Colunas 1 e 2 representam as coordenadas x e y dos pontos, respectivamente.</p> <p>Auxiliares:</p> <p> i (inteiro) – Índice para a matriz pontos/estrutura Para.</p> <p> A (real) – Área do polígono.</p> <p>Saida:</p> <p> C_x (real) - Coordenada x do centro.</p> <p> C_y (real) - Coordenada y do centro.</p> <p>Data: 7-10-2010</p> <p>Versão: 1.0</p> <p>Autor: Paulo Nunes.</p>	<p>Início:</p> <p> Escrever "Número de pontos?"</p> <p> Fazer</p> <p> Ler n</p> <p> Enquanto n < 0</p> <p> Para i</p> <p> Para i ← 1 Até n Fazer</p> <p> Escrever "Ponto ", i, " x y ?"</p> <p> Ler P[i][1], P[i][2]</p> <p> FimPara</p> <p> A ← Area_poligono(n, P) ' Chamada à função: Area_poligono ()</p> <p> c_x ← 0</p> <p> c_y ← 0</p> <p> Para i ← 1 Até n-1 Fazer</p> <p> c_x ← c_x + (P[i][1] + P[i + 1][1]) (P[i][1]P[i + 1][2] - P[i + 1][1]P[i][2])</p> <p> c_y ← c_y + (P[i][2] + P[i + 1][2]) (P[i][1]P[i + 1][2] - P[i + 1][1]P[i][2])</p> <p> FimPara</p> <p> A ← A / 2</p> <p> c_x ← c_x $\frac{1}{6A}$</p> <p> c_y ← c_y $\frac{1}{6A}$</p> <p> Escrever "Centro (", c_x, ", ", c_y, ")"</p> <p>Fim.</p>
--	---

1.3 OPERAÇÕES ENTRE OBJECTOS

1.3.1 DISTÂNCIA ENTRE PONTOS

Elabore um algoritmo e um programa em VBA que permita calcular a distância (d) entre dois pontos. A distância entre dois pontos, p_0 e p_1 é dada pela seguinte equação:

$$d = \sqrt{(x_1 - x_0)^2 - (y_1 - y_0)^2}$$

1.3.1.1 ALGORITMO

Algoritmo: distancia_entre_dois_pontos	
Objectivo: Permita calcular a distância Euclidiana (d) entre dois pontos com coordenadas bidimensionais.	
Variáveis:	
Entradas:	
x_0 (real)	– Coordenada x do primeiro ponto.
y_0 (real)	– Coordenada y do primeiro ponto.
x_1 (real)	– Coordenada x do segundo ponto.
y_1 (real)	– Coordenada y do segundo ponto.
Saída:	
d (real)	– Distância entre os pontos.
Data: 7-10-2010	
Versão: 1.0	
Autor: Paulo Nunes.	
Início:	
Escrever "Ponto 1: x y ?"	
Ler x_0, y_0	
Escrever "Ponto 2: x y ?"	
Ler x_1, y_1	
$d = \sqrt{(x_1 - x_0)^2 - (y_1 - y_0)^2}$	
Escrever "Distância: ", d	
Fim.	

1.3.2 DISTÂNCIA ENTRE LINHAS: DISTÂNCIA MÍNIMA, ...

1.3.3 DISTÂNCIA ENTRE POLÍGONOS

Bibliografia.

João Luís de Matos, Fundamentos de Informação Geográfica, 2ª Edição, Lidel, 2001 [Capítulo 4]

<http://www.isa.utl.pt/dm/sigdr/sigdr/SumariosTeoricas.html>