



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Computação Aplicada

**CESAR ARTURO SANCHEZ PENA
MATHEUS PEREIRA BENTO DA COSTA**

**STRIP TREE: REPRESENTAÇÃO HIERÁRQUICA PARA
APROXIMAÇÃO DE CURVAS**

São José dos Campos

2020

Lista de ilustrações

Figura 1 – O algoritmo básico de Douglas-Peucker. Fonte (WU; MARQUEZ, 2003) . . .	4
Figura 2 – Exemplo de generalização cartográfica para uma determinada linha de curva. Fonte (JARAMILLO, 2006)	5
Figura 3 – Exemplo de uma curva de dois dimensões que utiliza uma estrutura hierár- quica. Fonte (BALLARD, 1981)	6
Figura 4 – Definição do Strip (retângulo ou banda). Fonte (BALLARD, 1981)	6
Figura 5 – Exemplo do Strip-Tree formado a partir de um segmento simples. Fonte (BALLARD, 1981)	7
Figura 6 – Estrutura de um Strip-Tree.	8
Figura 7 – Primeira e segunda aproximação da curva. Fonte (GÜNTHER; WONG, 1990)	9
Figura 8 – Estrutura do árvore Arc-Tree.	10
Figura 9 – Retângulos para aproximações de um círculo. Fonte (FIGUEIREDO; STOLFI; VELHO, 2002)	10
Figura 10 – Funções utilizadas para os testes.	11
Figura 11 – Ordem dos dados de cada retângulo, para os introduzir nas árvores binários. Fonte (BALLARD, 1981).	12
Figura 12 – Elementos do Strip-Tree usados para obter o comprimento aproximado. . . .	12
Figura 13 – Digitalização da função 1.	13
Figura 14 – Digitalização da função 2.	14
Figura 15 – Digitalização da função 3.	15
Figura 16 – Digitalização da função 4.	16
Figura 17 – Mudança da taxa de Erro para cada uma das funções usadas.	17
Figura 18 – Função que representa um litoral.	18
Figura 19 – Aplicação de Generalização cartográfica.	18

Sumário

1	INTRODUÇÃO	3
2	METODOLOGIA	4
2.1	Algoritmo de Douglas–Peucker	4
2.2	Generalização Cartográfica	5
2.3	Strip-Tree	6
2.4	Ponto de divisão	9
2.5	Arc-Tree	9
3	DADOS	11
4	RESULTADOS	12
4.1	Strip-Tree	12
4.2	Generalização Cartografica	18
5	CONCLUSÃO	19
	REFERÊNCIAS	20

1 Introdução

Neste documento apresenta o trabalho de grupo para a matéria de Computação Aplicada, fornecido pelo Professor Gilberto Ribeiro, do programa de Computação Aplicada - INPE.

A extração de características de uma determinada fonte geográfica e a forma podem representada de um modo simples, mas mantendo certas informações relevantes, conhecida como generalização cartográfica. A utilização de curvas para representar diferentes estruturas é uma parte importante em diferentes áreas de investigação, por exemplo, a representação de rodovias, rios, lagos, fronteiras de países ou até mesmo cidades e entre outras representações. Por conseguinte, a representação destas curvas é de extrema importância. O algoritmo de Douglas–Peucker apresenta uma metodologia para resolver este problema.

Agora, se dispusermos de um grande volume de dados (embora não seja necessário), precisará de uma metodologia para realizar operações de forma mais eficiente, com o objetivo de representar as curvas de interesse, com tal finalidade, é utilizada uma representação hierárquica utilizando árvores binárias, em que um determinado segmento da curva é definido como **Strip**, e a árvore que contém esses segmentos (dados do segmento) é denominada como **Strip Tree**, estas árvores foram introduzidas pelo Ballard e inspirada pelo algoritmo de Douglas–Peucker (BALLARD, 1981).

A generalização cartográfica em conjunto com a Strip-tree tem uma grande quantidade de aplicações (comprimento, intersecção, conteúdo, área e outras operações) no entanto para este trabalho apenas utilizamos pontos concatenados fornecidos por diferentes funções algébricas, que formam uma determinada curva, para obtivermos o comprimento aproximado da mesma.

2 Metodologia

2.1 Algoritmo de Douglas–Peucker

Uma tarefa importante na cartografia é o mapeamento e a extração de características de dados detalhados e também a sua representação em um mapa simples e legível, por esse motivo surge a importância de desenvolver algoritmos eficientes para a execução destas tarefas. O algoritmo de simplificação de linhas da Douglas-Peucker é reconhecido como oferecendo as melhores representações perceptivas das linhas originais.

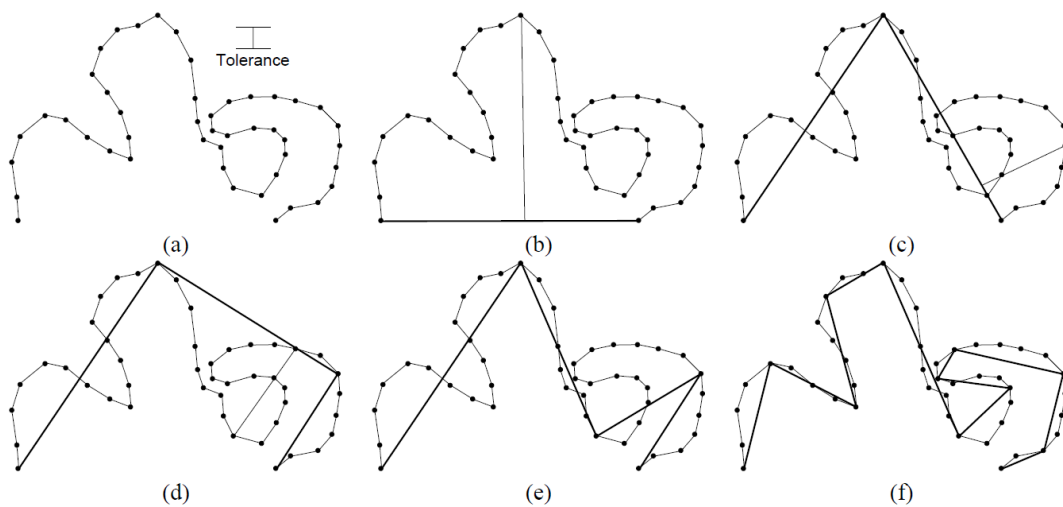


Figura 1 – O algoritmo básico de Douglas-Peucker. Fonte (WU; MARQUEZ, 2003)

Douglas e Peucker (DOUGLAS; PEUCKER, 1973) desenvolveram seu algoritmo, onde dada uma cadeia poligonal $V = v_0, v_1, v_2, \dots, v_n$. O método de simplificação de linha recursiva procede da seguinte maneira:

Inicialmente, aproxima V pelo segmento de linha, $\overline{v_0 v_n}$, após isto, determina o vértice mais distante (v_f) a partir da linha $\overline{v_0 v_n}$, se a distância entre os vértices for superior a uma determinada tolerância ($\epsilon \geq 0$), o segmento é aceito como uma boa aproximação de V . Caso contrário, quebre V em v_f e aproxime recursivamente a sub-cadeia. Assim, o tempo de execução desse algoritmo satisfará uma recorrência, com o melhor comportamento de $O(n \log n)$, (HERSHBERGER; SNOEYINK, 1992), (HERSHBERGER; SNOEYINK, 1994) e (WU; MARQUEZ, 2003). (VISVALINGAM; WHYATT, 1990) apresenta a problemática presente na cartografia de generalização das linhas e avaliam o algoritmo de Douglas-Peucker.

2.2 Generalização Cartográfica

A generalização cartográfica consiste em reduzir a escala de um mapa de base e adaptar todos os seus elementos a uma nova escala. O objetivo da generalização cartográfica é produzir um mapa claramente legível e interpretá-lo a partir de informação de base considerada excessivamente abundante ou densa, o conteúdo do mapa deve ser reduzido a um nível necessário, ou seja, enfatizando as características mais importantes e suprimindo o que é de interesse. As principais operações no processo de generalização do mapeamento manual são as seguintes (MCMASTER; SHEA, 1992):

- Simplificação: remover pontos de um polígono para desenvolver uma geometria mais simples.
- Alisamento: substituir de formas afiadas e complexas por formas mais suaves.
- Quebra: substituir de uma área por uma combinação de pontos e linhas.
- Agregação: substituir de um conjunto de objetos por um número menor de objetos.
- Fusão: substituir de vários objetos por um único objeto.
- Mistura: substituir de um conjunto de linhas por um número menor de linhas.
- Refinamento: substituir de um padrão de objetos para uma forma geral do padrão.
- Exagero: aumento relativo de um objeto para preservar suas características.
- Ampliação: alterar o tamanho e a forma dos símbolos.
- Movimento: movimentar os objetos a partir das suas posições originais para preservar a sua visibilidade.

Neste trabalho apenas é considerada uma generalização da cartografia aplicada às linhas ou seja simplificação e alisamento, como se mostra na figura a continuação.

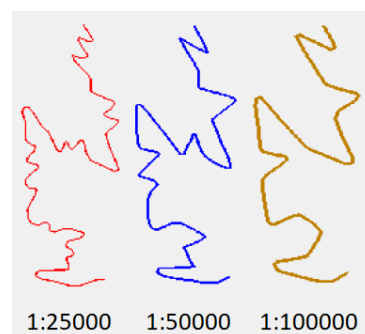


Figura 2 – Exemplo de generalização cartográfica para uma determinada linha de curva. Fonte (JARAMILLO, 2006)

2.3 Strip-Tree

O principal objetivo do Strip-Tree é representar cada parte da curva por um retângulo delimitação (chamado Strip) que contém esse segmento. Desenvolvendo-o de forma hierárquica, começando com toda a curva, depois para segmentos maiores e assim por diante, diminuindo com o qual obtemos uma árvore de retângulos (Strip ou bandas) onde cada retângulo contém uma determinada parte da curva.

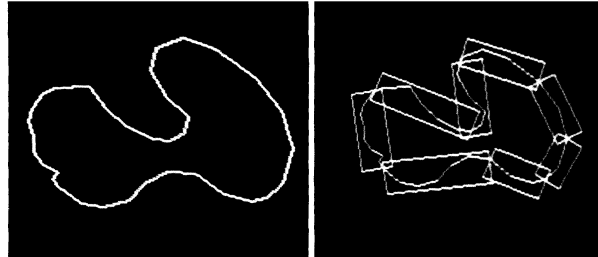


Figura 3 – Exemplo de uma curva de dois dimensões que utiliza uma estrutura hierárquica. Fonte (BALLARD, 1981)

Para a representação de curvas utilizaremos as árvores binárias, sendo os níveis inferiores correspondem a resoluções superiores. A partir da figura 3 pode-se ver que cada retângulo contém um determinado segmento da curva, esta é a ideia principal do Strip-Tree com a qual se pode obter, por exemplo, uma aproximação do comprimento da curva.

Para definir os retângulos (ou bandas) têm a seguinte notação (x_b, x_e, w_r, w_l) , onde x_b e y_b é o ponto inicial do retângulo, x_e e y_e o ponto final, w_r e w_l são as distâncias a partir dos bordas do retângulo direito e esquerdo da linha na direção (x_b, y_b) a (x_e, y_e) , estas definições são apresentadas na figura 4, Juntamente com isto, o w é definido como $w = w_r + w_l$. Com estes parâmetros pode facilmente definir o retângulo que contém o segmento da curva, por exemplo, a área da banda é definida por $w|x_b - x_e|$, que em termos rápidos representa a base/comprimento multiplicada pela altura.

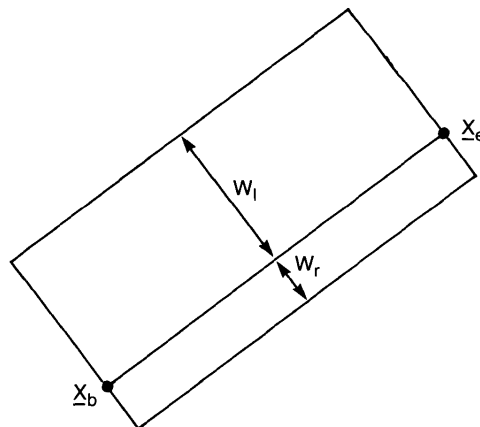


Figura 4 – Definição do Strip (retângulo ou banda). Fonte (BALLARD, 1981)

Para este caso, uma curva é definida como um conjunto de pontos discretos que são colineares e consecutivos, estes pontos são considerados ligados, sim, a curva é representada por uma resolução w^* , as aproximações desejadas devem ter uma resolução inferior a w^* , isto é $(w_r + w_l) < w^*$, a largura do retângulo deve ser inferior à resolução da curva.

Tomando um exemplo simples de um determinado segmento definido por 5 pontos, o Strip-Tree é mostrado na figura 5, em que o ponto de partida é determinado pelo par ordenado (3,7) e o ponto final é (20,7) e a partir disto é formado o Strip-Tree.

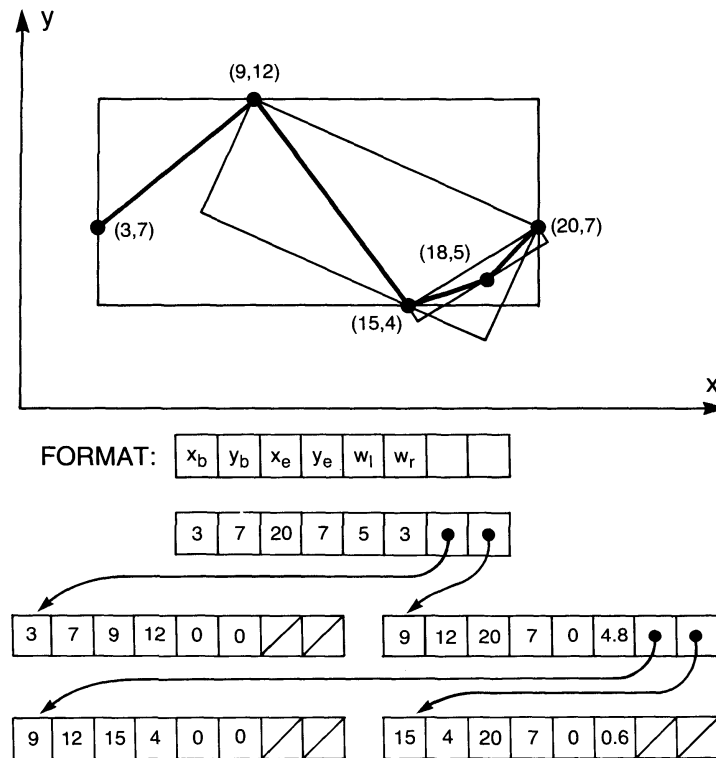


Figura 5 – Exemplo do Strip-Tree formado a partir de um segmento simples. Fonte (BALLARD, 1981)

Primeiro é formado o retângulo que contém todo o segmento, desde o ponto de partida até o final, também é computada a largura direita e esquerda, estes valores serão inseridos como a raiz da árvore binária. Entre o ponto inicial e o ponto final, a largura mínima do retângulo é igual a 3, pelo que consideramos a resolução da curva como $w^* = 3$.

O segmento deve ser dividido em duas partes, desde o ponto inicial (3,7) até um ponto em que o retângulo contenedor tenha uma resolução inferior a w^* , isto é verdade para o segmento de (3,7) a (9,12) que tem $w_r = 0$ e $w_l = 0$, enquanto o segundo segmento é definido por um retângulo desde (9,12) a (20,7) com $w_r = 0$ e $w_l = 4,8$. O segmento seguinte é de (9,12) a (15,4) novamente com resolução 0, ou seja, uma linha recta, e o segundo de (15,4) a (20,7) com $w_r = 0$ e $w_l = 0,6$.

Tendo em conta que o último retângulo tem três pontos em que já não é necessário recalcular outro retângulo desde (18,5) a (20,7) porque a resolução é de $w_r + w_l = 0,6$, que é inferior à resolução do segmento w^* , completando assim o Strip-Tree que representa o segmento indicado na figura 5.

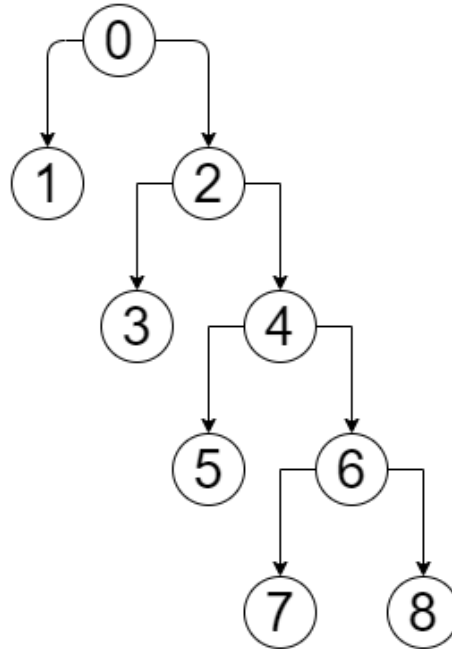


Figura 6 – Estrutura de um Strip-Tree.

Para sintetizar, são definidos um ponto inicial e um ponto final da curva, onde a raiz da árvore representa a curva completa, logo são considerados dois sub-segmentos, o primeiro deve ter uma resolução inferior à resolução da curva (w^*), enquanto que o segundo vai do ponto final do segmento um até ao ponto final da curva, depois o segundo sub-segmento vai ser novamente dividido outros dois sub-segmentos, repetindo o passo anterior e assim sucessivamente até que a resolução do último sub-segmento seja inferior a w^* , como os sub-segmentos são definidos obtém-se o retângulo contendor, o filho esquerdo é o retângulo com resolução inferior a w^* enquanto o filho direito será o retângulo que representa o segundo sub-segmento (o resto da curva), a figura 6 mostra um exemplo do ordenamento do árvore.

A partir da figura 6 pode ver-se que os filhos esquerdos (1, 3, 5 e 7) são um retângulo com resoluções inferiores a w^* , pelo que, se a diagonal de cada retângulo for calculada e depois somadas, obtém-se uma aproximação do comprimento da curva. Como observação para obter um melhor desempenho, estes retângulos não estão necessariamente alinhados com os eixos x e y, A figura 5 já mostra retângulos inclinados. Finalmente, o algoritmo para a criação do retângulo é normalmente calculado num período de $O(\log n)$, onde n é o numero de pontos que descreve a curva.

2.4 Ponto de divisão

Como obter o ponto de divisão de uma curva com o qual se divide em dois segmentos, Ballard (BALLARD, 1981) mostrou esta segmentação em relação ao parâmetro w^* , onde o retângulo que representa o primeiro segmento deve ter uma resolução inferior e assim sucessivamente ao longo da curva. Outra consideração para o ponto de divisão, pode ser o ponto médio da curva, embora o ponto médio possa ser difícil de calcular de acordo com a curva, esta solução foi adoptada por Gunther e Wong (GÜNTHER; WONG, 1990) no Arc-Tree.

2.5 Arc-Tree

Semelhança do caso de Arc-Tree apresentado anteriormente, é utilizada a metodologia adoptada por Gunther e Wong (GÜNTHER; WONG, 1990), em que a principal diferença se encontra no ponto de divisão da curva, o esquema de Arc-Tree, é dividir a curva recursivamente a partir do seu ponto médio, pelo que as duas sub-curvas geradas têm o mesmo comprimento, do qual se obtêm os retângulos de contenção de cada segmento de curva, o ponto médio é sempre definido em termos do comprimento do arco, daí o seu nome de Arc-Tree, a figura 2 mostra uma representação desta divisão.

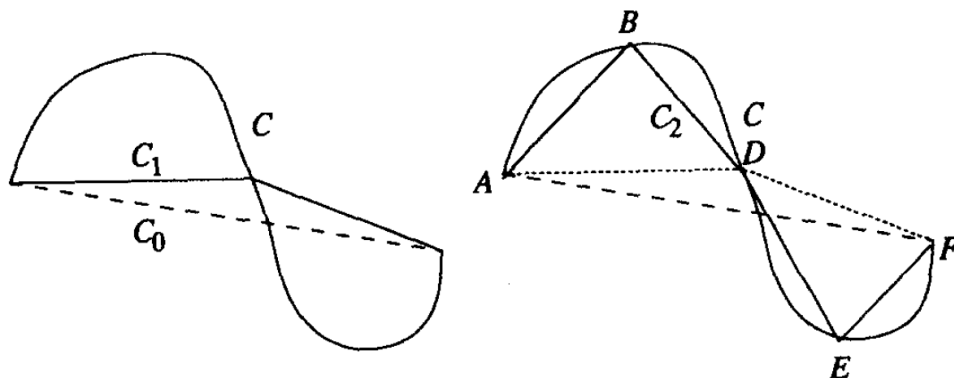


Figura 7 – Primeira e segunda aproximação da curva. Fonte (GÜNTHER; WONG, 1990)

A fim de aproximar uma determinada curva, a raiz da árvore seria simplesmente um retângulo contendo toda a curva, com um comprimento igual à recta desde o ponto inicial até ao ponto final, esta seria a primeira aproximação.

Em seguida, começa com o cálculo do ponto médio da curva (D), gerando assim duas sub-curvas (ou comprimento do arco, \widehat{AD} e \widehat{DF}), esta seria a primeira aproximação da curva original, que será representada pela linhas \overline{AD} e \overline{DF} , o que finalmente ainda gera uma aproximação muito deficiente, mas um pouco melhor do que a linha original \overline{AF} .

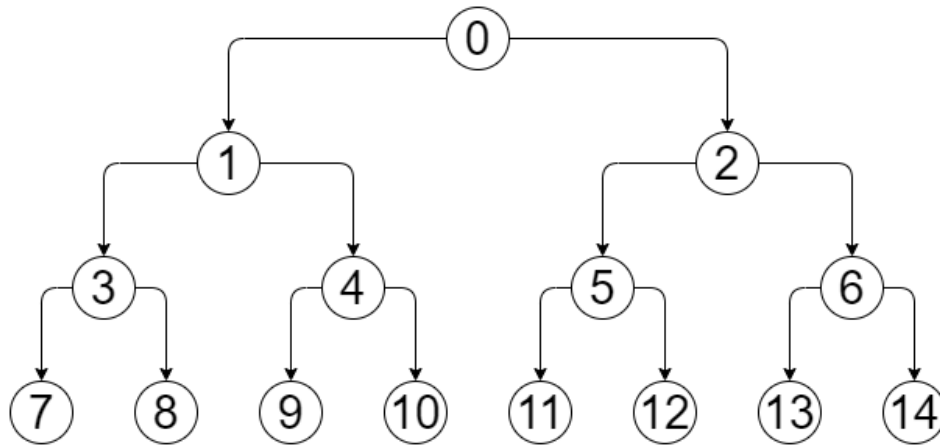


Figura 8 – Estrutura do árvore Arc-Tree.

Uma segunda aproximação pode agora ser gerada através de obter os pontos médios dos arcos \widehat{AD} com ponto meio B e \widehat{DF} com ponto meio E, obtendo então 4 arcos novos (\widehat{AB} , \widehat{BD} , \widehat{DE} , \widehat{EF}), tendo então as linhas proporcionam-se uma aproximação mais adequada à curva original. Como mostra a figura 8, a estrutura da Arc-Tree mostra uma árvore binária perfeito, uma vez que até se parar o análise se obtém metade de um arco, que é representado por dois filhos. Para este caso, o nível da árvore é um nível de resolução diferente da curva, ou seja, representações melhores ou piores da curva original.

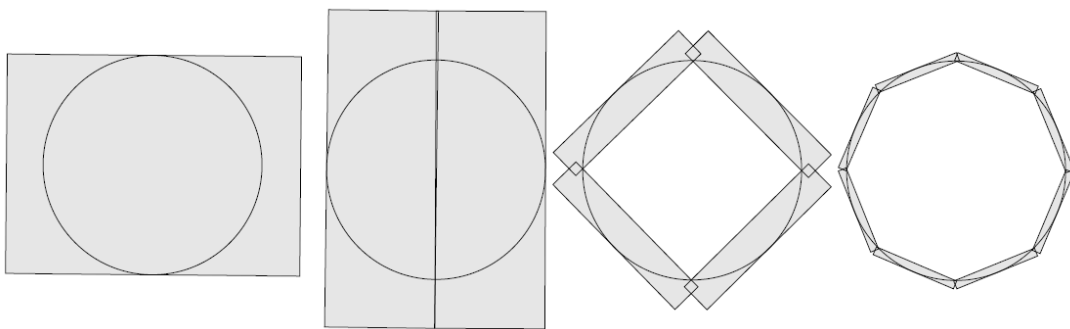


Figura 9 – Retângulos para aproximações de um círculo. Fonte (FIGUEIREDO; STOLFI; VELHO, 2002)

A figura 9 representa o processo de Arc-Tree para obter uma aproximação de um círculo definido como $x = \cos(t)$, $y = \sin(t)$ para $t \in [0, 2\pi]$, mostrando os níveis 0, 1, 2 e 3. Por conseguinte, o nível 3 fornece a melhor aproximação para o círculo. Como pode-se ver, existirá sempre um número par de retângulos contentores.

3 Dados

A fim de testar a metodologia apresentada junto com a implementação, foram criadas algumas funções, a tabela 1 mostra os detalhes. Estas funções serão utiliza-se tanto para Strip-Tree como para Arc-Tree para comparar os resultados.

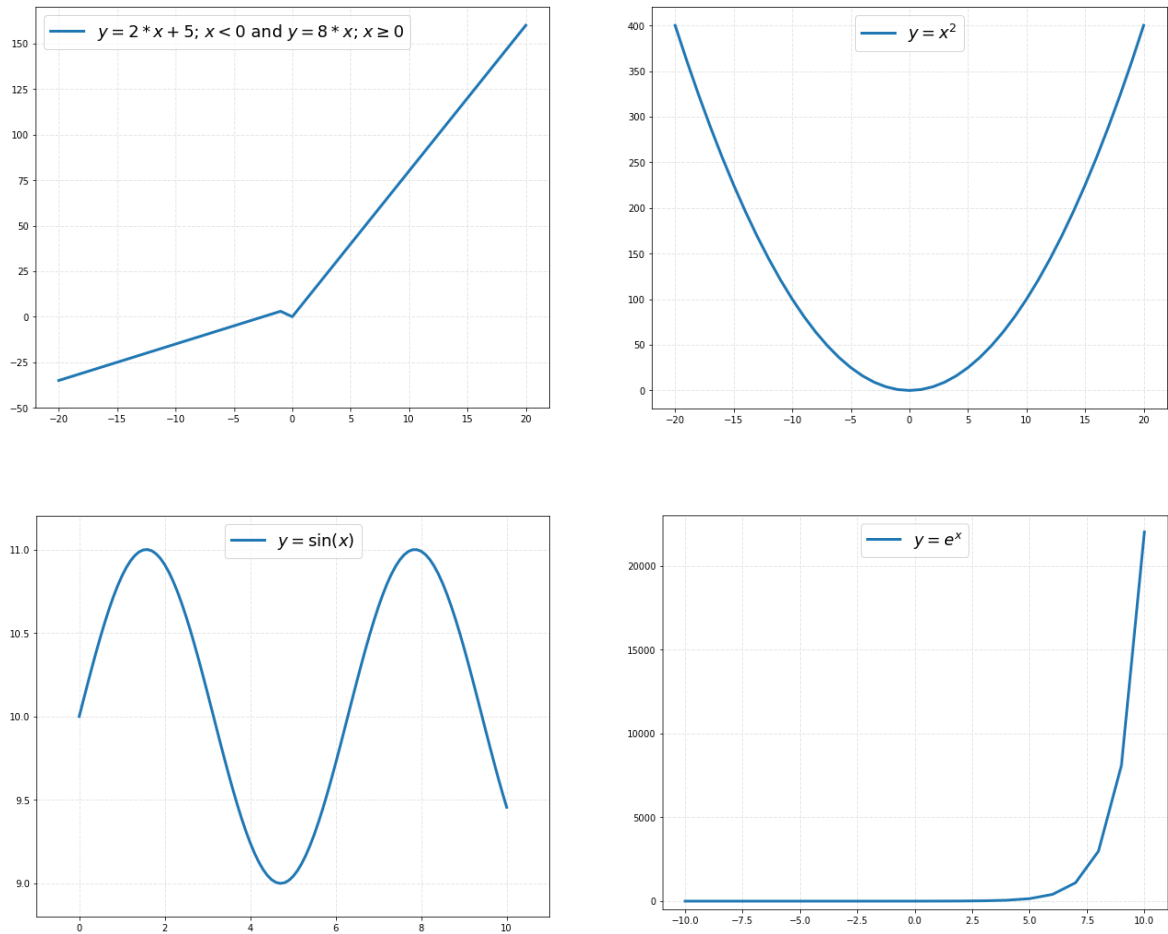


Figura 10 – Funções utilizadas para os testes.

Estas funções serão consideradas como as curvas originais, ou seja, representam alguma característica da realidade, tal como um rio ou uma estrada. A partir daí, as representações serão obtidas variando a resolução até se obter a melhor aproximação.

Tabela 1 – Informação necessária para criar as funções utilizadas.

Nome	Função	x mínimo	x máximo	comprimento
Função 1	$y = 2x + 5 \wedge y = 8x$	-20	20	41
Função 2	$y = x^2$	-20	20	41
Função 3	$y = \sin(x)$	0	10	101
Função 4	$y = e^x$	-10	10	21

4 Resultados

Para cada uma das funções da figura 10, as aproximações foram calculadas usando o código de digitalização para diferentes resoluções, no entanto, aqui apenas 4 resoluções por função serão apresentadas. Para cada resolução, o comprimento da curva é calculado.

Agora do capítulo anterior, é possível verificar a forte relação entre as secções, uma vez que o objectivo inicial é a generalização cartográfica (secção 2.2) para obter esta apresentação uma alternativa é usar o algoritmo de Douglas-Peucker (secção 2.1) para finalmente a informação obtida será armazenada sob a forma de Strip-Tree (secção 2.3).

x_s	y_s	x_e	y_e	w_l	w_r
-----	-----	-----	-----	-----	-----

Figura 11 – Ordem dos dados de cada retângulo, para os introduzir nas árvores binários. Fonte (BALLARD, 1981).

4.1 Strip-Tree

Com base no que foi apresentado na seção 2.3, foi criado um algoritmo digitalizador para obter os pontos que serão inseridos na árvore binária, gerando o Strip-Tree, mantendo a ordem estabelecida na figura 6. Portanto, adicionando todos os filhos esquerdos (somatório) e o último filho direito, é obtida uma aproximação da curva, para a resolução usada inicialmente.

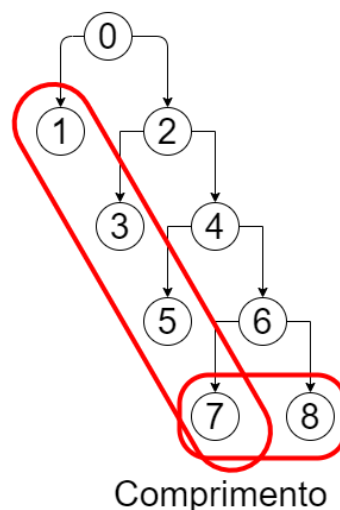


Figura 12 – Elementos do Strip-Tree usados para obter o comprimento aproximado.

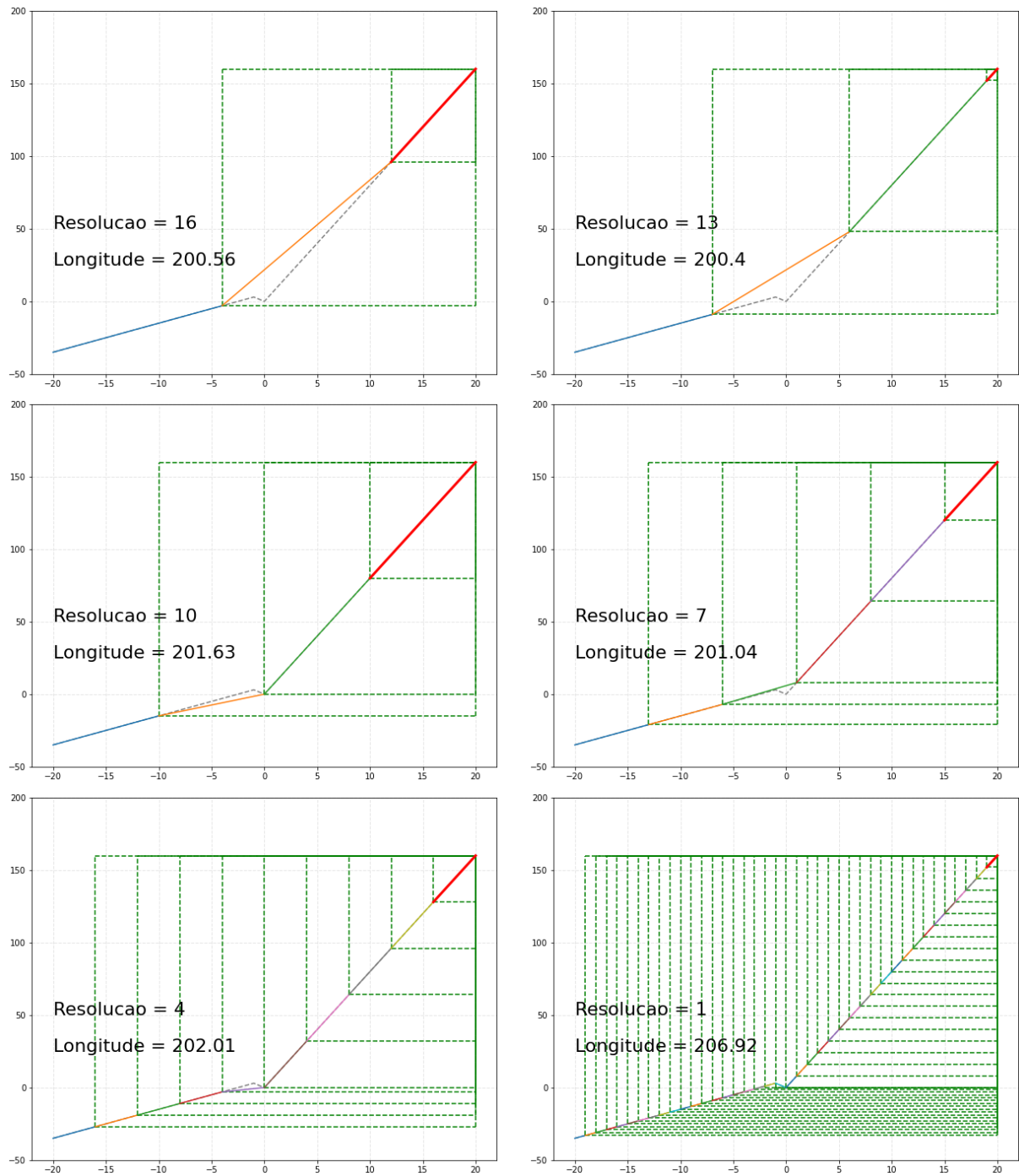


Figura 13 – Digitalização da função 1.

A figura 12 mostra o processo de obtenção das linhas aproximadas e as métricas dos retângulos. O comprimento para uma resolução igual a 1 unidade (adimensional), serão consideradas como o comprimento de referência, este comprimento também foi obtido separadamente para verificar seu valor correto.

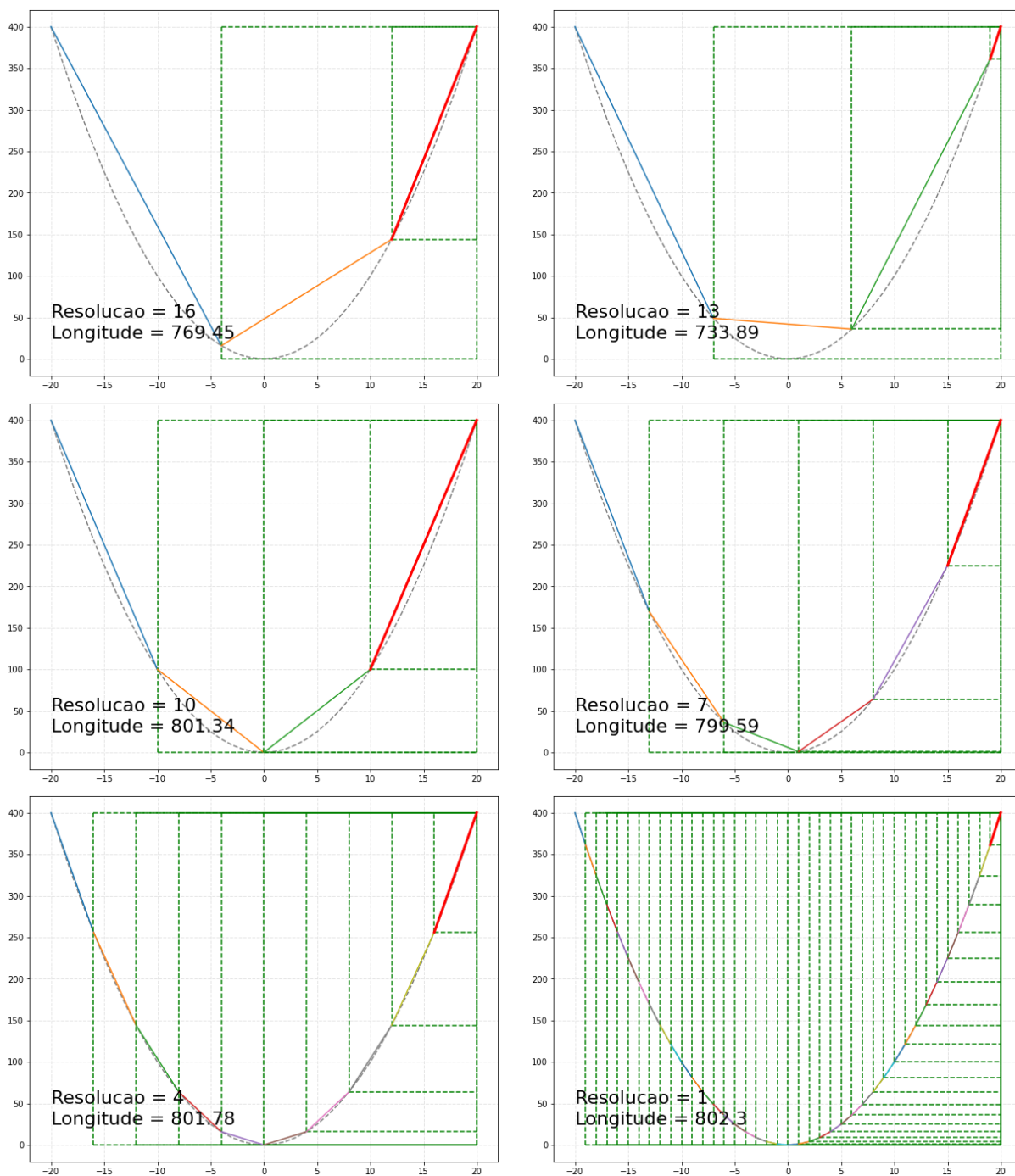


Figura 14 – Digitalização da função 2.

A função exponencial mostra seu comportamento simétrico em relação ao eixo y (função par), a partir daqui pode-se verificar como, mesmo na resolução de 10 unidades, bons resultados são obtidos para o comprimento da curva.

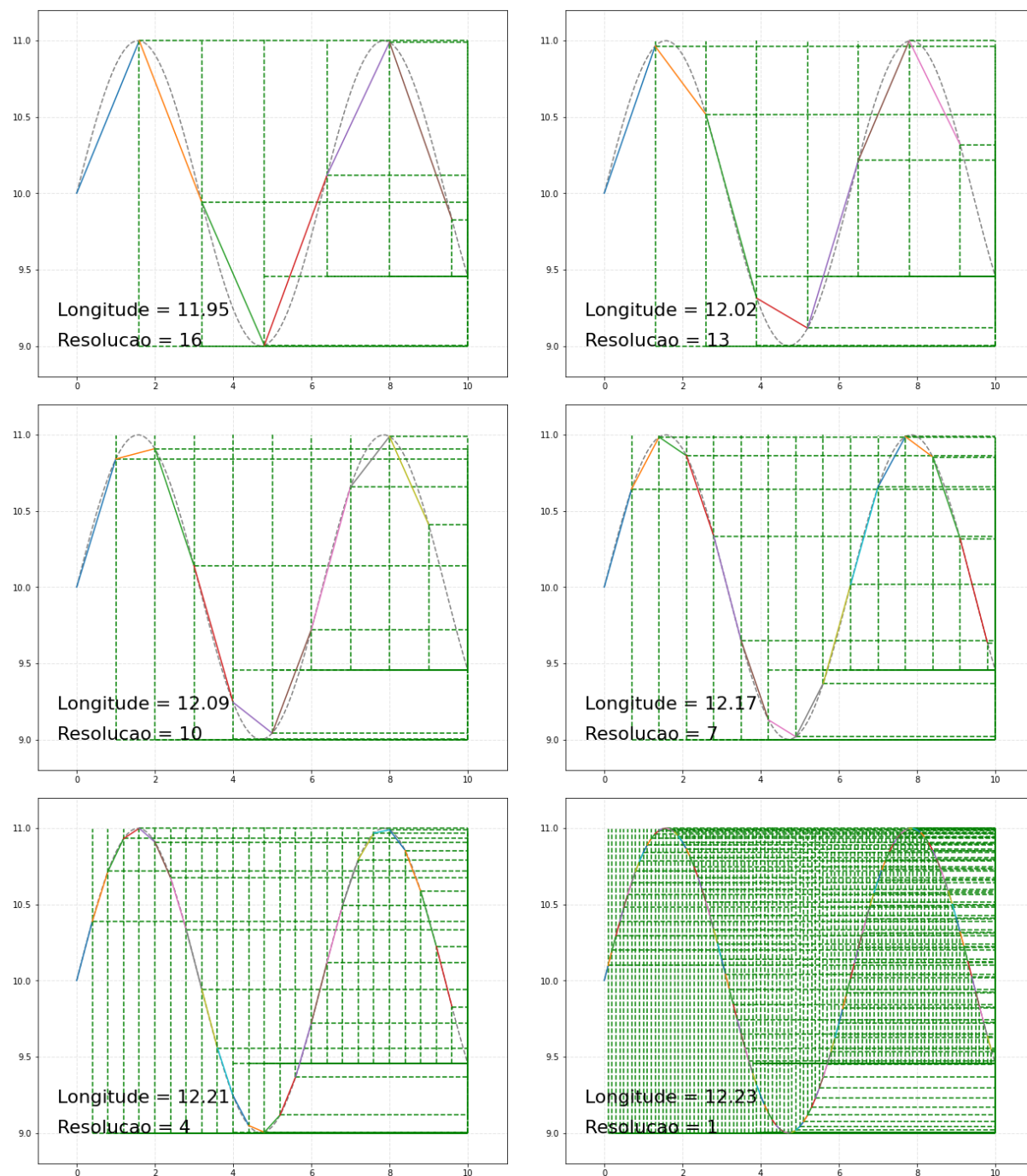


Figura 15 – Digitalização da função 3.

A função senoidal é a que apresenta a maior quantidade de erros à medida que a resolução piora, devido ao formato que possui, uma vez que quanto maiores e mais suaves as curvas, será necessária uma aproximação muito mais fina (alta resolução). para obter resultados favoráveis.

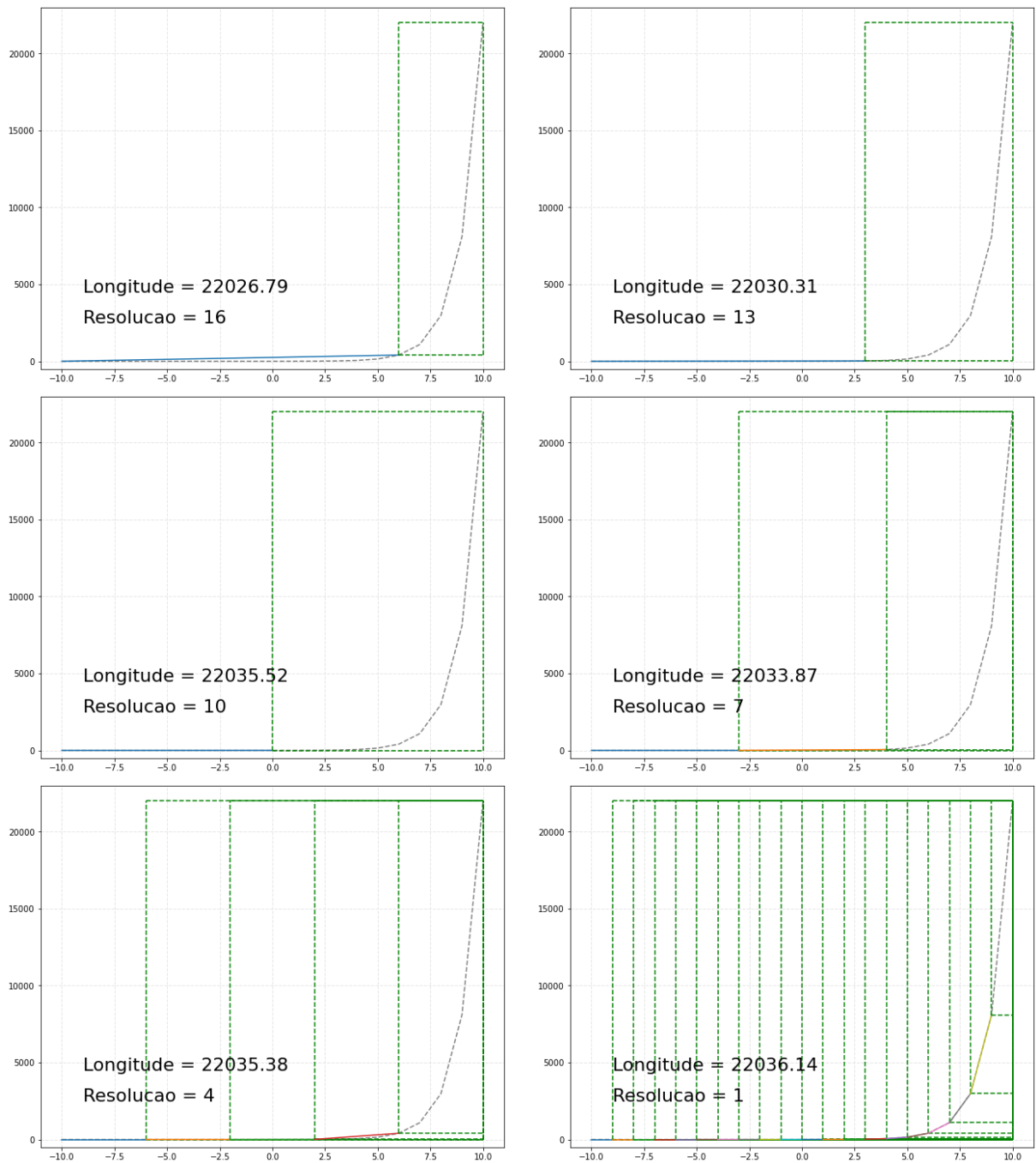


Figura 16 – Digitalização da função 4.

A Figura 17 mostra a taxa de erro em relação à resolução para cada uma das funções utilizadas. Observe que a diferença entre o erro máximo e o mínimo (ou seja, a faixa de valores de erro) é menor para funções com um número baixo de curvas (função linear e exponencial), enquanto que para funções mais curvilíneas, como a quadrática e a senoidal, é maior. No caso da função sinusoidal, a razão entre a taxa de erro e a resolução é praticamente linear, ou seja, para qualquer caso em que a resolução piore, o erro obtido será maior; No entanto, embora a variação dos erros não seja muito alta, eles aumentam linearmente; portanto, se a função for maior, ou seja, tiver um comprimento muito longo, esses erros serão ainda mais graves.

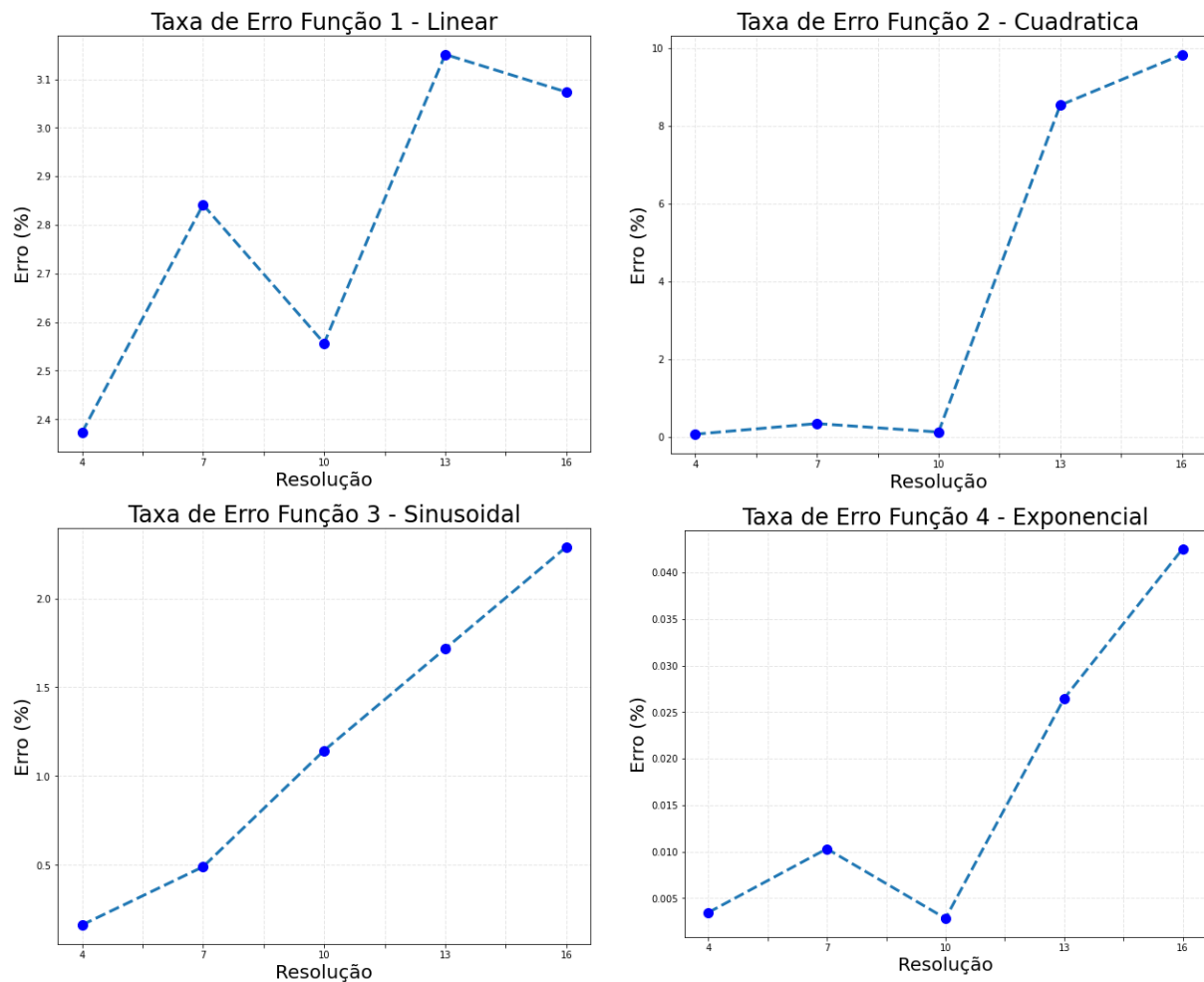


Figura 17 – Mudança da taxa de Erro para cada uma das funções usadas.

A função quadrática apresenta resultados interessantes, pois, sendo uma função simétrica, permite boa aproximação até uma resolução de 10 pontos, levando em consideração o número total de pontos utilizados para esta função (41) essa aproximação é bastante boa.

4.2 Generalização Cartografica

Agora, com o objetivo de apresentar uma aplicação de Strip-Tree, foi criada uma função (Polylinea.csv) com a qual é possível simular uma costa continental (figura 18). Foram calculados 5 níveis de resolução, onde a figura 19 mostra esses resultados.

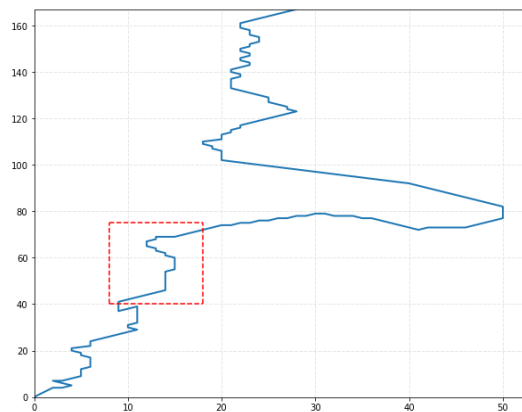


Figura 18 – Função que representa um litoral.

A caixa vermelha na figura 18 mostra o último nível de ampliação na figura 19 (resolução 2), enquanto a resolução 18 é a que tem a pior aproximação.

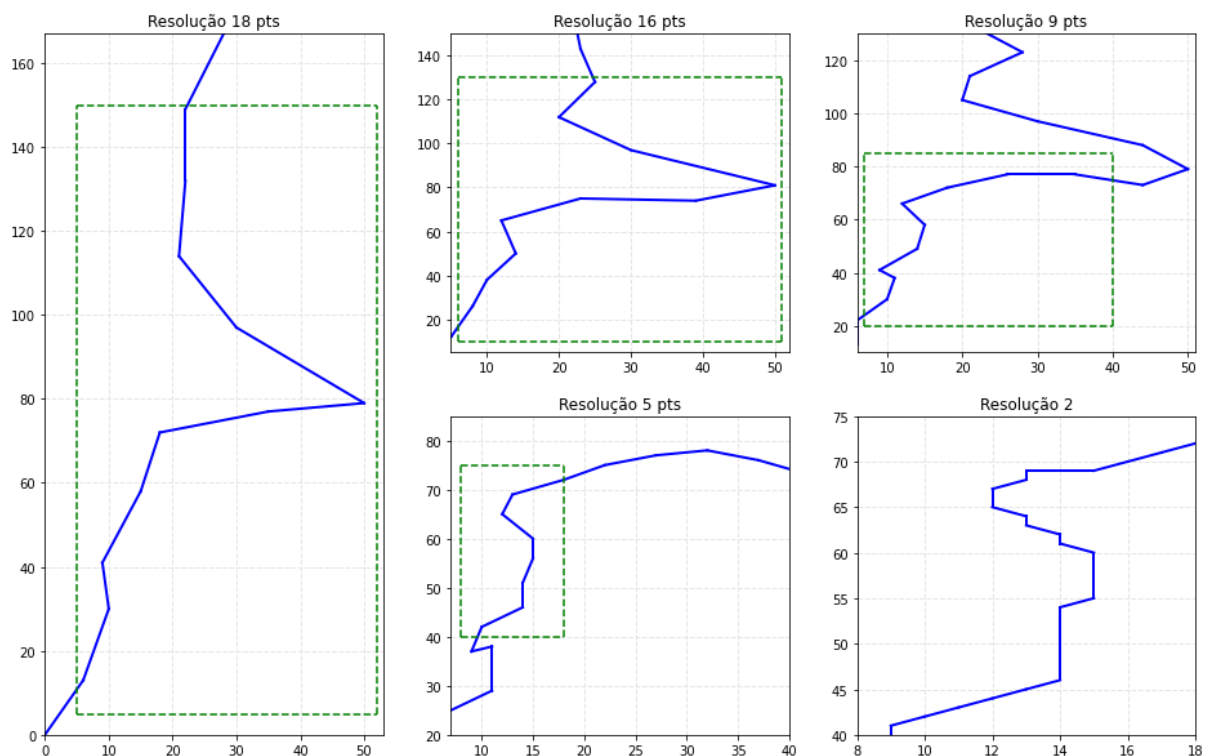


Figura 19 – Aplicação de Generalização cartográfica.

5 Conclusão

O Strip-Tree apresenta uma metodologia muito interessante, desde a digitalização da curva até como armazenar as informações dentro de uma Árvore Binária, levando em consideração que as métricas necessárias para obter o retângulo (Banda ou Strip) também são armazenadas, a operação mais simples a partir disso, é a obtenção do comprimento da curva (como foi apresentado na seção 4.1), porém este é apenas o primeiro passo, pois, usando o Strip-Tree, também pode-se obter a área, intersecção entre outras operações.

Deve-se notar que, para uma determinada resolução definida, será obtida a aproximação da curva, que a sua vez será armazenada no Strip-Tree; portanto, se deseja-se obter outra aproximação para uma resolução diferente (seja maior ou menor), Esta informação será armazenada em outra árvore; Isto é, cada árvore possui apenas 1 resolução para a curva; Se o objetivo é gerar 5 curvas cada uma com uma resolução, 5 árvores precisarão ser criadas. Juntamente com a metodologia ao momento de digitalizar a imagem, esta é a principal diferença entre o Strip-Tree e a árvore Arc-Tree. O Arc-Tree apresenta uma resolução diferente da curva em cada nível da árvore, portanto, uma única árvore pode conter resoluções diferentes.

No momento da digitalização, o número mínimo de pontos de acordo com cada caso era usado como valor de resolução; no entanto, o mais apropriado seria implementar o uso do limite de percepção visual, pois esse fator determina a apreciação máxima que podemos obter ao apreciar uma coordenar em um mapa ou plano e é de extrema importância na aplicação da generalização cartográfica.

Referências

BALLARD, D. H. Strip trees: A hierarchical representation for curves. *Communications of the ACM*, ACM New York, NY, USA, v. 24, n. 5, p. 310–321, 1981. Citado nas páginas 1, 3, 6, 7, 9 e 12.

DOUGLAS, D. H.; PEUCKER, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, University of Toronto Press, v. 10, n. 2, p. 112–122, 1973. Citado na página 4.

FIGUEIREDO, L. H. D.; STOLFI, J.; VELHO, L. Approximating parametric curves with strip trees using affine arithmetic. In: IEEE. *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*. [S.l.], 2002. p. 163–170. Citado nas páginas 1 e 10.

GÜNTHER, O.; WONG, E. The arc tree: an approximation scheme to represent arbitrary curved shapes. *Computer Vision, Graphics, and Image Processing*, Elsevier, v. 51, n. 3, p. 313–337, 1990. Citado nas páginas 1 e 9.

HERSHBERGER, J.; SNOEYINK, J. An $O(n \log n)$ implementation of the douglas-peucker algorithm for line simplification. In: *Proceedings of the tenth annual symposium on Computational geometry*. [S.l.: s.n.], 1994. p. 383–384. Citado na página 4.

HERSHBERGER, J. E.; SNOEYINK, J. *Speeding up the Douglas-Peucker line-simplification algorithm*. [S.l.]: University of British Columbia, Department of Computer Science Vancouver, BC, 1992. Citado na página 4.

JARAMILLO, L. Principios básicos de generalización cartográfica. Instituto Geográfico Militar, 2006. Citado nas páginas 1 e 5.

MCMASTER, R. B.; SHEA, K. S. Generalization in digital cartography. In: ASSOCIATION OF AMERICAN GEOGRAPHERS WASHINGTON, DC. [S.l.], 1992. Citado na página 5.

VISVALINGAM, M.; WHYATT, J. D. The douglas-peucker algorithm for line simplification: Re-evaluation through visualization. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 1990. v. 9, n. 3, p. 213–225. Citado na página 4.

WU, S.-T.; MARQUEZ, M. R. G. A non-self-intersection douglas-peucker algorithm. In: IEEE. *16th Brazilian symposium on computer graphics and Image Processing (SIBGRAPI 2003)*. [S.l.], 2003. p. 60–66. Citado nas páginas 1 e 4.