

ANDRÉ LUIZ BEHRENSDORF DERRAIK

UM ESTUDO COMPARATIVO DE REPRESENTAÇÕES DE  
MULTI-RESOLUÇÃO PARA LINHAS POLIGONAIS

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA

PUC-Rio

Rio de Janeiro, 25 de abril de 1997

ANDRÉ LUIZ BEHRENSDORF DERRAIK

UM ESTUDO COMPARATIVO DE REPRESENTAÇÕES DE  
MULTI-RESOLUÇÃO PARA LINHAS POLIGONAIS

Dissertação apresentada ao Departamento de  
Informática da PUC-Rio como parte dos  
requisitos para a obtenção do título de Mestre  
em Ciências em Informática.

Orientador: Marcelo Gattass

Co-orientador: Luiz Henrique de Figueiredo

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 25 de abril de 1997

A Liane e Karen

# Agradecimentos

A Liane e Karen pela compreensão e apoio dados durante todos esses anos.

Ao orientador e Prof. Marcelo Gattass pelo apoio e orientação nesses anos todos de minha vida acadêmica.

Ao amigo e orientador Prof. Luiz Henrique de Figueiredo, pela ajuda em todo o processo de construção deste trabalho.

A todos os amigos do TeCGraf, que de uma forma ou outra contribuíram para a realização deste trabalho.

A CAPES e ao SEPROC/CENPES/PETROBRÁS pelo auxílio financeiro.

# Resumo

Este trabalho apresenta um estudo comparativo de algumas representações em multi-resolução para linhas poligonais. São estudados as estruturas *strip tree*, *arc tree* e *box tree*, e suas variantes, comparando as velocidades de construção, de percorrimento (*drawing*), de operações de interseção e seleção (*pick*); e custo de armazenagem em memória. A comparação é feita usando bancos de dados cartográficos reais disponíveis na Internet.

O objetivo do estudo é identificar técnicas e algoritmos adequados para exploração interativa de grandes bancos de dados cartográficos.

# Abstract

This work presents a comparative study of some multiresolution representations for polygonal lines. We study the strip tree, the arc tree and the box tree data structures, comparing their performance for construction, drawing, intersection and selection; and memory storage costs. The comparison uses actual databases available in the Internet.

The goal of this study is to identify techniques and algorithms for interactive exploration of large cartographic data bases.

# Sumário

<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos.....	2
1.2 Organização .....	3
<b>2 Multi-resolução</b>	<b>4</b>
2.1 Definição.....	4
2.2 Aplicação .....	6
2.3 Operações .....	12
2.3.1 Desenho .....	12
2.3.2 Seleção.....	13
2.3.3 Interseção .....	15
2.3.4 Cerceamento .....	16
<b>3 Representações em Multi-resolução</b>	<b>19</b>
3.1 Strip Tree .....	19
3.2 Box Tree.....	23
3.3 Arc Tree.....	25
3.4 Dumb.....	28
3.5 Plain .....	29
3.6 BBox Tree - Uma Variante da Box Tree .....	30
3.7 Comparações na Implementação das Representações .....	30
<b>4 Testes e Avaliações</b>	<b>32</b>
4.1 Descrição dos Bancos de Dados .....	32
4.2 Resultados por Operação .....	43

4.2.1 Criação .....	43
4.2.2 Desenho .....	43
4.2.3 Desenho com Tolerância.....	45
4.2.4 Desenho com Tolerância e Cerceamento .....	45
4.2.5 Interseção .....	47
4.2.6 Seleção.....	48
<b>4.3 Resultados por Banco de Dados.....</b>	<b>49</b>
4.3.1 Estados do Brasil.....	49
4.3.2 Municípios do Brasil.....	49
4.3.3 Municípios dos Estados Unidos.....	50
4.3.4 CIA World Data Bank (África) .....	51
4.3.5 Curvas de Nível de Anchor .....	51
4.3.6 Curvas de Nível de Madison.....	51
4.3.7 Floresta Amazônica I .....	52
<b>4.4 Conclusões Preliminares .....</b>	<b>52</b>
<b>5 Conclusões</b>	<b>54</b>
5.1 Bancos de Dados .....	54
5.2 Considerações Gerais .....	57
5.3 Trabalhos Futuros.....	58
<b>Referências</b>	<b>60</b>



## Lista de Figuras

Figura 2.1: Representação da multi-resolução de forma abstrata. ....	5
Figura 2.2: Respresentação da curva da Figura 2.2 nos níveis 1 e 2.....	5
Figura 2.3: Mapa do Brasil, todos os segmentos desenhados. ....	6
Figura 2.4: Mapa do Brasil com tolerância de 1 <i>pixel</i> . ....	7
Figura 2.5: Mapa do Brasil com tolerância de 10 <i>pixels</i> .....	7
Figura 2.6: Mapa do Brasil com tolerância de 20 <i>pixels</i> .....	8
Figura 2.7: Mapa do Brasil com tolerância de 50 <i>pixels</i> .....	8
Figura 2.8: Mapa do Brasil com tolerância de 100 <i>pixels</i> .....	9
Figura 2.9: Mapa do Brasil com <i>zoom out</i> desenhando todos os segmentos. ....	10
Figura 2.10: Mapa do Brasil com <i>zoom out</i> e tolerância de 1 <i>pixel</i> .....	11
Figura 2.11: Mapa do Brasil sem <i>zoom</i> e a tolerância de 1 <i>pixel</i> .....	11
Figura 2.12: Definição do retângulo de seleção.....	14
Figura 2.13: O segmento <i>ab</i> é parcialmente visível. ....	18
Figura 3.1: Definição de uma <i>strip</i> .....	21
Figura 3.2: Exemplo de uma <i>strip tree</i> de nível 2.....	21
Figura 3.3: Representação da árvore do exemplo anterior.....	23
Figura 3.4: Definição da <i>box</i> . ....	23
Figura 3.5: Exemplo de uma <i>box tree</i> . ....	25
Figura 3.6: Definição da região limitante da <i>arc tree</i> . ....	26
Figura 4.1: Formato externo dos Bancos de Dados. ....	32
Figura 4.2: Banco de dados do Continente Africano. ....	36
Figura 4.3: Histograma da distribuição do número de pontos por poligonal da CIA World Data Bank (África). ....	36
Figura 4.4: Banco de dados da Floresta Amazônica I.....	37
Figura 4.5: Histograma da distribuição do número de pontos por poligonal do banco de dados da Floresta Amazônica I. ....	37

Figura 4.6: Banco de dados dos Municípios do Brasil. ....	38
Figura 4.7: Histograma da distribuição do número de pontos por poligonal dos Municípios do Brasil. ....	38
Figura 4.8: Banco de dados de Curvas de Nível de Madison. ....	39
Figura 4.9: Histograma da distribuição do número de pontos por poligonal das Curvas de Nível de Madison. ....	39
Figura 4.10: Banco de dados de Curvas de Nível de Anchor. ....	40
Figura 4.11: Histograma da distribuição do número de pontos por poligonal das Curvas de Nível de Anchor. ....	40
Figura 4.12: Banco de dados dos Municípios dos Estados Unidos. ....	41
Figura 4.13: Histograma da distribuição do número de pontos por poligonal dos Municípios dos Estados Unidos. ....	41
Figura 4.14: Banco de dados dos Estados do Brasil. ....	42
Figura 4.15: Histograma da distribuição do número de pontos por poligonal dos Estados do Brasil. ....	42
Figura 4.16: Definição do tamanho da área de desenho. ....	44
Figura 4.17: Diminuição da área de desenho em 200 <i>pixels</i> de cada borda.....	46
Figura 4.18: Linha poligonal englobando toda a região do desenho para o cálculo da interseção. ....	47
Figura 5.1: Exemplo de Mapa Político: banco de dados dos Municípios dos Estados Unidos. ....	55
Figura 5.2: Exemplo de Curvas de Nível: banco de dados de Madison. ....	55
Figura 5.3: Exemplo de Mapas Temáticos, banco de dados de CIA World Data Bank, Continente Africano. ....	56

## Lista de Tabelas

Tabela 2.1: Comparação do número de segmentos e tamanhos em <i>bytes</i> dos desenhos nas diversas resoluções. ....	10
Tabela 2.2: Comparação dos números de segmentos e tamanhos em <i>bytes</i> dos desenhos com <i>zoom out</i> e sem <i>zoom</i> . ....	12
Tabela 3.1: Médias e desvio padrão das correlações da regressão dos comprimentos acumulados de cada poligonal. ....	27
Tabela 4.1: Principais características dos bancos de dados utilizados nos testes. ....	34
Tabela 4.2: Médias, variâncias, máximos e mínimos dos números de pontos por poligonal dos bancos de dados. ....	35
Tabela 4.3: Tempos em segundos para a operação de criação. ....	43
Tabela 4.4: Tempos em segundos para a operação de desenho sem tolerância. ....	44
Tabela 4.5: Tempos em segundos para a operação de desenho com tolerância de 1 <i>pixel</i> . ....	45
Tabela 4.6: Tempos em segundos para a operação de desenho com tolerância de 1 <i>pixel</i> e <i>cerceamento</i> . ....	47
Tabela 4.7: Tempos em segundos para a operação de interseção. ....	48
Tabela 4.8: Tempos em segundos para a operação de seleção. ....	48
Tabela 4.9: Comparação das estruturas para o banco de dados dos Estados do Brasil. ....	49
Tabela 4.10: Comparação das estruturas para o banco de dados dos Municípios do Brasil. ....	50
Tabela 4.11: Comparação das estruturas para o banco de dados dos Municípios dos Estados Unidos. ....	50
Tabela 4.12: Comparação das estruturas para o banco de dados do CIA World Data Bank, Continente Africano. ....	51
Tabela 4.13: Comparação das estruturas para o banco de dados das Curvas de Nível de Anchor. ....	51
Tabela 4.14: Comparação das estruturas para o banco de dados das Curvas de Nível de Madison. ....	52

Tabela 4.15: Comparação das estruturas para o banco de dados da Floresta Amazônica I.....	52
Tabela 4.16: Comparação do tamanho do nós de cada representação. ....	53
Tabela 5.1: Resultado das melhores respresentações separados por tipos de banco de dados.....	57

# Capítulo 1

## Introdução

A exploração interativa de grandes bancos de dados cartográficos é um problema importante, dada a grande quantidade de informações a serem processadas. Tipicamente, um banco de dados cartográficos contém milhares de linhas poligonais com milhões de vértices; esse volume de dados pode impedir a exploração interativa.

Bancos de dados cartográficos geralmente contêm dados em alta resolução, obtidos por digitalização automática ou manual de fotografias. Entretanto, somente uma pequena fração destes dados pode ser exibida na tela. Em outras palavras, a resolução original dos dados não é necessária para gerar desenhos. Além disso, gerar um desenho usando a resolução original pode ser proibitivo numa aplicação interativa.

Uma técnica para diminuir o volume de dados a serem exibidos é representar as linhas poligonais em multi-resolução. Uma representação em multi-resolução de uma curva é uma representação hierárquica da curva, que pode ser percorrida somente até o nível adequado para a escala desejada [6]. Uma representação em multi-resolução permite calcular uma aproximação da poligonal que é adequada à resolução desejada<sup>1</sup>, ao mesmo tempo que

---

<sup>1</sup> Esse processo é chamado “generalização” em cartografia.

mantém intactos os dados originais. Desta maneira, é possível fazer *zooms* contínuos (até, claro, a resolução máxima dos dados).

Representações em multi-resolução são claramente úteis em síntese de imagens, pois reduzem o volume de dados a somente o necessário para gerar uma imagem numa dada escala. Esta é uma condição necessária para interatividade. Além disso, representações em multi-resolução também são muito úteis para processamento geométrico adaptativo, como cálculo de interseções e localização de pontos.

## 1.1 Objetivos

Este trabalho faz uma comparação de desempenho de algumas representações em multi-resolução elementares na exploração interativa de grandes bancos de dados cartográficos, com o objetivo principal de obter informações que permitam recomendar a escolha de uma das representações que se adapte melhor para cada tipo de dado. A única comparação encontrada na literatura [6] não usa dados cartográficos reais e sim linhas poligonais com poucos vértices (fontes de caracteres em *outline*).

Neste trabalho consideraremos somente representações elementares para linhas poligonais, embora recentemente haja grande interesse em métodos de multi-resolução usando *wavelets* [4].

Vale ressaltar que, neste trabalho, as estruturas de dados representando as linhas poligonais em multi-resolução serão todas armazenadas em memória principal e não em disco. Adotamos esta simplificação para poder fazer uma comparação isenta. Na prática, é necessário combinar representações em multi-resolução com técnicas de acesso eficiente a dados em disco [8, 9, 11].

## **1.2 Organização**

O Capítulo 2 apresenta uma definição abstrata de representação em multi-resolução e as definições das operações que serão realizadas nas comparações: visualização, seleção, interseção e cerceamento, incluindo algoritmos abstratos para execução destas operações.

No Capítulo 3 contém as definições e pseudo-código das representações estudadas.

O Capítulo 4 contém os procedimentos usados na comparação e os resultados das avaliações das representações. Nele são apresentados os bancos de dados utilizados e os resultados dos testes nas diversas estruturas implementadas.

No Capítulo 5, apresentamos as conclusões baseadas nos testes e sugestões para futuros trabalhos.

# Capítulo 2

## Multi-resolução

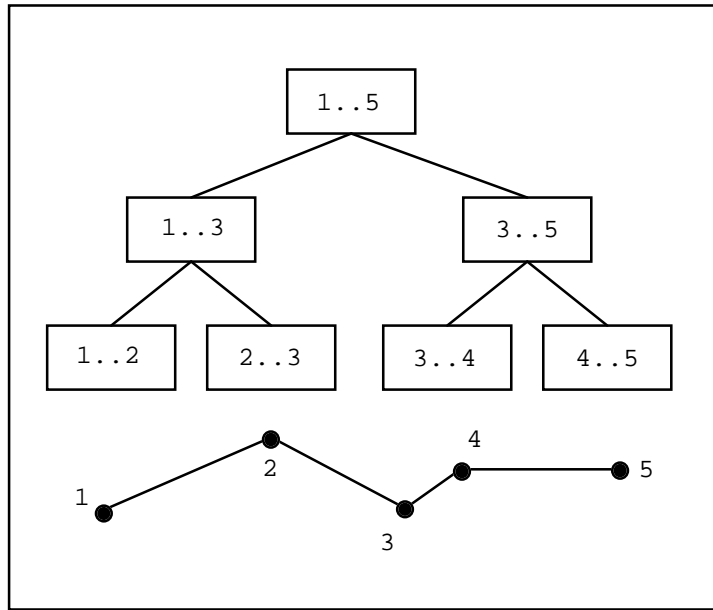
Neste capítulo, definimos o problema que estudamos e descrevemos os principais algoritmos para execução das operações primitivas escolhidas para a comparação empírica.

### 2.1 Definição

Abstratamente, uma *representação em multi-resolução* para uma linha poligonal  $p_1, \dots, p_n$  é uma árvore, normalmente binária, cujos nós representam *regiões limitantes* que contêm trechos da poligonal. Como veremos, cada escolha de região limitante define uma representação em multi-resolução. As folhas da árvore contêm os segmentos da linha poligonal, propriamente ditos. Mais precisamente, os nós internos estão associados a um trecho  $p_i, \dots, p_j$  e contêm uma descrição geométrica de um conjunto  $R_{i,j}$  que contém completamente a sublinha poligonal  $p_i, \dots, p_j$ . Como, normalmente, usam-se regiões convexas, basta que  $R_{i,j}$  contenha os pontos  $p_i, \dots, p_j$  do trecho associado ao nó.

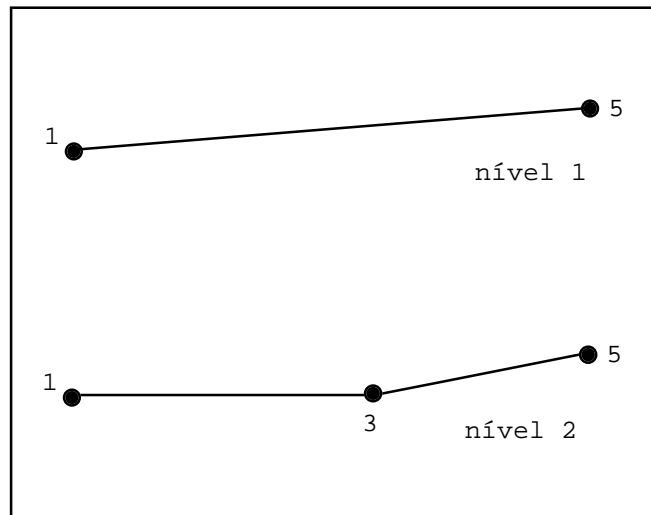
A profundidade dos nós intermediários destas árvores representa o nível de resolução (tolerância) que se deseja representar a poligonal original. A Figura 2.2 mostra a poligonal do exemplo da Figura 2.1 nos níveis de resolução 1 e 2.





**Figura 2.1: Representação da multi-resolução de forma abstrata.**

A Figura 2.2 mostra uma representação abstrata em multi-resolução de uma linha poligonal com 5 vértices. (Na prática, as linhas poligonais podem ter milhares de vértices.)



**Figura 2.2: Respresentação da curva da Figura 2.2 nos níveis 1 e 2.**

As regiões limitantes  $R_{ij}$  fornecem informações sobre a extensão geométrica do trecho associado  $p_i, \dots, p_j$ . Nas operações elementares que descrevemos na Seção 2.3, essas informações são usadas essencialmente de duas maneiras:

- **aproximação:** se uma região limitante  $R_{ij}$  tem diâmetro menor que uma tolerância (e.g., está contida num *pixel*), então todo o trecho contido na região pode ser substituído por uma versão muito mais simples: o segmento  $p_{ij}$  ou mesmo o ponto  $p_i$  ou mesmo ignorado. Isto é utilizado principalmente para síntese de imagens.
- **exclusão:** Os trechos cujas regiões limitantes estão longe da área de interesse para a operação podem ser imediatamente descartados. Em grandes bancos de dados, é essencial descartar rapidamente elementos que não participam da operação corrente.

## 2.2 Aplicação

As Figuras 2.3 a 2.8 ilustram a aplicação de multi-resolução num mapa dos Estados do Brasil utilizando várias tolerâncias: sem tolerância (ie., 0 *pixels*), 1 *pixel*, 10, 20, 50 e 100 *pixels* de tolerância.



**Figura 2.3: Mapa do Brasil, todos os segmentos desenhados.**



**Figura 2.4:** Mapa do Brasil com tolerância de 1 *pixel*.



**Figura 2.5:** Mapa do Brasil com tolerância de 10 *pixels*.



**Figura 2.6:** Mapa do Brasil com tolerância de 20 *pixels*.



**Figura 2.7:** Mapa do Brasil com tolerância de 50 *pixels*.



**Figura 2.8: Mapa do Brasil com tolerância de 100 *pixels*.**

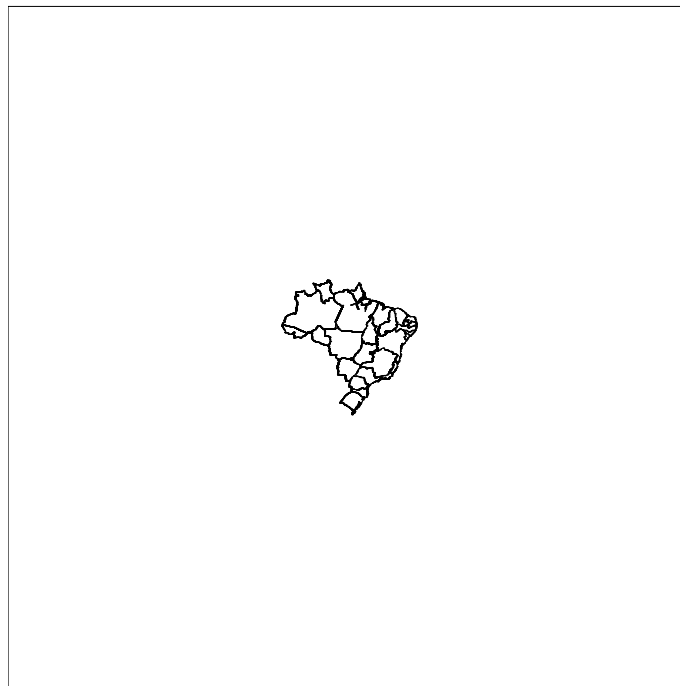
Um ponto importante da representação em multi-resolução é que ela preserva a característica visual do desenho, independente da precisão com a qual o desenho está sendo exibido. Isto é, podemos desenhar tanto todos os segmentos quanto desenhar apenas as raízes das árvores (baixa resolução) que visualmente conseguimos identificar o desenho.

A Tabela 2.1 apresenta uma comparação entre as diversas resoluções. Os valores em *bytes* referem-se aos tamanhos dos arquivos em PostScript gerados.

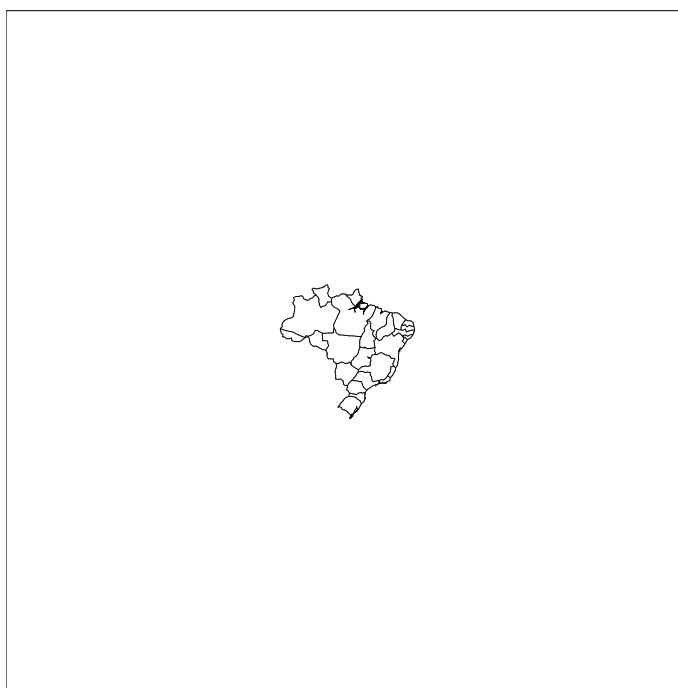
Tolerância em <i>pixels</i>	Número de Segmentos	Tamanho em <i>bytes</i>
<b>0</b>	20.955	558.583
<b>1</b>	16.285	433.817
<b>10</b>	3.585	96.371
<b>20</b>	1.685	45.547
<b>50</b>	643	17.759
<b>100</b>	363	10.281

**Tabela 2.1: Comparação do número de segmentos e tamanhos em *bytes* dos desenhos nas diversas resoluções.**

As Figuras 2.9, 2.10 e 2.11 mostram uma comparação entre os desenhos mantendo a mesma tolerância, isto é, um *zoom out* é feito no desenho e a mesma resolução sem o *zoom*.



**Figura 2.9: Mapa do Brasil com *zoom out* desenhando todos os segmentos.**



**Figura 2.10:** Mapa do Brasil com *zoom out* e tolerância de 1 *pixel*.



**Figura 2.11:** Mapa do Brasil sem *zoom* e a tolerância de 1 *pixel*.

A Tabela 2.2 mostra a comparação dos números de segmentos e tamanhos em *bytes* dos desenhos (arquivos PostScript).

Tolerância em <i>pixel</i>	Número de Segmentos	Tamanho em <i>bytes</i>
<b>0</b>	20.955	570.666
<b>1</b>	637	24.965
<b>1 (sem zoom)</b>	637	20.221

**Tabela 2.2:** Comparação dos números de segmentos e tamanhos em *bytes* dos desenhos com *zoom out* e *sem zoom*.

Os diferentes tamanhos em *bytes* para a mesma tolerância deve-se ao fato da construção dos arquivos PostScript. O importante é notar a significativa diferença entre os tamanhos dos desenhos sem tolerância e com 1 *pixel* de tolerância. O arquivo correspondente a 1 *pixel* da tolerância tem menos de 5% do tamanho original mas gera um desenho exatamente igual.

## 2.3 Operações

Neste estudo, estamos interessados nas seguintes operações elementares, típicas de uma exploração interativa: desenho, seleção, interseção e cerceamento. Estas operações estão definidas precisamente abaixo, juntamente com os algoritmos que implementam essas operações sobre representações em multi-resolução.

Nesses algoritmos, uma estrutura de multi-resolução abstrata tem em cada nó os campos:  $(i, j, R = R_{ij}, \text{left}, \text{right})$ .

### 2.3.1 Desenho

O procedimento de desenho (*draw*) consiste na visualização das curvas dada uma tolerância  $\epsilon$ . É a operação fundamental – tem que ser eficiente. Esta resolução irá indicar até que nível deve-se descer na árvore para a sua visualização. A aproximação calculada é tal que as regiões limitantes têm diâmetro menor que  $\epsilon$ , conforme o seguinte algoritmo:



---

**Algoritmo:** Desenha uma linha poligonal dada uma tolerância  $\epsilon$ .

```
draw(t):  
    se diâmetro de t.R <  $\epsilon$  ou t é folha então  
        desenha segmento  $p_{ij}$   
    senão  
        draw(t.left)  
        draw(t.right)
```

---

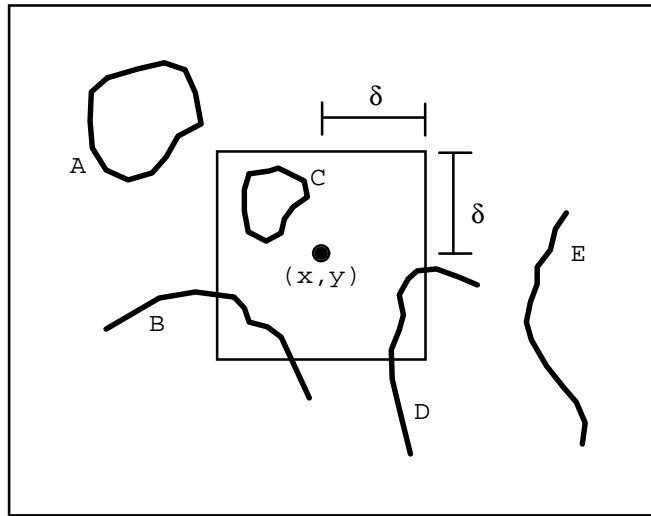
### 2.3.2 Seleção

A operação de seleção (*pick*) numa exploração interativa é a que permite identificar os objetos cartográficos que se deseja maiores informações, como por exemplo, fornecidas por um banco de dados convencional.

Geralmente, a seleção consiste em responder a seguinte pergunta:

“Dado um ponto  $(x,y)$ , e uma tolerância  $\delta$ , quais são as curvas que passam pelo retângulo  $\Delta = [x-\delta, x+\delta] \times [y-\delta, y+\delta]$ ?”

No exemplo da Figura 2.12, as linhas A e E não são selecionadas enquanto B, C e D são. Geralmente o valor de  $\delta$  é bem pequeno, da ordem de alguns *pixels*.



**Figura 2.12: Definição do retângulo de seleção.**

---

**Algoritmo:** Seleciona as linhas poligonais que passam pelo retângulo

$$\Delta = [x-\delta, x+\delta] \times [y-\delta, y+\delta].$$

`pick(t, Δ):`

```

se t.R ∩ Δ ≠ ∅ então
  se t é folha então
    se pipj ∩ Δ ≠ ∅ então
      seleciona t
  senão
    pick(t.left)
    pick(t.right)

```

---

O teste geométrico fundamental nesse algoritmo é o cálculo da interseção do segmento  $p_i p_j$  com o retângulo  $\Delta$ . Esta é a operação clássica de *clipping* (cerceamento), que pode ser implementada com um dos vários algoritmos de *clipping*, por exemplo, o algoritmo de Cohen e Sutherland [10].

### 2.3.3 Interseção

Esta operação faz os cálculos dos pontos de interseção de duas linhas poligonais.

---

**Algoritmo:** Calcula as interseções das linhas poligonais A e B.

```
inter(a,b): a = (i,j,Rij)      b = (k,l,Rkl)
  se a.R ∩ b.R ≠ ∅ então
    se a é folha e b é folha então
      calcula interseção entre pij e pkl
    senão se a é folha então
      inter(a,b.left)
      inter(a,b.right)
    senão se b é folha então
      inter(a.left,b)
      inter(a.right,b)
    senão
      inter(a.left,b.left)
      inter(a.right,b.left)
      inter(a.left,b.right)
      inter(a.right,b.right)
```

---

A operação geométrica fundamental nesse algoritmo é o cálculo da interseção de dois segmentos  $ab$  e  $cd$ , que é comum para todas as representações. O cálculo é feito através das equações das retas parametrizadas de  $ab$  e  $cd$ :

$$a + t(b-a) = c + s(d-c) \quad \text{onde: } s, t \in [0,1]$$

---

**Algoritmo:** Cálculo de interseção entre dois segmentos ab e cd.

```
interseg(ab,cd):
    -- determinante do sistema de equações: a + t(b-a) = c + s(d-c)
    det = (bx-ax)(cy-dy) - (by-ay)(cx-dx)

    -- ab e cd são paralelos se det é zero
    se det = 0 então
        retorna não há interseção

    -- cálculo dos parâmetros t e s
    t = [(cy-dy)(cx-ax) + (dx-cx)(cy-ay)] / det
    s = [(ay-by)(cx-ax) + (bx-ax)(cy-ay)] / det

    -- verifica se houve interseção nos segmentos
    se t ∉ [0,1] ou s ∉ [0,1] então
        retorna não há interseção

    -- cálculo do ponto de interseção
    retorna p = a + t(b-a)
```

---

### 2.3.4 Cerceamento

Uma das operações mais frequentes na exploração interativa de banco de dados cartográficos é *zoom in/out*, dada uma área de interesse selecionada pelo usuário. É essencial que essa operação seja rápida.

O cerceamento (*clipping*) objetiva acelerar a operação de desenho numa área de interesse. Na operação de desenho, o resultado é uma representação em multi-resolução de

todas as linhas poligonais. No cerceamento, antes de desenhar as linhas poligonais, testa-se se parte da poligonal está visível. Este teste permite descartar imediatamente várias poligonais.

O cerceamento define-se como: dada uma poligonal  $t$ , um retângulo  $\Delta$  e uma resolução  $\epsilon$ , saber que parte de  $t$  está contida no retângulo  $\Delta$  e desenhá-la com a resolução  $\epsilon$ .

---

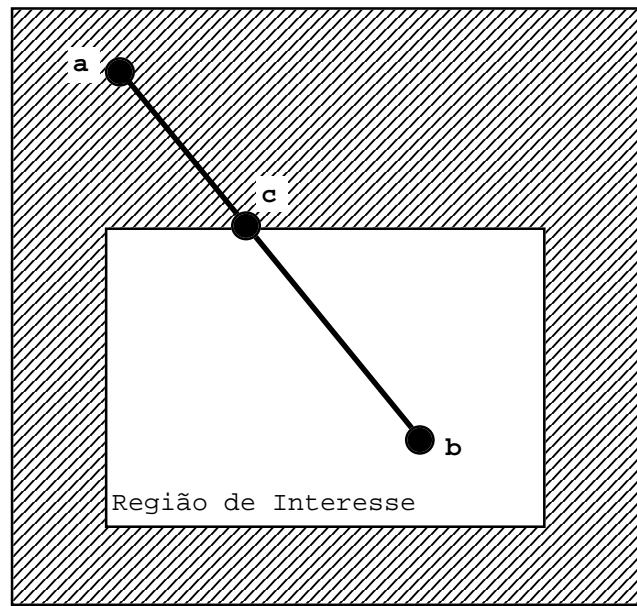
**Algoritmo:** Desenha uma linha poligonal dada uma resolução  $\epsilon$  e um retângulo  $\Delta$ .

`clip(t,  $\Delta$ ):`

```
se  $t.R \cap \Delta \neq \emptyset$  então
    se diâmetro de  $t.R < \epsilon$  ou  $t$  é folha então
        desenha segmento  $p_{ij}$ 
    senão
        clip( $t.left$ )
        clip( $t.right$ )
```

---

Esta operação não faz o cerceamento em si; apenas indica se o trecho da poligonal em questão é ou não visível. Ela não faz os cálculos dos pontos de interseção da poligonal com a região de visualização. Isto é, se o segmento  $ab$  está contido parcialmente na região de visualização, então o algoritmo irá responder que  $ab$  está visível. O cálculo do ponto  $c$  tal que  $ac$  não pertence à região de visualização e  $cb$  pertence, é deixado para o sistema gráfico utilizado (Figura 2.13).



**Figura 2.13: O segmento ab é parcialmente visível.**

Apesar de os algoritmos de cerceamento usados nos sistemas gráficos serem, em geral, eficientes ainda é necessário esse cerceamento em alto nível dado o grande volume de dados em questão.

## Capítulo 3

# Representações em Multi-resolução

Neste capítulo, descrevemos com detalhes as representações em multi-resolução estudadas: *strip tree*, *box tree*, *arc tree*, *dumb*, *plain* e *bbox tree*.

### 3.1 Strip Tree

A *strip tree* proposta por Ballard [1] é uma representação hierárquica de uma curva obtida por meio de retângulos contendo trechos da curva. Como descrito na Seção 2.1, a curva é representada por uma árvore binária. Os retângulos envolventes não são, em geral, paralelos aos eixos XY (Figura 3.1).

---

**Estrutura:** *strip tree*.

a = ponto inicial do segmento

b = ponto final do segmento

$w_l$  = distância máxima à esquerda de ab

$w_r$  = distância máxima à direita de ab

left = sub-árvore à esquerda

right = sub-árvore à direita

---

Na raiz de uma *strip tree*, temos um retângulo com dois lados paralelos ao segmento  $ab$ , onde  $a$  e  $b$  são os extremos da curva. Esses lados passam pelos pontos  $c$  e  $d$  mais distantes de  $ab$  em cada lado. No próximo nível, a curva é subdividida em  $c$  ou  $d$ , dependendo do que estiver mais longe de  $ab$  (digamos,  $c$ ), e representada com retângulos contendo os dois trechos  $ac$  e  $cb$ . Esse processo de subdivisão é repetido recursivamente nos dois trechos, até que os retângulos tenham largura menor que uma tolerância.

O refinamento é feito subdividindo-se a curva de um nó em duas partes de acordo com o seguinte algoritmo:

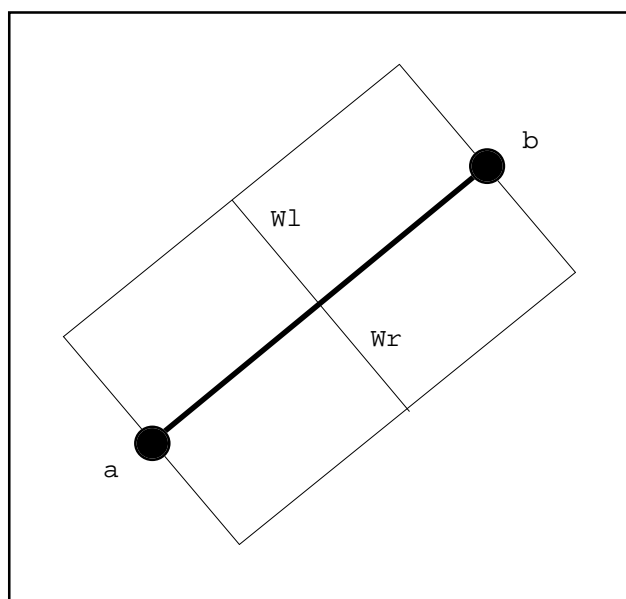
---

**Algoritmo:** Construção da *strip tree*.

```
strip(i, j):  
    t.a = pi  
    t.b = pj  
    l = pipj  
    t.wl = distância máxima à esquerda [l, Pi,j]  
    t.wr = distância máxima à direita [l, Pi,j]  
    k ∈ [i, j] | w(pk) = max (t.wl, t.wr)  
    t.left = strip(i, k)  
    t.right = strip(k, j)  
return t
```

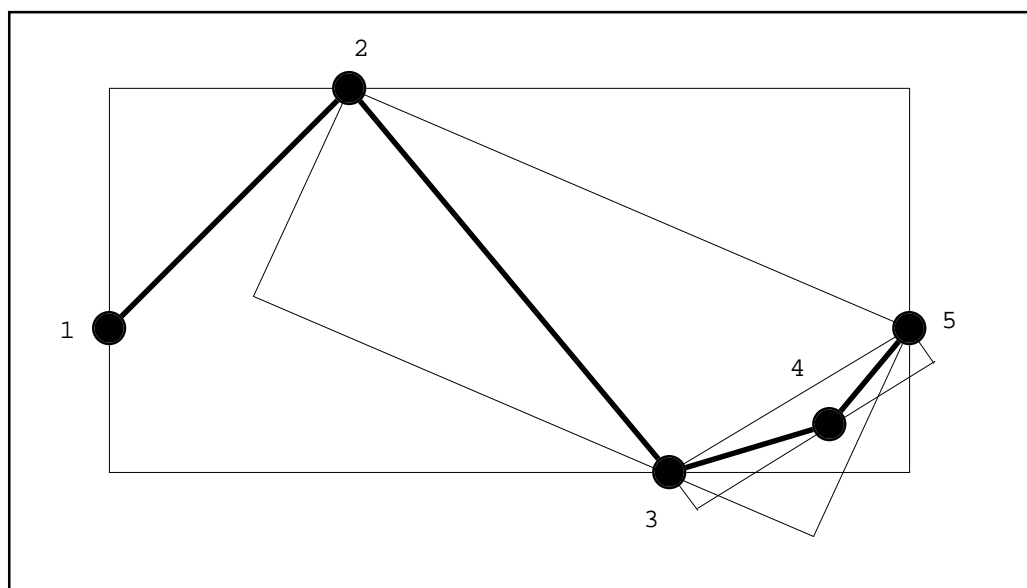
---





**Figura 3.1: Definição de uma strip.**

A Figura 3.2 apresenta um exemplo de dois níveis de recursão do algoritmo.



**Figura 3.2: Exemplo de uma *strip tree* de nível 2.**

Como o objetivo deste estudo é a representação das poligonais sem perda e alteração dos dados originais, a recursão do algoritmo vai até que a variável  $w$  atinja valor 0. O resultado final é uma representação completa da poligonal pela *strip tree*, isto é, nas folhas da árvore encontram-se todos os vértices da poligonal (Figura 3.3).

Esta estrutura de dados está intimamente ligada com o algoritmo de Douglas-Peucker para redução de pontos numa linha poligonal (generalização) [3]. As poligonais obtidas por digitalização tipicamente têm muito mais pontos do que é necessário tanto para gerar imagens quanto para processamento geométrico. Uma maneira de reduzir o número de pontos em uma poligonal  $p_1, \dots, p_n$  é começar com o segmento  $p_1p_n$  e calcular o ponto  $p_k$  mais distante deste segmento. Se a distância de  $p_k$  a  $p_1p_n$  for menor que a tolerância desejada, então todos os pontos estão perto do segmento  $p_1p_n$ , e a linha poligonal se reduz a esse único segmento. Caso contrário, processa-se recursivamente os trechos  $p_1p_k$  e  $p_kp_n$ . Ao final do processo, os pontos visitados formam uma linha poligonal que aproxima bem a linha poligonal original, mas que geralmente tem muito menos pontos.

Armazenando os pontos visitados e as distâncias desses pontos ao segmento corrente, e encerrando a recursão somente quando o trecho não puder ser subdividido por não ter pontos interiores, obtemos uma estrutura de dados que representa a linha poligonal completamente, e que é útil para processamento em qualquer resolução. Esta estrutura de dados é na verdade uma *strip tree* simplificada.

Note o deslanceamento existente na árvore da Figura 3.3. Na Seção 3.7, discutimos os problemas existentes nas representações.

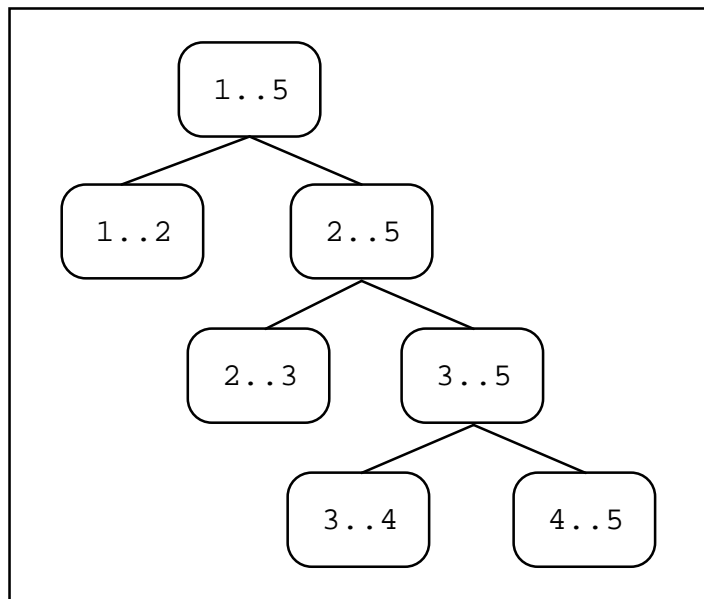


Figura 3.3: Representação da árvore do exemplo anterior.

### 3.2 Box Tree

Numa *Box tree*, as regiões limitantes são retângulos paralelos aos eixos XY. Apesar de utilizar o mesmo elemento geométrico (retângulos) que a *strip tree*, sua construção é muito mais simples (Figura 3.4).

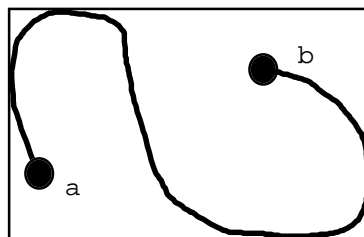


Figura 3.4: Definição da box.

---

**Estrutura:** *box tree*.

a = ponto inicial do segmento  
b = ponto final do segmento  
R = retângulo envolvente (xmin, xmax, ymin, ymax)  
left = sub-árvore à esquerda  
right = sub-árvore à direita

---

O refinamento da curva é feito na divisão ao meio do número de pontos  $n/2$ , onde  $n$  é o número de pontos do trecho, gerando uma árvore binária balanceada<sup>2</sup> (Figura 3.5).

---

**Algoritmo:** Construção da *box tree*.

```
box(i, j):  
    t.a = pi  
    t.b = pj  
    t.R = retângulo (pi, ..., pj)  
    k = (i+j)/2  
    t.left = box(i, k)  
    t.right = box(k, j)  
    return t
```

---

---

<sup>2</sup> Numa árvore balanceada, a maior diferença entre os níveis de quaisquer duas folhas é no máximo 1.

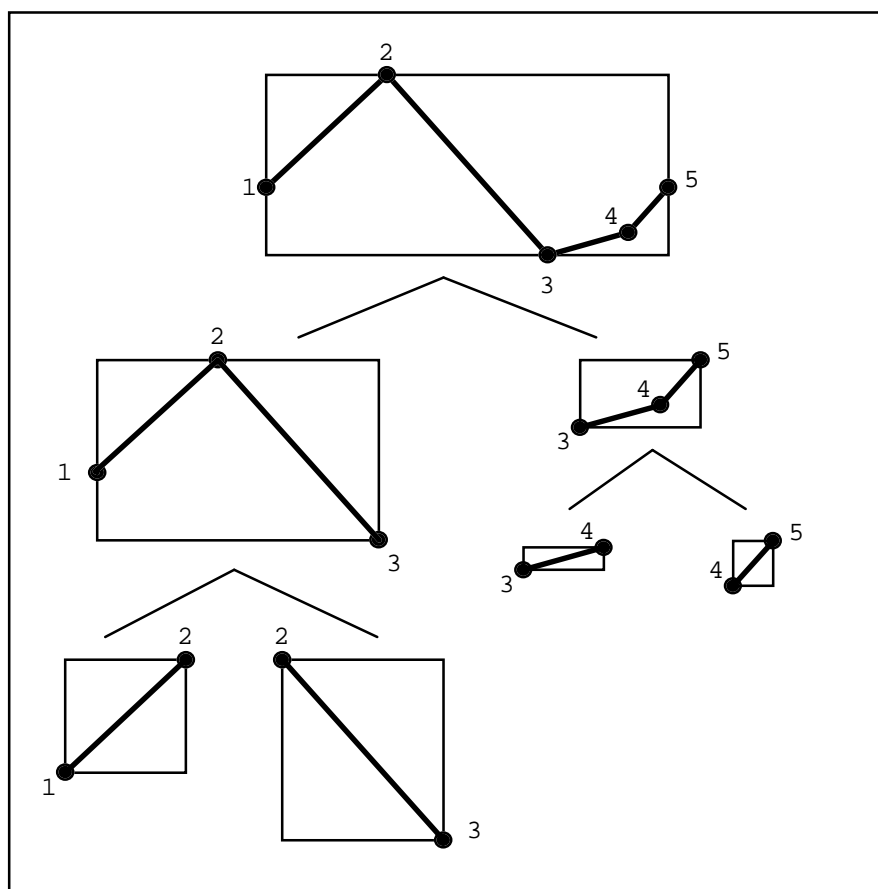


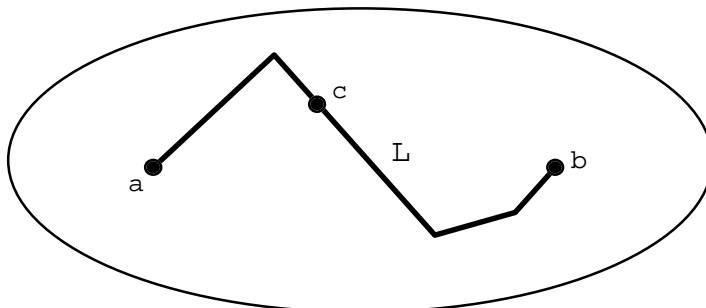
Figura 3.5: Exemplo de uma *box tree*.

### 3.3 Arc Tree

Uma alternativa para a escolha do ponto de subdivisão é o ponto médio do trecho da curva sendo considerado. Este é o ponto que está a meio caminho entre os extremos, onde a distância é medida *sobre* a curva. Para isto, o mais conveniente é que a curva esteja parametrizada pelo comprimento de arco. Isso é relativamente simples de garantir, especialmente para linhas poligonais. Entretanto, para linhas poligonais é ainda mais simples usar o ponto cujo índice está a meio caminho, e armazenar o comprimento de arco até esse ponto.

Uma *arc tree* é a árvore binária correspondente a uma subdivisão nos pontos médios [5]. As regiões limitantes da *arc tree* são elipses com focos nos extremos do trecho da poligonal e “comprimento”  $L$  igual ao comprimento do trecho da poligonal (Figura 3.6). Isto se deve ao

fato de que qualquer ponto  $c$  que esteja entre  $a$  e  $b$ , estará a uma distância menor que  $L$ , consequentemente, estará dentro da elipse.



**Figura 3.6:** Definição da região limitante da *arc tree*.

Como descrito na Seção 2.1, uma característica importante de representações hierárquicas é a existência de algum conjunto simples que contenha um trecho da curva em cada nível. Numa *strip tree*, esse conjunto é um retângulo. Numa *arc tree*, esse conjunto é uma elipse, que pode ser substituída por um círculo, ainda que sejam maiores do que necessários, pois os testes geométricos com círculos são mais simples.

Adotamos esta substituição (já proposta em [5]), o que simplificou bastante o procedimento de criação e a estrutura interna da *arc tree*.

---

**Estrutura:** *arc tree*.

$a$  = ponto inicial do segmento

$b$  = ponto final do segmento

$R$  = raio do círculo

$c$  = centro do círculo

$left$  = sub-árvore à esquerda

$right$  = sub-árvore à direita

---

Fizemos uma análise estatística dos comprimentos dos segmentos das linhas poligonais dos bancos de dados cartográficos escolhidos para o estudo, e observamos que os

comprimentos são quase constantes para cada poligonal. Partimos de seguinte hipótese: sendo os comprimentos dos segmentos constantes, plotando o comprimento acumulado contra o índice obteremos uma reta. A confirmação dessa hipótese justificaria a escolha da divisão pela metade do número de pontos e não pela metade do comprimento. Para verificar essa hipótese, utilizamos o método da regressão dos mínimos quadrados, onde calcula-se uma reta e o quão bom ela se aproxima dos dados originais.

Os resultados obtidos estão resumidos na Tabela 3.1, onde apresentamos as médias e desvios padrão das correlações da regressão dos mínimos quadrados utilizando como entrada os comprimentos acumulados de cada poligonal. Valores de  $\omega$  iguais a 1 representam 100% de correlação.

<b>Dados</b>	<b><math>\omega</math></b>	<b><math>\sigma</math></b>
<b>CIA World Data Bank</b>	0.999	0.007
<b>Vegetação Amazônica</b>	0.991	0.017
<b>Municípios do Brasil</b>	0.982	0.037
<b>Curvas de Nível (Madison)</b>	0.995	0.009
<b>Curvas de Nível (Anchor)</b>	0.991	0.014
<b>Municípios dos Estados Unidos</b>	0.975	0.029
<b>Estados do Brasil</b>	0.987	0.020

$\omega$  = Média da correlação       $\sigma$  = Desvio padrão da correlação

**Tabela 3.1: Médias e desvio padrão das correlações da regressão dos comprimentos acumulados de cada poligonal.**

A Tabela 3.1 mostra que a aproximação das somas dos tamanhos de cada poligonal é próxima a uma reta e a variância destas médias é bem baixa. Sendo assim, pode-se afirmar que cada segmento de cada poligonal tem aproximadamente o mesmo comprimento, e podemos aproximar a divisão por comprimento de arco pela divisão pelo número dos pontos. Com esta simplificação, conseguimos gerar uma árvore binária balanceada, assim como na *box tree*.

---

**Algoritmo:** Construção da *arc tree*.

```
arc(i, j):  
    t.a = pi  
    t.b = pj  
    t.R = comprimento da poligonal (pi, ..., pj) / 2  
    t.c = ponto médio de pipj  
    k = (i+j)/2  
    t.left = arc(i, k)  
    t.right = arc(k, j)  
    return t
```

---

### 3.4 Dumb

A estrutura *dumb* não é uma representação em multi-resolução, mas sim a mais simples representação de poligonais: um *array* contendo todos os vértices da poligonal. Esta é a estrutura mais frequentemente utilizada nos programas.

---

**Estrutura:** *dumb structure*.

```
n = número de vértices  
P[1..n] = vértices
```

---

Incluimos esta estrutura no presente estudo para a comparação dos tempos das operações, ou seja, para saber como se comporta um banco de dados cartográficos sem uma representação em multi-resolução. Queremos, ao mesmo tempo, avaliar os benefícios e os custos de usar uma representação em multi-resolução.



### 3.5 Plain

Nesta representação, adicionamos um pouco de “inteligência” à estrutura *dumb*: Simplesmente acrescentamos uma região limitante global. Esta região é um retângulo paralelo aos eixos XY, como a *box tree*, mas só no nível mais alto.

---

**Estrutura:** *plain structure*.

n = número de vértices

P[1..n] = vértices

R = retângulo envolvente global (xmin, xmax, ymin, ymax)

---

Note que não existe refinamento da poligonal, isto é, somente existe a região limitante na raiz, englobando toda a poligonal. Essa região permite descarte rápido de poligonais fora da região de interesse. Todas as outras representações estudadas contêm essa informação global.

---

**Algoritmo:** Construção da *plain structure*.

```
plain():  
    t.R = retângulo (p1, ..., pn)  
    return t
```

---

Novamente, a escolha dessa estrutura visa basicamente responder à seguinte pergunta: vale a pena implementar estruturas “pesadas” para se conseguir interatividade com bancos de dados cartográficos? Queremos avaliar se os ganhos de interatividade se devem principalmente ao descarte rápido.

### 3.6 BBox Tree - Uma Variante da Box Tree

Esta representação é a *box tree* com uma pequena modificação: a retirada dos pontos inicial e final do segmento. Estes pontos podem ser facilmente calculados no momento das operações. Esta tentativa foi feita para verificar se vale a pena reduzir o consumo de memória em detrimento de um potencial aumento de tempo.

---

**Estrutura:** *bbox tree*.

```
R = retângulo envolvente (xmin, xmax, ymin, ymax)
left = sub-árvore à esquerda
right = sub-árvore à direita
```

---

O processo de criação da *bbox tree* é igual ao da *box tree*, apenas não são mais armazenados os pontos do segmento principal da poligonal.

---

**Algoritmo:** Construção da *bbox tree*.

```
bbox(i, j):
    t.R = retângulo (pi, ..., pj)
    k = (i+j)/2
    t.left = bbox(i, k)
    t.right = bbox(k, j)
    return t
```

---

### 3.7 Comparações na Implementação das Representações

A *strip tree* mostrou-se a mais complexa para implementar, já que suas regiões limitantes são retângulos não necessariamente paralelos aos eixos XY. Como todas as

operações são em função da região limitante, esperávamos que fosse a pior nos testes, fato não confirmado. Apesar de em alguns testes não ser a mais rápida, ela é a que gera o menor número de segmentos, como veremos no Capítulo 4.

Um problema comum em representações em árvores é o desbalanceamento. Fato que não ocorre com a maioria das estruturas estudadas, já que a divisão é feita no número de pontos em  $n/2$ .

No caso da *strip tree* é alto o grau de desbalanceamento da árvore, já que a divisão dos nós é baseada na geometria da curva e não no número de elementos que ela possui.

No caso da *arc tree*, contornamos o problema dividindo pelo número de pontos em  $n/2$ . Conforme explicado na Seção 3.3, dividir em  $n/2$  é equivalente a dividir no ponto médio.

Para as demais representações e operações, não houve dificuldades em suas implementações.

# Capítulo 4

## Testes e Avaliações

Neste capítulo apresentamos as descrições dos bancos de dados utilizados nos teses e também os seus resultados nas diversas representações estudadas. Agrupamos os resultados em duas seções:

- Resultados por Operação: onde apresentamos os resultados dos diversos bancos de dados separados por operações; e
- Resultados por Banco de Dados: onde aparecem os resultados das diversas operações separados por banco de dados.

### 4.1 Descrição dos Bancos de Dados

Para os nossos objetivos, um *banco de dados cartográficos* é um conjunto não estruturado de linhas poligonais. Cada linha poligonal é uma sequência de pontos  $p_1, \dots, p_n$ , definidos por suas coordenadas cartesianas:  $p_i = (x_i, y_i)$ . Externamente, um conjunto de linhas poligonais é armazenado em um arquivo binário da forma mostrada na Figura 4.1. Por conveniência, um banco de dados pode estar organizado em mais de um arquivo deste tipo.

$n_1$	$p_{11}$	$p_{12}$	...	$p_{1n_1}$	$n_2$	...
-------	----------	----------	-----	------------	-------	-----

**Figura 4.1: Formato externo dos Bancos de Dados.**

Para uma avaliação precisa dos algoritmos, utilizamos dados reais gerados por aplicações ou disponíveis na Internet. Como exemplo, dados gerados pelo Censo Norte-Americano [15], dados geo-político do Mundo [14], o mapa do Brasil com divisões por estados e municípios [16], regiões de vegetação da Floresta Amazônica e Curvas de Níveis de Anchorage (Anchor) e Madison [17]. Os bancos de dados usados estão mostrados nas Figuras 4.2, 4.4, 4.6, 4.8, 4.10, 4.12 e 4.14.

Alguns desses bancos de dados não cabem em memória principal: dados geo-político do Mundo e dados da Floresta Amazônica. Para estes, foi feita uma análise por partes:

- **CIA World Data Bank:** Este banco de dados é dividido por continentes, adotamos assim a mesma divisão:
  - África;
  - América do Norte;
  - América do Sul;
  - Ásia; e
  - Europa;
  
- **Floresta Amazônica:** Este banco de dados é composto por 26 arquivos. A divisão foi feita baseada no número de arquivos e na possibilidade do carregamento para a memória principal.
  - Floresta Amazônica I: arquivos 01 a 09;
  - Floresta Amazônica II: arquivos 10 a 19;
  - Floresta Amazônica III: arquivos 20 a 26;

Para estes dois bancos de dados escolhemos um módulo de cada: o continente Africano do CIA World Data Bank e a Floresta Amazônica I.

Os testes visam medir os tempos gastos pelos procedimentos de desenho (*draw*), seleção (*pick*), interseção e cerceamento (*clipping*). Para o teste de desenho, contabiliza-se além do tempo, o número de segmentos desenhados.

Foram medidos o *overhead* de construção para cada algoritmo, isto é, tempo de construção das estruturas de dados e quantidade de memória utilizada.

A Tabela 4.1 mostra uma visão geral das principais características dos bancos de dados utilizados para os testes.

<b>Dados</b>	<b>Curvas</b>	<b>Segmentos</b>	<b>Bytes</b>
<b>CIA World Data Bank (global)</b>	31.845	5.654.624	45.619.132
<b>Floresta Amazônica (global)</b>	19.956	2.347.643	19.020.616
<b>Municípios do Brasil</b>	16.596	700.337	5.801.848
<b>Curvas de Nível (Madison)</b>	6.176	662.834	5.376.784
<b>Curvas de Nível (Anchor)</b>	4.247	220.969	1.818.716
<b>Municípios dos Estados Unidos</b>	4.085	164.555	1.365.460
<b>Estados do Brasil</b>	173	20.955	169.716

**Tabela 4.1: Principais características dos bancos de dados utilizados nos testes.**

A Tabela 4.2 mostra um fato importante: as médias de pontos são muito baixas em relação ao número de linhas. Em outras palavras, os bancos de dados contêm muitas linhas com poucos pontos. Este fato pode ser observado para os bancos de dados dos Municípios do Brasil e Municípios dos Estados Unidos.

Dados	Mínimo	Máximo	$\omega$	$\sigma$
<b>CIA World Data Bank (global)</b>	2	9744	178	367,2
<b>Floresta Amazônica (global)</b>	4	5431	118	165,1
<b>Municípios do Brasil</b>	2	434	43	46,5
<b>Curvas de Nível (Madison)</b>	2	2534	108	293,1
<b>Curvas de Nível (Anchor)</b>	2	2710	53	184,3
<b>Municípios dos Estados Unidos</b>	2	866	41	76,7
<b>Estados do Brasil</b>	2	1293	122	233,9

$\omega$  = Média de pontos por poligonal

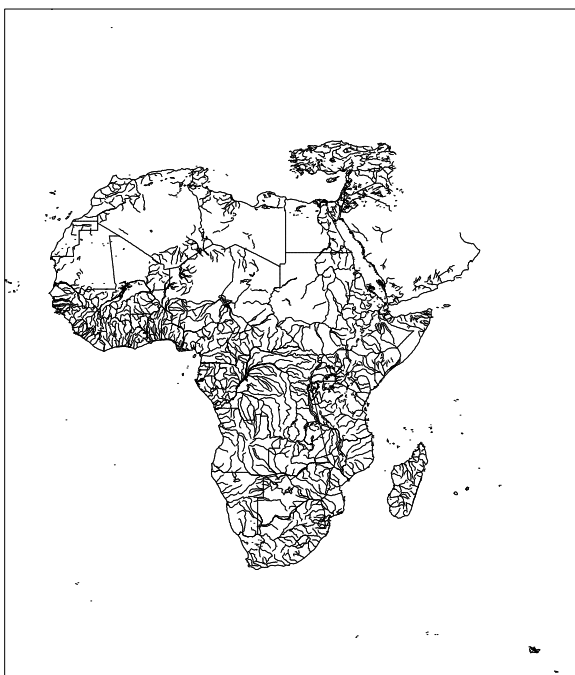
$\sigma$  = Desvio padrão dos pontos

**Tabela 4.2: Médias, variâncias, máximos e mínimos dos números de pontos por poligonal dos bancos de dados.**

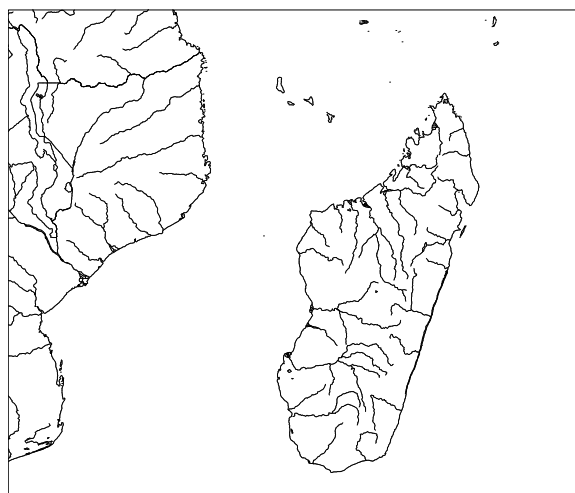
Para uma visão mais clara deste fato, são apresentados os histogramas dos diversos bancos de dados, que mostram as distribuições dos números de pontos por linhas (Figuras 4.3, 4.5, 4.7, 4.9, 4.11, 4.13 e 4.15).

Os diversos histogramas confirmam tal fato: muitas linhas com poucos pontos. Esta característica é muito importante para as representações em multi-resolução, já que estas foram propriamente projetadas para linhas com muitos pontos. O que não invalida o seu uso.

Nas seções seguintes são mostradas as tabelas de comparações entre as diversas representações. As tabelas estão separadas em duas seções: comparação por operação e comparação por banco de dados, proporcionando assim, uma visão global do desempenho das estruturas. Nas tabelas de resultados dos testes, os valores em negrito apontam o melhor desempenho.

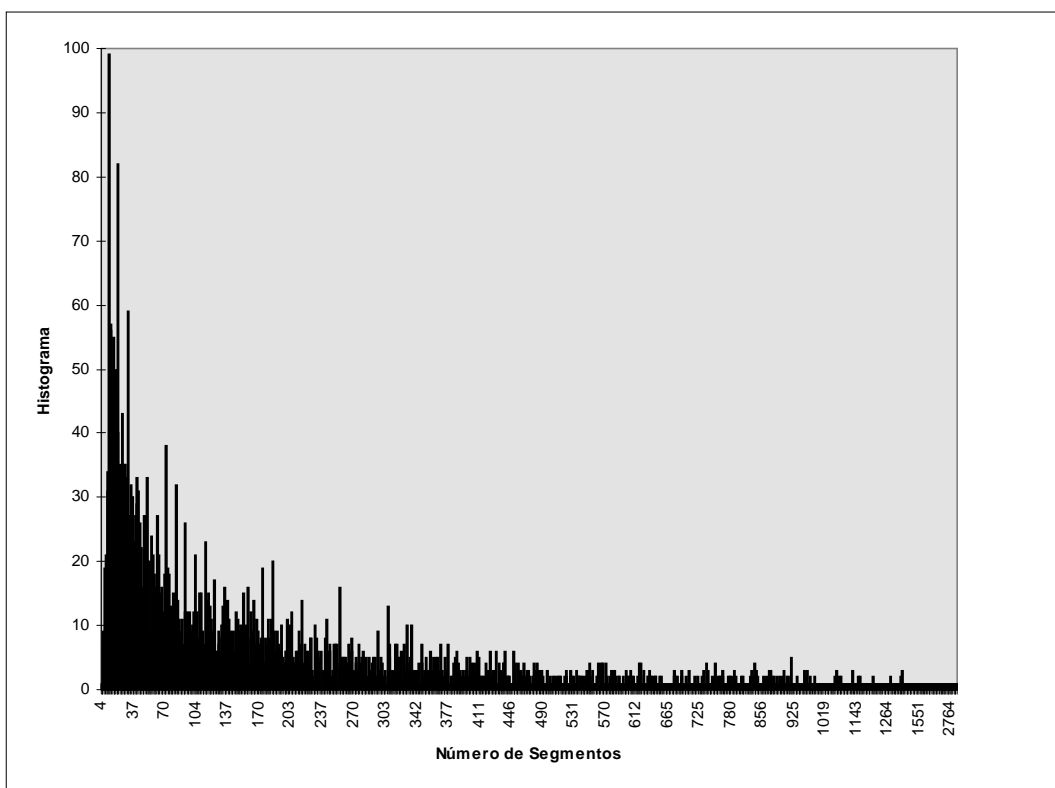


Mapa da África. Resolução de 5 pixels.



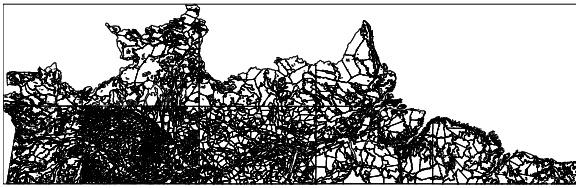
Detalhe da Ilha de Madagascar. Resolução de 5 pixels.

**Figura 4.2: Banco de dados do Continente Africano.**

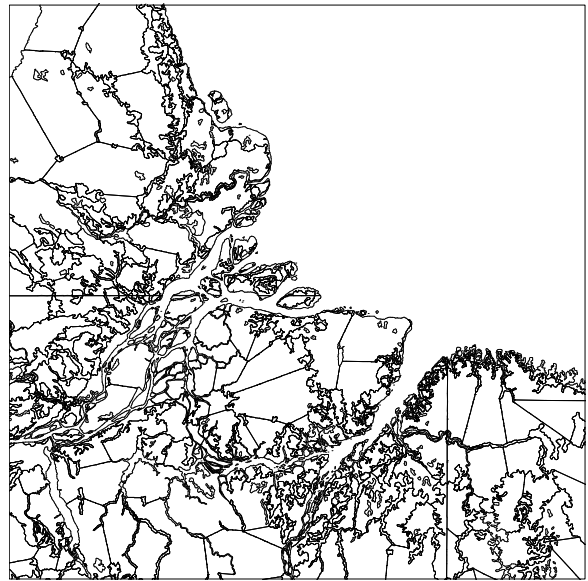


**Figura 4.3: Histograma da distribuição do número de pontos por poligonal da CIA World Data Bank (África).**



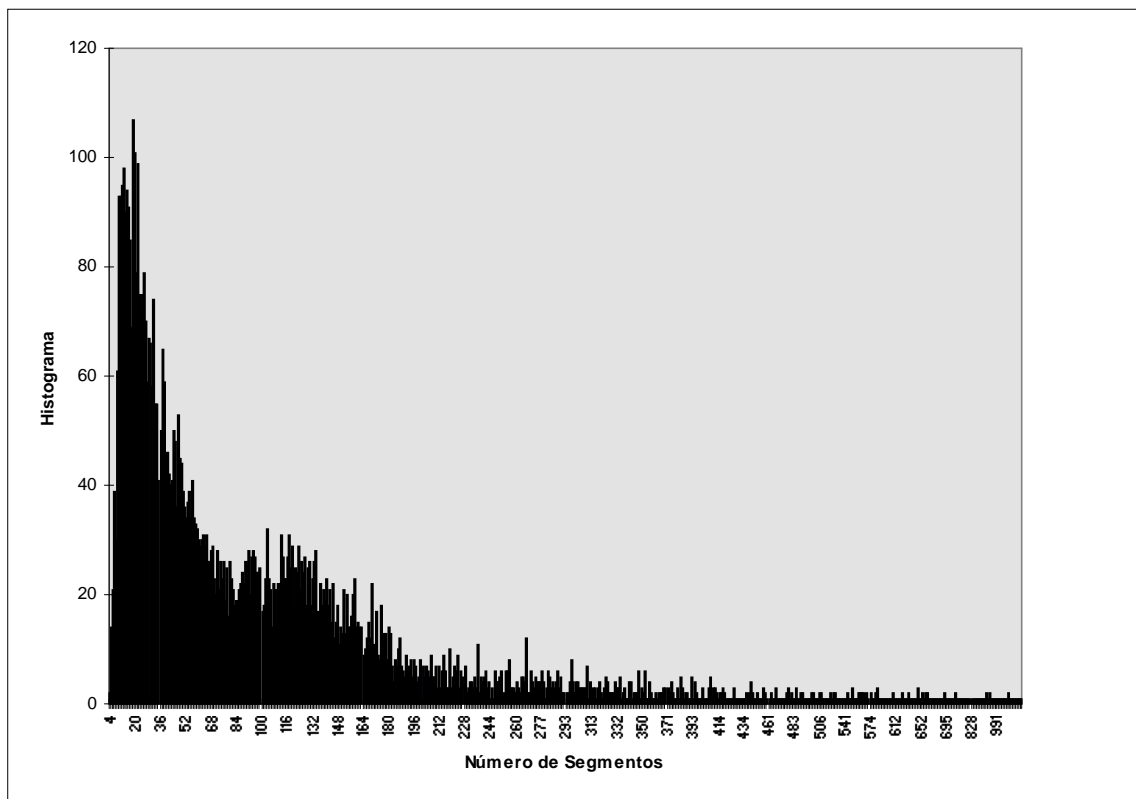


Floresta Amazônica. Resolução de 5 pixels.

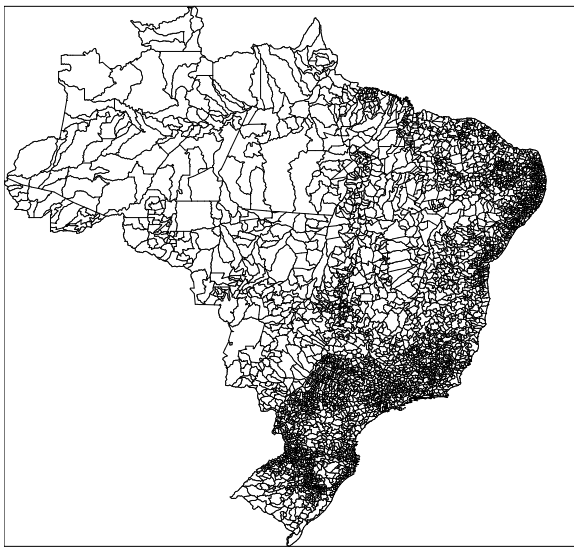


Detalhe da Ilha de Marajó. Resolução de 5 pixels.

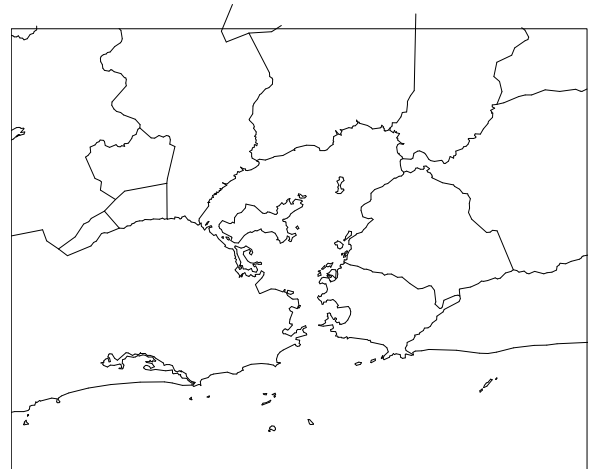
**Figura 4.4: Banco de dados da Floresta Amazônica I.**



**Figura 4.5: Histograma da distribuição do número de pontos por poligonal do banco de dados da Floresta Amazônica I.**

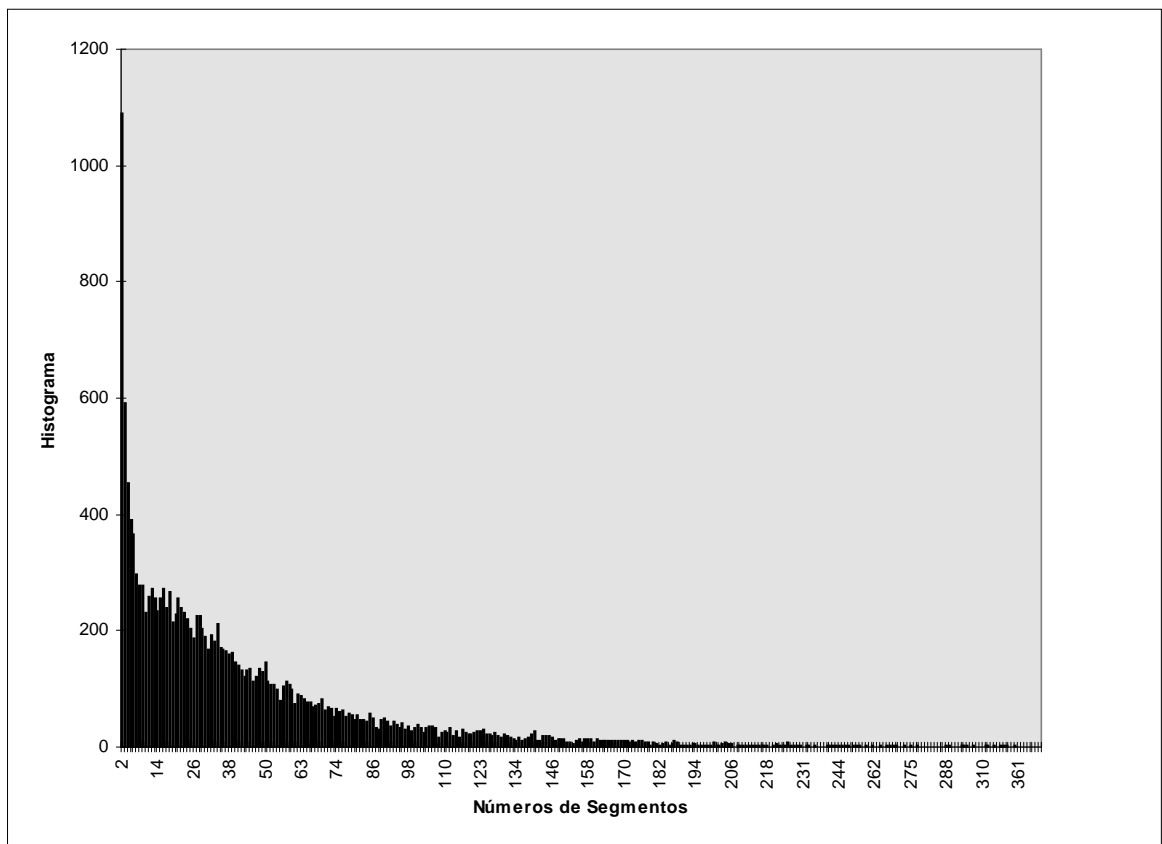


Mapa do Brasil com Municípios. Resolução de 5 *pixels*.

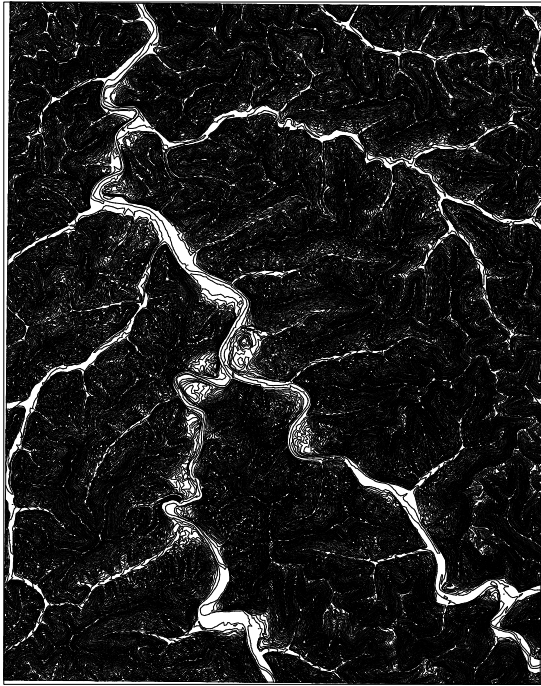


Detalhe da Baía de Guanabara. Resolução de 5 *pixels*.

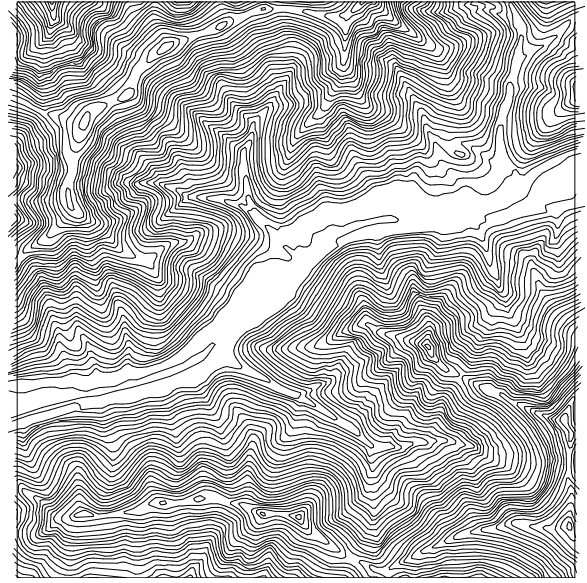
**Figura 4.6: Banco de dados dos Municípios do Brasil.**



**Figura 4.7: Histograma da distribuição do número de pontos por poligonal dos Municípios do Brasil.**

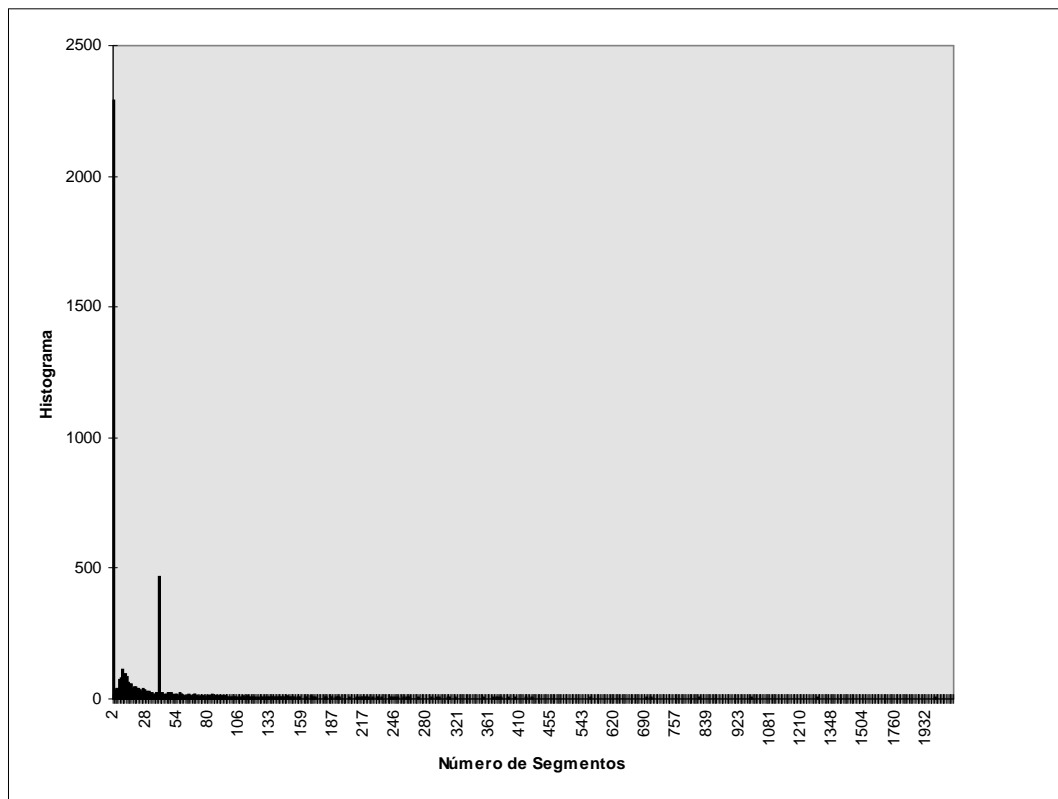


Mapa das Curvas de Nível. Resolução de 5 pixels.



Detalhe. Resolução de 5 pixels.

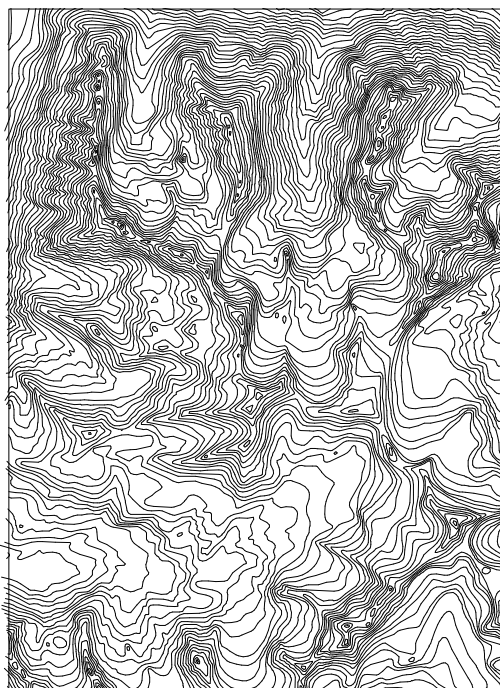
**Figura 4.8: Banco de dados de Curvas de Nível de Madison.**



**Figura 4.9: Histograma da distribuição do número de pontos por poligonal das Curvas de Nível de Madison.**

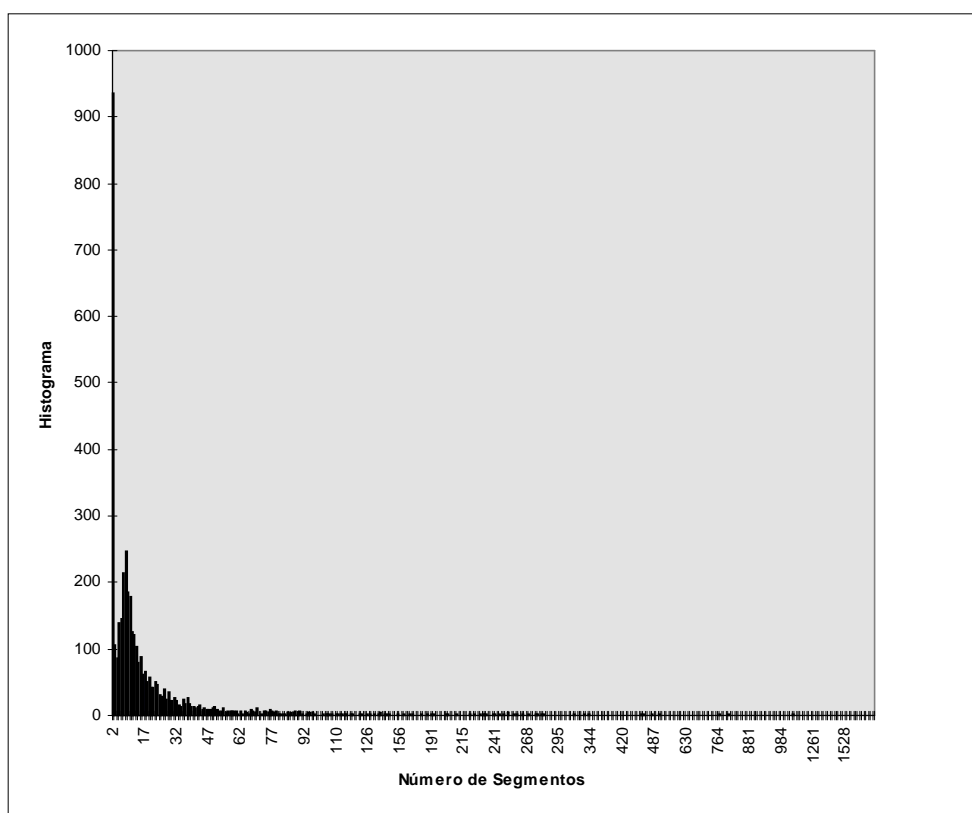


Mapa das Curvas de Nível. Resolução de 5 pixels.

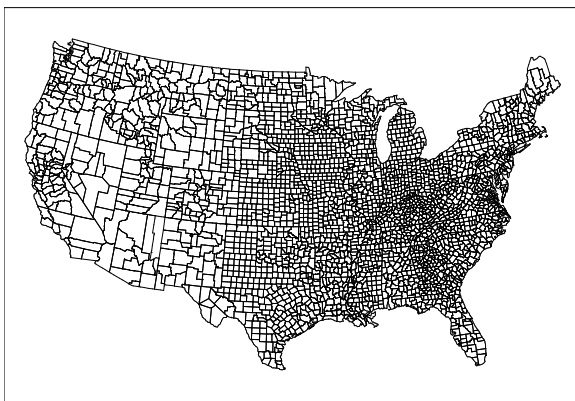


Detalhe. Resolução de 5 pixels.

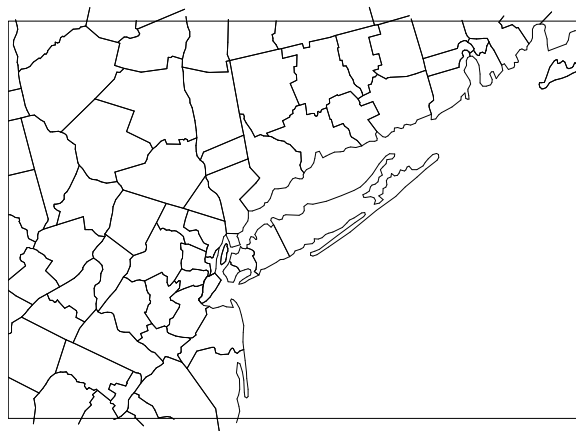
**Figura 4.10: Banco de dados de Curvas de Nível de Anchor.**



**Figura 4.11: Histograma da distribuição do número de pontos por poligonal das Curvas de Nível de Anchor.**

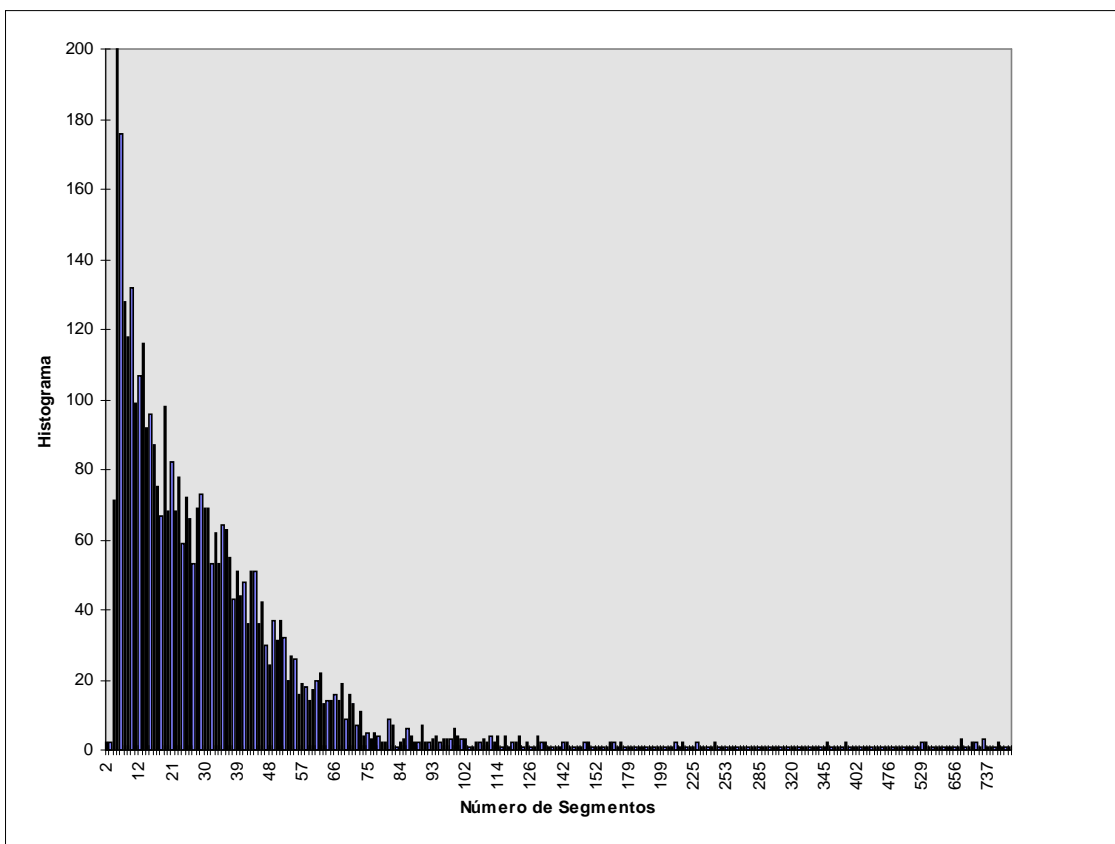


Mapa dos Estados Unidos com Municípios.  
Resolução de 5 *pixels*.

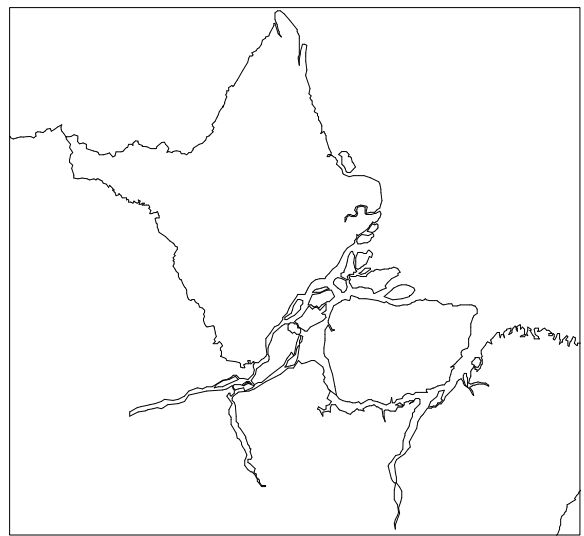


Detalhe da perto de Manhattan e Long Island.  
Resolução de 5 *pixels*.

**Figura 4.12: Banco de dados dos Municípios dos Estados Unidos.**



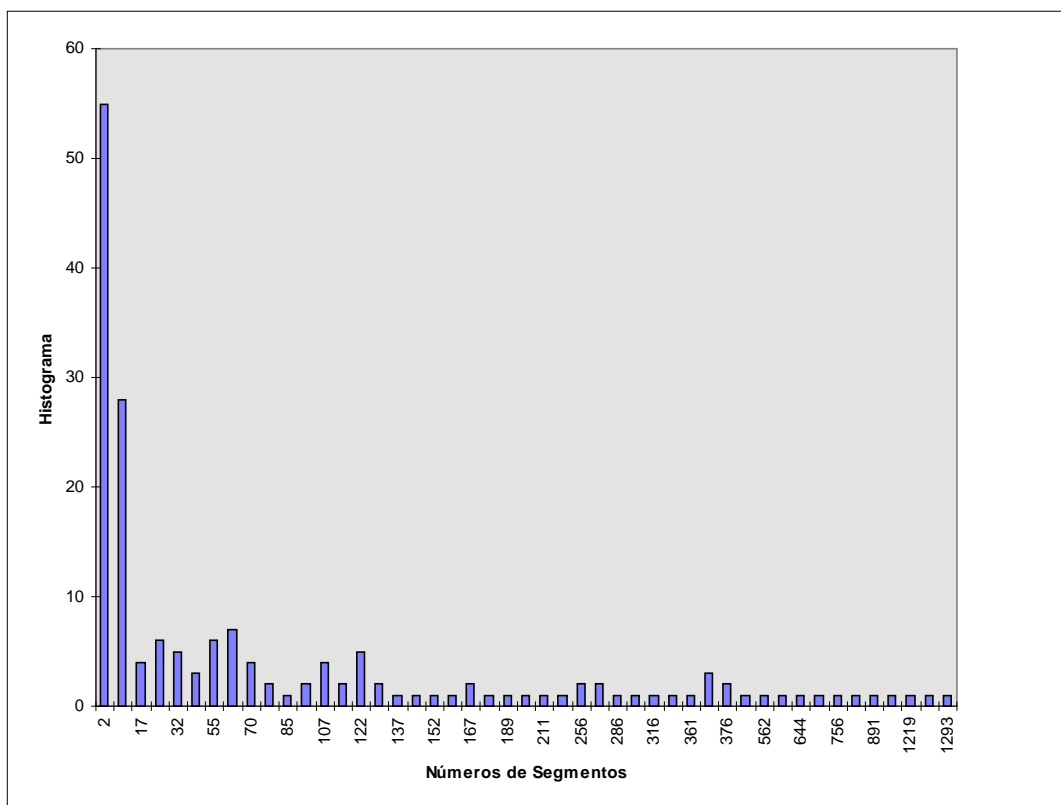
**Figura 4.13: Histograma da distribuição do número de pontos por poligonal dos Municípios dos Estados Unidos.**



Mapa dos Estados do Brasil. Resolução de 10 *pixels*.

Detalhe da Ilha de Marajó. Resolução de 10 *pixels*.

**Figura 4.14: Banco de dados dos Estados do Brasil.**



**Figura 4.15: Histograma da distribuição do número de pontos por poligonal dos Estados do Brasil.**

## 4.2 Resultados por Operação

### 4.2.1 Criação

O tempo contabilizado se refere à criação em si das estruturas, não envolvendo a leitura (carga dos dados para a memória principal) dos bancos de dados.

A Tabela 4.3 mostra que a *strip tree* é a mais demorada na criação. Isto se deve ao fato de que o cálculo das regiões limitantes é mais complexo que as outras. Como não poderia deixar de ser, tanto a *dumb* quanto a *plain* são as mais rápidas, já que não são necessários cálculos para sua construção.

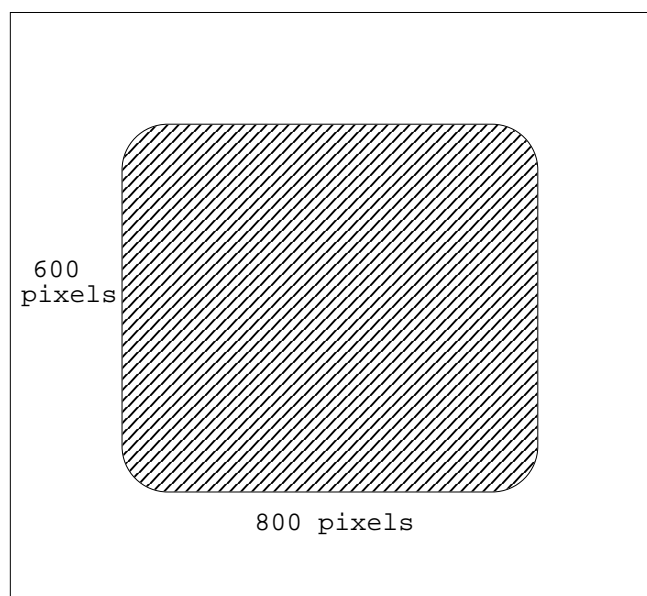
Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Estados do Brasil	<b>0,03</b>	0,04	0,16	0,16	0,29	0,70
Municípios do Brasil	<b>1,68</b>	1,94	6,84	6,57	10,05	20,63
Municípios dos Estados Unidos	<b>0,38</b>	0,45	1,73	1,61	2,40	4,98
CIA World Data Bank (África)	<b>1,25</b>	1,35	8,84	8,45	12,97	32,63
Curvas de Nível (Anchor)	0,53	<b>0,51</b>	2,16	1,99	3,00	7,47
Curvas de Nível (Madison)	<b>1,02</b>	1,17	5,89	5,41	8,65	21,66
Floresta Amazônica I	1,31	<b>1,17</b>	6,53	5,96	9,84	24,45

Tabela 4.3: Tempos em segundos para a operação de criação.

### 4.2.2 Desenho

O desenho é o percorrimento de todas as folhas, isto é, com resolução 0. A área de visualização dos desenhos foi fixada em 800×600 *pixels*.

Este teste serve para comparar as velocidades de percorrimento de todas as estruturas sem levar em consideração a multi-resolução.



**Figura 4.16: Definição do tamanho da área de desenho.**

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Estados do Brasil	0,14	<b>0,13</b>	0,43	0,42	0,44	0,88
Municípios do Brasil	<b>3,86</b>	4,10	14,56	14,70	14,35	31,04
Municípios dos Estados Unidos	<b>0,91</b>	0,98	3,41	3,39	2,40	7,22
CIA World Data Bank (África)	<b>5,52</b>	5,61	21,11	21,39	20,74	42,98
Curvas de Nível (Anchor)	1,14	<b>1,22</b>	4,61	4,74	4,46	10,05
Curvas de Nível (Madison)	<b>3,53</b>	3,69	13,86	14,21	13,28	5,16
Floresta Amazônica I	<b>4,09</b>	<b>4,09</b>	15,41	15,86	15,08	32,16

**Tabela 4.4: Tempos em segundos para a operação de desenho sem tolerância.**

Como pode-se observar na Tabela 3.4, as estruturas *plain* e *dumb* são as mais rápidas, já que não precisam percorrer nenhuma estrutura, pois já armazenam as próprias linhas.

A *strip tree*, por outro lado, leva mais tempo para o percorrimento da árvore, isto se deve ao fato dela ser mais precisa em relação as outras (*arc tree* e *box tree*), gerando árvores com maiores profundidades.



### 4.2.3 Desenho com Tolerância

Neste teste, utilizamos uma resolução de 1 *pixel* para o desenho.

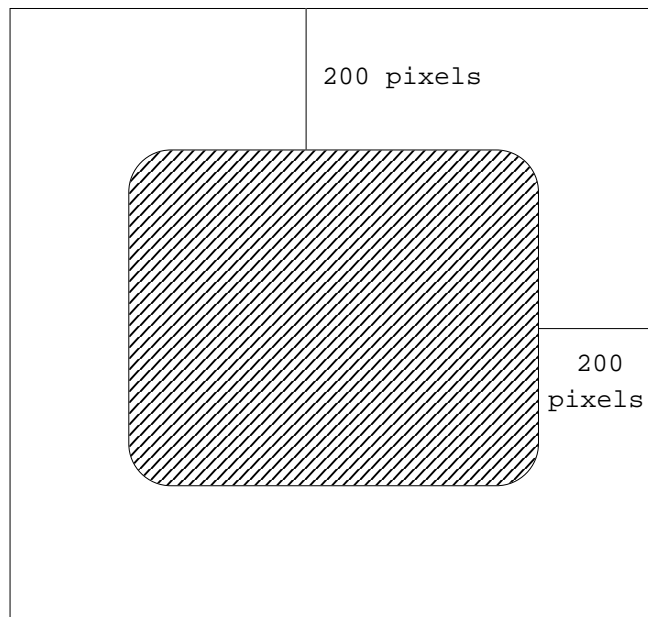
A vantagem da utilização de uma representação em multi-resolução já pode ser notada na Tabela 4.5. Os tempos chegam a ser 15 vezes inferiores que as estruturas *dumb* e *plain* as quais não utilizam o recurso da multi-resolução. Um bom exemplo é o banco de dados da CIA, onde na estrutura *dumb* levou 5,55 segundos e na *strip tree* 0,36 segundos.

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Estados do Brasil	<b>0,12</b>	0,14	0,22	0,25	0,18	0,13
Municípios do Brasil	3,89	4,07	1,98	2,02	1,44	<b>1,21</b>
Municípios dos Estados Unidos	0,95	0,92	1,28	1,25	1,02	<b>0,91</b>
CIA World Data Bank (África)	5,55	5,60	0,87	1,29	0,58	<b>0,36</b>
Curvas de Nível (Anchor)	<b>1,29</b>	1,30	4,31	4,64	3,72	1,97
Curvas de Nível (Madison)	<b>3,53</b>	3,69	12,84	12,46	9,89	5,16
Floresta Amazônica I	4,19	4,33	5,37	5,11	3,29	<b>2,62</b>

Tabela 4.5: Tempos em segundos para a operação de desenho com tolerância de 1 *pixel*.

### 4.2.4 Desenho com Tolerância e Cerceamento

Na operação de cerceamento, foi retirada uma faixa de 200 *pixels* de cada borda do desenho e usada uma resolução de 1 *pixel*, como mostra a Figura 4.17.



**Figura 4.17: Diminuição da área de desenho em 200 pixels de cada borda.**

Novamente, é notada a utilização da representação em multi-resolução (Tabela 4.6). A redução dos tempos chega a aproximadamente 10 vezes inferior que as estruturas *dumb* e *plain*, como é o caso do banco de dados da CIA: 0,48 segundos na *strip tree* e 5,55 na estrutura *dumb*.

Tempo (s)	Dumb	Plain	Bbox	Box	Arc	Strip
Estados do Brasil	0,12	<b>0,04</b>	0,13	0,12	0,07	0,07
Municípios do Brasil	3,92	1,86	1,90	1,76	1,16	<b>0,75</b>
Municípios dos Estados Unidos	1,00	<b>0,14</b>	0,38	0,35	0,33	0,33
CIA World Data Bank (África)	5,55	4,28	1,38	1,45	0,87	<b>0,48</b>
Curvas de Nível (Anchor)	1,23	<b>0,83</b>	1,62	1,60	1,65	1,11
Curvas de Nível (Madison)	3,64	<b>2,46</b>	4,17	4,13	4,05	2,51
Floresta Amazônica I	4,11	1,64	3,65	3,54	2,42	<b>1,63</b>

Tabela 4.6: Tempos em segundos para a operação de desenho com tolerância de 1 *pixel* e cercejamento.

#### 4.2.5 Interseção

A escolha da curva para o cálculo de interseção foi feita para atender a todos os bancos de dados, isto é, que force pelo menos um ponto de interseção. Esta curva compreende uma linha poligonal com apenas 4 segmentos, mas que percorre toda a extensão do desenho.

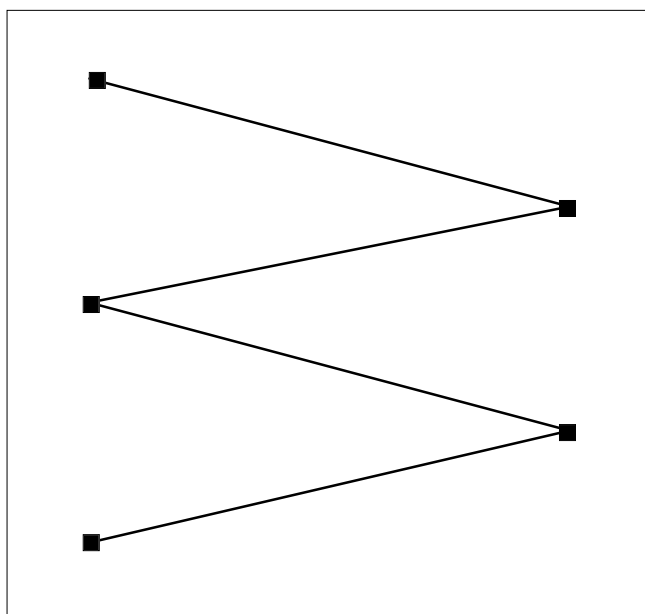


Figura 4.18: Linha poligonal englobando toda a região do desenho para o cálculo da interseção.

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Estados do Brasil	0,31	0,32	<b>0,15</b>	0,17	0,59	0,27
Municípios do Brasil	10,34	10,51	<b>5,11</b>	5,65	20,05	16,36
Municípios dos Estados Unidos	2,43	2,44	<b>1,21</b>	1,35	5,06	4,76
CIA World Data Bank (África)	15,07	15,17	7,27	8,06	29,01	<b>6,26</b>
Curvas de Nível (Anchor)	3,23	3,28	<b>1,61</b>	1,78	4,77	5,16
Curvas de Nível (Madison)	10,04	10,13	<b>5,01</b>	5,48	15,62	8,21
Floresta Amazônica I	10,76	10,74	<b>5,35</b>	5,92	23,26	6,93

**Tabela 4.7: Tempos em segundos para a operação de interseção.**

#### 4.2.6 Seleção

O ponto escolhido para seleção foi o centro do desenho.

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Estados do Brasil	<b>0,10</b>	0,07	0,01	0,01	0,01	0,01
Municípios do Brasil	<b>2,24</b>	0,04	0,03	0,03	0,03	0,03
Municípios dos Estados Unidos	<b>0,51</b>	0,01	0,01	0,01	0,01	0,01
CIA World Data Bank (África)	<b>3,26</b>	0,03	0,01	0,01	0,01	0,01
Curvas de Nível (Anchor)	<b>0,72</b>	0,09	0,01	0,01	0,01	0,01
Curvas de Nível (Madison)	<b>2,11</b>	0,23	0,01	0,02	0,01	0,02
Floresta Amazônica I	<b>2,18</b>	0,03	0,01	0,01	0,01	0,02

**Tabela 4.8: Tempos em segundos para a operação de seleção.**

Na Tabela 4.8, colocamos em evidência os piores tempos. Aqui podemos perceber a vantagem de se ter uma representação que faça algum tipo de descarte. Por menor que seja este descarte, como exemplo a *plain*, o tempo chega a ser reduzido em até 56 vezes, é o caso do banco de dados dos Municípios do Brasil: 2,24 segundos para a *dumb* enquanto que utiliza 0,04 segundos para a *plain*.

### 4.3 Resultados por Banco de Dados

As seções a seguir mostram as comparações das operações do ponto de vista dos bancos de dados, proporcionando uma visão global do desempenho das representações em multi-resolução.

O valor da redução, mostrado nas tabelas a seguir, se refere a redução do número de segmentos desenhados nas operações de desenho quando a resolução é maior que 0. O valor é expresso em % e formado pela divisão do número de segmentos desenhados pela representação dividido pelo número total de segmentos do desenho. Por esta razão que na estrutura *plain* o valor da redução é sempre 0.

#### 4.3.1 Estados do Brasil

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	<b>0,03</b>	0,04	0,16	0,16	0,29	0,70
Desenho	0,14	<b>0,13</b>	0,43	0,42	0,44	0,88
Desenho c/ tolerância	<b>0,12</b>	0,14	0,22	0,25	0,18	0,13
% redução	—	—	44	44	59	<b>88</b>
Des. c/ tolerância e clip	0,12	<b>0,04</b>	0,12	0,13	0,07	0,07
% redução	—	41	75	75	81	<b>94</b>
Interseção	0,31	0,32	<b>0,15</b>	0,17	0,59	0,27
Seleção	0,10	0,07	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>
Memória Total (Mb)	<b>0,01</b>	<b>0,01</b>	1,28	0,96	1,12	0,91

**Tabela 4.9: Comparação das estruturas para o banco de dados dos Estados do Brasil.**

Pontos de interseção: 74. Entidades selecionadas: 3.

### 4.3.2 Municípios do Brasil

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	<b>1,68</b>	1,94	6,84	6,57	10,05	20,63
Desenho	<b>3,86</b>	4,10	14,56	14,70	14,35	31,04
Desenho c/ tolerância	3,89	4,07	1,98	2,02	1,44	<b>1,21</b>
% redução	—	—	86	86	90	<b>96</b>
Des. c/ tolerância e clip	3,92	1,86	1,76	1,90	1,16	<b>0,75</b>
% redução	—	53	88	88	92	<b>98</b>
Interseção	10,34	10,51	<b>5,11</b>	5,65	20,05	16,36
Seleção	2,24	0,04	<b>0,03</b>	<b>0,03</b>	<b>0,03</b>	<b>0,03</b>
Memoria Total (Mb)	<b>0,51</b>	<b>0,51</b>	42,75	32,19	37,47	32,19

Tabela 4.10: Comparação das estruturas para o banco de dados dos Municípios do Brasil.

Na operação de interseção foram encontrados 372 pontos e na seleção 5 entidades.

### 4.3.3 Municípios dos Estados Unidos

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	<b>0,38</b>	0,45	1,73	1,61	2,40	4,98
Desenho	<b>0,91</b>	0,98	3,41	3,39	2,40	7,22
Desenho c/ tolerância	0,95	0,92	1,28	1,25	1,02	<b>0,91</b>
% redução	—	—	62	62	70	<b>87</b>
Des. c/ tolerância e clip	1,00	<b>0,14</b>	0,35	0,38	0,33	0,33
% redução	—	89	90	90	90	<b>97</b>
Interseção	2,43	2,44	<b>1,21</b>	1,35	5,06	4,76
Seleção	0,51	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>
Memoria Total (Mb)	<b>0,12</b>	<b>0,12</b>	10,04	7,56	8,80	7,47

Tabela 4.11: Comparação das estruturas para o banco de dados dos Municípios dos Estados Unidos.

Pontos de interseção: 542. Entidades selecionadas: 3.

Note que na Figura 3.12, alguns segmentos *sairam* do retângulo de visualização. Como foi dito na Seção 2.1.4, o *clipping* é feito pelo sistema gráfico e não pelas representações. Elas apenas fazem o descarte dos segmentos que realmente *não* são visíveis.

#### 4.3.4 CIA World Data Bank (África)

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	<b>1,25</b>	1,35	8,84	8,45	12,97	32,63
Desenho	<b>5,52</b>	5,61	21,11	21,39	20,74	42,98
Desenho c/ tolerância	5,55	5,60	0,87	1,29	0,58	<b>0,36</b>
% redução	—	—	96	96	97	<b>99</b>
Des. c/ tolerância e clip	5,55	4,28	1,45	1,38	0,87	<b>0,48</b>
% redução	—	25	94	94	96	<b>99</b>
Interseção	15,07	15,17	7,27	8,06	29,01	<b>6,26</b>
Seleção	3,26	0,03	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>
Memoria Total (Mb)	<b>0,14</b>	<b>0,14</b>	62,23	46,70	54,46	43,07

Tabela 4.12: Comparação das estruturas para o banco de dados do CIA World Data Bank, Continente Africano.

Pontos de interseção: 137. Entidades selecionadas: 4.

#### 4.3.5 Curvas de Nível de Anchor

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	0,53	<b>0,51</b>	2,16	1,99	3,00	7,47
Desenho	1,14	<b>1,22</b>	4,61	4,74	4,46	10,05
Desenho c/ tolerância	<b>1,29</b>	1,30	4,31	4,64	3,72	1,97
% redução	—	—	5	5	18	<b>81</b>
Des. c/ tolerância e clip	1,23	<b>0,83</b>	1,60	1,62	1,65	1,11
% redução	—	32	65	65	65	<b>89</b>
Interseção	3,23	3,28	<b>1,61</b>	1,78	4,77	5,16
Seleção	0,72	0,09	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>
Memoria Total (Mb)	<b>0,13</b>	<b>0,13</b>	13,49	10,15	11,82	10,14

Tabela 4.13: Comparação das estruturas para o banco de dados das Curvas de Nível de Anchor.

Pontos de interseção: 991. Entidades selecionadas: 5.

#### 4.3.6 Curvas de Nível de Madison

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	<b>1,02</b>	1,17	5,89	5,41	8,65	21,66
Desenho	<b>3,77</b>	3,82	13,86	14,21	13,28	30,12
Desenho c/ tolerância	<b>3,53</b>	3,69	12,84	12,46	9,89	5,16
% redução	—	—	9	9	27	<b>83</b>
Des. c/ tolerância e clip	3,64	<b>2,46</b>	4,13	4,17	4,05	2,51
% redução	—	35	70	70	71	<b>92</b>
Interseção	10,04	10,13	<b>5,01</b>	5,48	15,62	8,21
Seleção	2,11	0,23	<b>0,01</b>	0,02	<b>0,01</b>	0,02
Memoria Total (Mb)	<b>0,19</b>	<b>0,19</b>	40,46	30,39	35,42	30,17

Tabela 4.14: Comparação das estruturas para o banco de dados das Curvas de Nível de Madison.

Pontos de interseção: 2.147. Entidades selecionadas: 20.

#### 4.3.7 Floresta Amazônica I

Tempo (s)	Dumb	Plain	Box	Bbox	Arc	Strip
Criação	1,31	<b>1,17</b>	6,53	5,96	9,84	24,45
Desenho	<b>4,09</b>	<b>4,09</b>	15,41	15,86	15,08	32,16
Desenho c/ tolerância	4,19	4,33	5,37	5,11	3,29	<b>2,62</b>
% redução	—	—	66	66	79	<b>92</b>
Des. c/ tolerância e clip	4,11	1,64	3,54	3,65	2,42	<b>1,63</b>
% redução	—	62	77	77	84	<b>95</b>
Interseção	10,76	10,74	<b>5,35</b>	5,92	23,26	6,93
Seleção	2,18	0,03	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>	0,02
Memoria Total (Mb)	<b>0,21</b>	<b>0,21</b>	45,23	33,97	39,60	32,62

Tabela 4.15: Comparação das estruturas para o banco de dados da Floresta Amazônica I.

Pontos de interseção: 1454. Entidades selecionadas: 7.



## 4.4 Conclusões Preliminares

Após uma análise das tabelas, chegamos aos seguintes resultados:

- A *strip tree* é a mais rápida, exceto na operação de interseção. Apesar de mais precisa que a *box tree*, a profundidade das árvores é maior.
- A *strip tree* é a que gera menor número de segmentos; apesar de levar mais tempo para a geração dos desenhos, tem maior precisão.
- A *box tree* é a que ocupa maior espaço em memória. Isto é devido as informações que a estrutura armazena: além da região limitante, os vértices visitados. Este armazenamento extra dos vértices acaba sendo significativo quanto ao espaço em memória. Em contra-partida, a *strip tree*, dentre as representações de multi-resolução é a que ocupa menos memória juntamente com a *bbox tree* (Tabela 4.16).

Representação	Tamanho em <i>bytes</i>
Arc	28
Bbox	24
Box	32
Dumb	8
Plain	8
Strip	24

**Tabela 4.16: Comparação do tamanho do nós de cada representação.**

# Capítulo 5

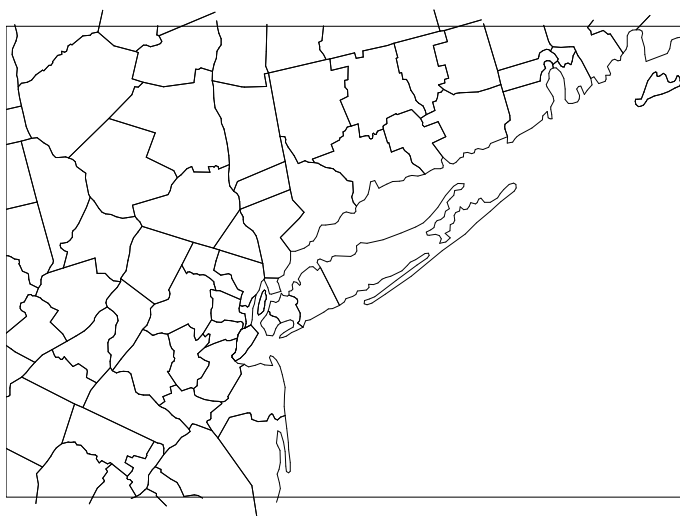
## Conclusões

A principal conclusão deste estudo é que é essencial a utilização qualquer uma das estruturas mencionadas, na representação de grandes bancos de dados cartográficos, para que se obtenha interatividade na exploração. A interatividade é possível graças ao descarte rápido que as estruturas implementam. Este descarte é o principal ponto das representações em multi-resolução.

### 5.1 Bancos de Dados

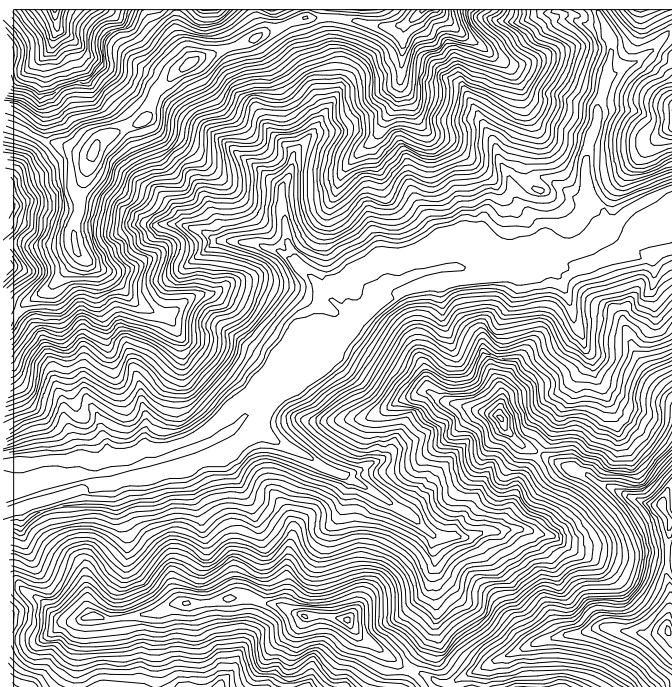
Apesar de não ser o objetivo principal deste estudo, ao analisarmos os resultados demonstrados no Capítulo 4, conseguimos caracterizar alguns tipos de bancos de dados: mapas políticos, curvas de nível e mapas temáticos.

- **Mapas Políticos:** Bancos de dados representados por sub-divisões planares, isto é, regiões bem definidas, onde as interseções entre as linhas poligonais, quando ocorrem, estão localizadas em suas extremidades (ponto inicial ou ponto final). Como exemplo, podemos citar o bando de dados dos Estados do Brasil, Municípios do Brasil, Municípios do Estados Unidos (Figura 5.1).



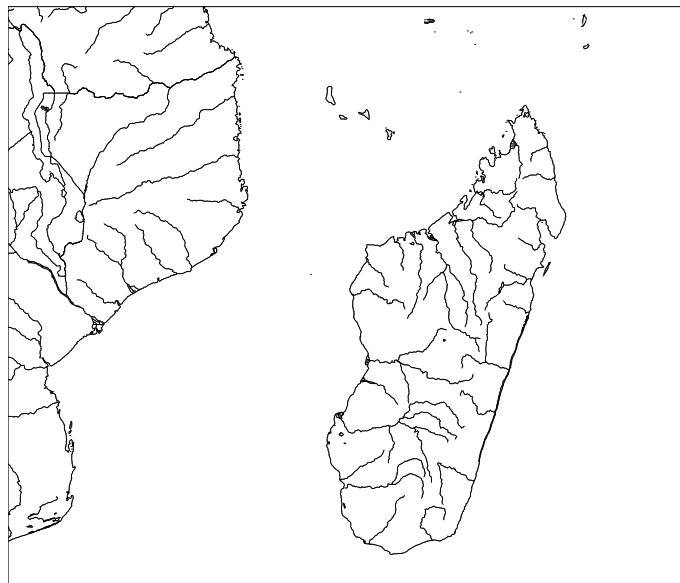
**Figura 5.1: Exemplo de Mapa Político: banco de dados dos Municípios dos Estados Unidos.**

- **Curvas de Nível:** Bancos de dados constituído de poligonais longas, que se estendem por grande parte do desenho. Por serem curvas de nível, não existem interseções entre as linhas poligonais. Como exemplo, as curvas de nível de Anchor e Madison (Figura 5.2).



**Figura 5.2: Exemplo de Curvas de Nível: banco de dados de Madison.**

- **Mapas Temáticos:** Bancos de dados constituídos de linhas poligonais dos mais variados tamanhos. É uma mistura de mapas políticos e curvas de nível, onde se apresentam tanto as divisões por regiões quanto acidentes geográficos (rios, montanhas). Pode existir um grande número de interseções entre as linhas. Como exemplo: Vegetação Amazônica e CIA World Data Bank (Figura 5.3).



**Figura 5.3: Exemplo de Mapas Temáticos, banco de dados de CIA World Data Bank, Continente Africano.**

Baseados nesta caracterização dos bancos de dados, concluímos a Tabela 5.1, onde apresentamos as melhores representações para cada tipo de dado.

Mapas Políticos	Curvas de Nível
<ul style="list-style-type: none"> <li>• Strip</li> <li>• Plain</li> <li>• Box</li> </ul>	<ul style="list-style-type: none"> <li>• Plain</li> <li>• Strip</li> </ul>
Mapas Temáticos	Impressão
<ul style="list-style-type: none"> <li>• Strip</li> <li>• Arc</li> </ul>	<ul style="list-style-type: none"> <li>• Strip</li> <li>• Arc</li> </ul>

**Tabela 5.1: Resultado das melhores representações separados por tipos de banco de dados.**

Vários outros resultados foram obtidos a partir dos estudos dos bancos de dados e suas representações em multi-resolução:

- Bancos de dados reais contêm uma variedade de linhas poligonais, algumas *longas*<sup>3</sup> (número de pontos na faixa dos milhares), algumas muito curtas (número de pontos na faixa das dezenas).
- Bancos de dados reais contêm muitos segmentos, mas não necessariamente linhas muito longas. Frequentemente, existem muitas linhas curtas, como pode-se observar nas médias de pontos por linha na Tabela 4.2. As médias dos números de segmentos por poligonal são bem baixas em relação ao número total de linhas.

## 5.2 Considerações Gerais

Com relação às representações em multi-resolução, observamos o seguinte:

- Usar uma representação em multi-resolução não é suficiente; é necessário usar estruturas de busca eficientes como *R-tree* [7, 8, 11]. Os tempos, principalmente, de localização

---

<sup>3</sup> O significado de “longa” é fazendo referência ao número de pontos, isto é, uma linha *longa* tem seu número de pontos na faixa dos milhares (dezenas, centenas) ou mais pontos.

especial de uma linha podem ser reduzidos. Neste estudo sempre são percorridas todas as linhas para qualquer uma das operações estudadas.

- Manter todos os dados em memória não é conveniente. Talvez seja suficiente manter os índices em memória mas a geometria em disco, principalmente se usarmos um mecanismo de *cache* eficiente.

### 5.3 Trabalhos Futuros

Como sugestões para futuros trabalhos, podemos citar:

- Implementação da representação de *Bézier*: A implementação de mais uma representação em multi-resolução pode aprimorar o estudo, neste caso a utilização de uma curva *Bézier* na substituição dos segmentos intermediários. O ponto é que uma curva de *Bézier* pode representar uma linha poligonal que tenha 1.000 pontos com talvez apenas 100 pontos de controle.
- Implementação de multi-resolução utilizando *wavelets*: É uma técnica matemática para representação em multi-resolução, bastante utilizada em processamento de sinais e compressão de imagens [4].
- Implementação da variante *arc-strip tree*: Sendo a *arc tree* mais rápida e a *strip tree* mais precisa, combinando a construção da *strip tree* com os métodos de acesso da *arc tree* é possível que se tenha uma nova representação melhor.
- Combinar representação em multi-resolução com métodos de acesso espacial: A utilização de métodos mais rápidos de acesso às linhas poligonais já representadas em multi-resolução fará com que os resultados temporais das operações sejam mais rápidos.
- Implementar a representação em multi-resolução para regiões: A utilização da representação em multi-resolução pode ser estendida para regiões, onde estas são

compostas de linhas e, por sua vez, já representadas em multi-resolução, conseguindo assim, aplicar as operações de seleção e interseção.

- Disponibilizar uma biblioteca contendo todas as representações implementadas neste estudo.
- Implementar a *ball tree*: Uma outra representação em multi-resolução que tem como região limitante um círculo. Diferente da implementação utilizada na *arc tree*, a *ball tree* utiliza o menor círculo que engloba a poligonal.

## Referências

- [1] BALLARD, D.H., "Strip Trees: Hierarchical Representation for Curves", *Communications of the ACM*, **24**, No.5, 1981, pp.310–321.
- [2] CARVALHO, P. C. e FIGUEIREDO, L. H., *Introdução à Geometria Computacional*, IMPA, 1991.
- [3] DOUGLAS, D.H. e PEUCKER, T.K., "Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature", *The Canadian Cartographer*, **10**, No.2, 1973, pp.112–122.
- [4] FINKELSTEIN, A. e SALESIN, D.H., "Multiresolution Curves", *Proceedings of SIGGRAPH '94*, ACM, New York, 1994, pp.261–268.
- [5] GÜNTHER, O. e WONG, E., "The Arc Tree: an Approximation Scheme to Represent Arbitrarily Curved Shapes", *Computer Vision, Graphics and Image Processing*, **51**, No.3, 1990, pp.313–337.
- [6] GÜNTHER, O. e DOMINGUEZ, S., "Hierarchical Schemes for Curve Representation", *IEEE Computers Graphics & Applications*, **13**, No.3, 1993, pp.55–63.
- [7] HERSHBERGER, J. E SNOEYINK, J., "Speeding up the Douglas-Peucker line simplification algorithm", *Proc. 5<sup>th</sup> Intl. Symp. Spatial Data Handling*, 1992, pp.134–143.
- [8] MEDIANO, M.R., GATTASS, M. e CASANOVA, M.A., "HPS-tree: Um método de acesso para armazenar mapas longos com multi-resolução geométrica e topológica", *Anais do IX SIBGRAPI*, 1996, pp.219–226.
- [9] MEDIANO, M.R., CASANOVA, M.A. e DREUX, M., "V-trees: A Storage Method for Long Vector Data", *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994, pp.321–330.
- [10] NEWMAN, W.N. e SPROULL, R.F., *Principles of Interactive Computer Graphics*, 2ª edição, McGraw-Hill, 1979.



- [11] SAMET, H., *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990, pp.228–233.
- [12] SCHNEIDER, P.J., "Solving the Nearest-Point-on-Curve Problem", in: *Graphics Gems*, Andrew S.Glassner, Academic Press, 1990, pp.607–611.
- [13] SEDERBERG, T.W., WHITE, S.C. e ZUNDEL, A.K., "Fat arcs: A bounding region with cubic convergence", *Computer Aided Geometric Design*, **6**, 1989, pp.205–218.
- [14] CIA World Data Bank: [ftp://sepftp.stanford.edu/pub/World\\_Map/](ftp://sepftp.stanford.edu/pub/World_Map/)
- [15] Municípios dos Estados Unidos: <ftp://ftp.blm.gov/pub/gis/>
- [16] Estados e Municípios do Brasil: <http://www.ibge.gov.br>
- [17] Curvas de Nível: <ftp://spectrum.xerox.com/ds9/map/dlg/1:24,000/optional/>

# **Um estudo comparativo de representações de multi-resolução para linhas poligonais**

**Dissertação de Mestrado apresentada por ANDRÉ LUIZ BEHRENSDORF  
DERRAIK em 25 de abril de 1997 ao Departamento de Informática da PUC-Rio e  
aprovada pela comissão julgadora formada por:**

---

**Prof. Marcelo Gattass (DI/PUC-Rio)  
Orientador**

---

**Prof. Luiz Henrique de Figueiredo (LNCC)  
Co-orientador**

---

**Prof. Luiz Fernando Martha (DEC/PUC-Rio)**

---

**Prof. Paulo Cezar Pinto Carvalho (IMPA)**

**Visto e permitida a impressão.**

**Rio de Janeiro,**

---

**Coordenador dos Programas de Pós-  
Graduação do Centro Técnico Científico da  
PUC-Rio**