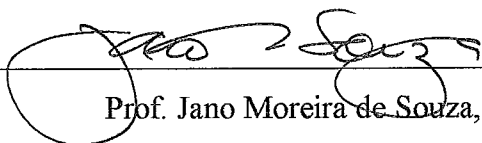


# FILTROS RASTER PARA JUNÇÕES DE POLILINHAS

Leonardo Guerreiro Azevedo

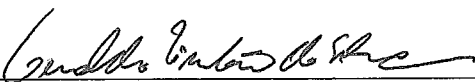
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



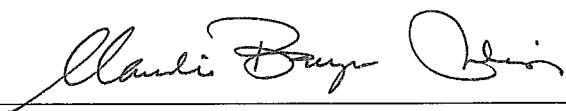
---

Prof. Jano Moreira de Souza, Ph.D.



---

Prof. Geraldo Zimbrão da Silva, D.Sc.



---

Prof. Claudia Bauzer Medeiros, Ph.D.



---

Prof. Claudio Esperança, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 2001

AZEVEDO, LEONARDO GUERREIRO

Filtros Raster para Junções de Polilinhas  
[Rio de Janeiro] 2001.

VI, 119 p. 29,7 cm (COPPE/UFRJ, M.Sc.,  
Engenharia de Sistemas e Computação, 2001)

Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1. Banco de Dados Espaciais

I. COPPE/UFRJ II. Título ( série )

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## FILTROS RASTER PARA JUNÇÃO DE POLILINHAS

Leonardo Guerreiro Azevedo

Agosto/2001

Orientadores:           Jano Moreira de Souza

Geraldo Zimbrão da Silva

Programa: Engenharia de Sistemas e Computação

O processamento eficiente de consultas é uma característica muito importante para bancos de dados em geral. O acesso à representação completa de um objeto espacial durante o processamento de uma consulta pode ser muito custoso. Pesquisas de algoritmos e estruturas de dados baseados na utilização de representações mais simples e compactas dos objetos são realizadas objetivando o processamento eficiente das consultas. Esta tese apresenta a RIOS e a 5CDRS, propostas de aproximações *raster* a serem utilizadas para polilinhas durante o processamento de consultas espaciais. Os resultados aqui apresentados concentram-se na utilização destas aproximações para o processamento de junções espaciais de polilinhas com polilinhas utilizando interseção como predicado espacial, consulta mais frequente em bancos de dados espaciais. Os resultados experimentais obtidos demonstraram que a eficiência das aproximações *raster* é superior às aproximações encontradas na literatura.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## RASTER FILTERS FOR POLYLINE JOINS

Leonardo Guerreiro Azevedo

August/2001

Advisors: Jano Moreira de Souza

Geraldo Zimbrão da Silva

Department: Systems and Computing Engineering

Efficient query processing is an important requirement on databases in general. The access to the full representation of a spatial object during a query processing can be very expensive. Researches on algorithms and data structures, based on simpler and more compact objects representations, have been done aiming a more efficient query processing. This thesis introduces RIOS and 5CDRS, raster approximations proposed to be used with polylines during spatial query processing. The results presented here focus the use of such approximations in processing spatial joins on polylines with polylines using intersection as the spatial predicate, most common operation in a spatial database. The experimental results demonstrate that the efficiency of raster approximation is higher than the ones found in the literature.

# Índice

1. Introdução .....	1
2. Processamento de consultas espaciais .....	4
2.1. Definição de conceitos básicos .....	5
2.2. Junção espacial .....	8
2.3. Arquitetura para processamento de junção espacial .....	9
2.4. Métodos de acesso espacial .....	11
2.4.1 Quadtree .....	12
2.4.2. Árvore R .....	19
2.5. Filtro geométrico .....	24
2.6. Teste de geometria exata .....	25
3. Filtros geométricos .....	27
3.1. Tipos de aproximações .....	28
3.2. Assinatura Raster de Quatro Cores .....	32
3.3. Problemas com a utilização das aproximações para polígonos na representação de polilinhas .....	34
3.4. Divisão do espaço em células .....	38
3.5. Sumário .....	40
4. RIOS .....	41
4.1. Composição da assinatura .....	42
4.2. Geração da aproximação .....	44
4.3. Mapa de posições candidatas .....	49
4.4. Mudança de resolução .....	51
4.5. Algoritmo de comparação .....	58
4.6. Armazenamento .....	61
4.7. Sumário .....	62
5. 5CDRS .....	63
5.1. Composição da aproximação .....	64

5.2. Geração da aproximação.....	68
5.3. Armazenamento.....	73
5.4. Algoritmo de comparação.....	73
5.5. Mudança de resolução .....	77
5.6. Sumário.....	80
6. Resultados Experimentais .....	81
6.1. Comparação entre as aproximações RIOS e 5CDRS.....	81
6.2. Estruturas utilizadas na arquitetura MSQP .....	82
6.2. Descrição dos dados.....	83
6.3.1. Dados dos objetos reais.....	83
6.3.2. Aproximações e Árvores.....	85
6.4. Análise sobre a execução de junções .....	86
6.5. Sumário.....	94
7. Conclusão .....	95
8. Bibliografia .....	97
ANEXOS .....	102
ANEXO A .....	103
ANEXO B.....	105
ANEXO C.....	107
ANEXO D .....	108
ANEXO E.....	110
ANEXO F .....	112
ANEXO G .....	114
ANEXO H .....	116
ANEXO I.....	118

# 1. Introdução

Sistemas de banco de dados espaciais vêm adquirindo importância cada vez maior tanto em setores públicos como privados. Aplicações relacionadas ao planejamento urbano, gerenciamento de recursos ambientais, planejamento de cultivo do solo e de identificação das melhores áreas para exploração econômica são alguns exemplos de aplicações que necessitam lidar com dados que possuem tanto características escalares como também espaciais. Entendem-se como dados espaciais os objetos compostos de pelo menos um atributo espacial tal como pontos, linhas, polígonos, regiões, retângulos, superfícies, volumes além de outros em maiores dimensões.

Idealmente, bancos de dados espaciais devem ser capazes de atender, de maneira eficiente, tanto as consultas realizadas sobre os dados espaciais, como os não espaciais ou ainda a combinação de ambos, sem favorecer um tipo de dado específico. Todavia, esta situação ideal ainda não foi contemplada. Ainda existem muitas questões a serem respondidas. Dados espaciais têm a peculiaridade de se apresentar em um grande volume e estarem dispostos em várias dimensões. Tais características, além de muitas outras, fazem com que novas estruturas de dados e algoritmos tenham que ser desenvolvidos para atender requisitos de armazenamento e execução de consultas. Em (ZIMBRÃO, 1999) é enfatizado o grande número de pesquisas que têm sido realizadas para definição de um padrão para construção de banco de dados espaciais. Como exemplo, esforços vêm sendo empregados para a criação de um padrão SQL para consultas espaciais (SQL3, 1996; SAIF, 1995; KAUSHIK e RUNDENSTEINER, 1998). Estruturas hierárquicas foram desenvolvidas para o armazenamento de índices para objetos espaciais sendo amplamente utilizadas. A árvore R (GUTTMAN, 1984) e a Quadtree (SAMET, 1990) são exemplos de tais estruturas. Estas serão detalhadas na seção 2.4 do capítulo 2.

Assim como em bancos de dados convencionais, consultas espaciais devem ser decompostas em consultas menores, cuja simplicidade permita execução de um número pequeno de operadores. Os algoritmos empregados na execução de consultas relacionais não são apropriados para a execução de consultas espaciais. BRINKHOFF et al. (1994) apresentam um processador modular para a execução de junções espaciais. Este modelo

tem sido amplamente citado e empregado na literatura, e com isso o trabalho aqui apresentado foi desenvolvido de forma a ser compatível com tal processador.

Considerando-se que os três tipos de dados espaciais mais comuns aos bancos de dados espaciais são pontos, polilinhas e polígonos, teremos nove classes de junções espaciais diferentes. Pela sua utilidade e dificuldade, a junção de polígonos com polígonos tem sido a mais investigada, ao passo que pela sua semelhança com a junção relacional as junções envolvendo pontos são as menos pesquisadas (SAMET, 1990). No entanto, a junção de polilinhas com polilinhas ou com polígonos também são operações de grande utilidade nos bancos de dados espaciais, e que apresentam dificuldades semelhantes às junções de polígonos com polígonos.

A importância da utilização de polilinhas na representação de dados espaciais pode ser demonstrada através de várias aplicações. Um primeiro exemplo seria a representação da malha rodoviária, ferroviária e metroviária de uma cidade permitindo à administração municipal determinar políticas de correção, controle e acompanhamento das redes de transportes. Simulações e consultas sobre estes dados e outros existentes na base de dados permitem estimar a melhor configuração segundo o custo e recursos disponíveis. Outro exemplo seria a modelagem de redes elétricas, de gás, de TV a cabo, telecomunicações, entre outras, atendendo necessidades de gerência de recursos valiosos pela definição de suas localizações e conectividades, além de recuperar o estado corrente de cada recurso. Podemos citar, ainda, a exploração e produção de hidrocarbonetos. Uma base de dados espaciais para estas aplicações armazena dutos (oleodutos, gasodutos, carbodutos e ignodutos), hidrografia (rios), hipsometria (curvas) e sistemas viários (rodovias e ferrovias) representados por polilinhas. Análises sobre a base possibilitam planejamentos de forma a não ter junção entre sistema viário e dutos ou hidrografia e dutos. Simulações de vazamento de fluido permitem fazer análises de áreas de risco ajudando a agir de forma pró-ativa impedindo que catástrofes ocorram.

A maior parte dos esforços atualmente se concentram na elaboração de algoritmos eficientes para a realização de junções espaciais envolvendo polígonos (regiões), tais como (ZIMBRÃO, 1998). No entanto, as técnicas desenvolvidas pressupõem o uso de aproximações (filtros) que nem sempre podem ser eficientemente aplicadas a polilinhas.



Neste trabalho, apresentaremos aproximações especificamente projetadas para representar polilinhas, e que podem ser utilizadas para a realização de junções tanto de polilinhas com polilinhas quanto polilinhas com polígonos. Estas aproximações são a RIOS publicada no relatório técnico ZIMBRÃO et al. (2001) e a 5CDRS publicada no artigo (ZIMBRAO et al., 2000), que foi submetido, aprovado e apresentado no XV Simpósio Brasileiro de Banco de Dados de 2000 . Em particular, aqui apresentaremos apenas os resultados de junções entre conjuntos de polilinhas. As junções de polilinhas com polígonos é um trabalho apresentado em (MONTEIRO, 2001b). Os resultados obtidos com a utilização das aproximações como filtro geométrico no segundo passo da arquitetura de múltiplos passos de BRINKHOFF et al. (1994) foram promissores. Uma redução de mais de 50% no número necessário de testes de interseção exata foi obtido sendo superior ao das aproximações existentes na literatura.

Este trabalho está organizado da seguinte forma. O capítulo 1 é a presente introdução. O capítulo 2 apresenta os conceitos de processamento de junções espaciais que mais se relacionam com o foco desta tese. O capítulo 3 detalha o segundo passo da arquitetura MSQP (BRINKHOFF et al., 1994) no qual se enquadram as estruturas aqui propostas para aproximação de polilinhas. No capítulo 4 a RIOS é apresentada, sua principal característica consiste no fato de aproximar uma polilinha armazenando os pontos de entrada e saída da mesma nas células de uma grade. O capítulo 5 apresenta a 5CDRS a qual armazena os sucessivos deslocamentos da polilinha nas células da grade para construir a aproximação. O capítulo 6 expõe os resultados experimentais, apresentando uma comparação teórica entre a RIOS e a 5CDRS, além dos testes e dos resultados. Finalmente, no capítulo 7 temos as conclusões dessa tese.

## 2. Processamento de consultas espaciais

Este capítulo destina-se a apresentar conceitos e estruturas relacionadas ao processamento de consultas espaciais. Para tanto, uma revisão da literatura é apresentada enfatizando as principais estruturas e métodos referenciados por trabalhos da área. O ambiente e o escopo da tese são aqui elucidados. É importante ressaltar que o presente capítulo faz parte de um trabalho conjunto com (MONTEIRO, 2001b) constituindo também um capítulo do mesmo.

Caso o leitor já tenha conhecimento sobre processamento de consultas em um banco de dados espaciais e da arquitetura de múltiplos passos (BRINKHOFF et al., 1994), poderá se dirigir ao capítulo 3. Por outro lado, para a leitura do presente capítulo é pressuposto o conhecimento dos conceitos relacionados ao processamento de consultas em banco de dados tradicionais.

Um banco de dados espacial deve oferecer as mesmas funcionalidades para os dados espaciais que um banco de dados relacional oferece aos dados convencionais. Logo, um banco de dados espacial deve ser capaz de responder consultas *ad-hoc* sobre os atributos espaciais armazenados. Consultas devem ser decompostas em consultas menores, mais simples, que possam ser implementadas através de um conjunto pequeno de operadores espaciais.

Considere, como exemplo, um banco de dados que possua uma relação de todos os estados dos Estados Unidos e outra com todos os rios deste país (Figura 2.1). Poderíamos executar uma consulta neste banco solicitando uma relação por estado de todos os rios que cruzam o seu território. Poderíamos também querer saber os rios que passam por pelo menos três estados diferentes, a indicação do estado interceptado pelo maior número de rios, além de diversas outras possibilidades.

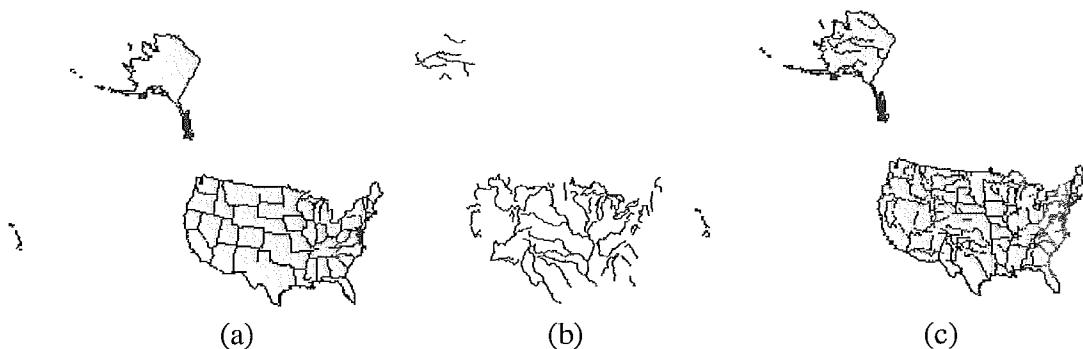


Figura 2.1 – (a) Mapa político (a) e hidrográfico (b) dos Estados Unidos e sua sobreposição (c).

As consultas apresentadas acima combinam objetos espaciais das relações de acordo com seus atributos geométricos caracterizando consultas espaciais.

Este capítulo está estruturado da seguinte forma: na seção 2.1 são definidos conceitos relevantes para o melhor entendimento do presente trabalho; a seção 2.2 apresenta com maiores detalhes o conceito de junção espacial; na seção 2.3 é apresentada uma arquitetura modular para o processamento de consultas espaciais. As seções seguintes enfatizam cada passo desta arquitetura. Na seção 2.4 serão estudados alguns métodos de acesso espacial, destacando *Quadtree* e *Árvore R*. A seção 2.5 apresenta filtros geométricos; e, finalmente, a seção 2.6 o teste de geometria exata.

## 2.1. Definição de conceitos básicos

Esta seção tem por finalidade definir formalmente alguns termos e conceitos relevantes para o entendimento do presente trabalho.

- **Dado espacial** é um termo usado para descrever dados que contém informações sobre o espaço ocupado pelos objetos de uma base de dados. Estes dados são geométricos e diversificados. Consistem de pontos, linhas, retângulos, polígonos, superfícies, volumes e até mesmo de dados em dimensões superiores. Dados espaciais são normalmente acompanhados por atributos ou dados não espaciais, como por exemplo, o nome de um rio ou tipo de solo de uma região.

- **Banco de dados espaciais** é um sistema de banco de dados que oferece suporte a tipos de dados espaciais tanto em seu modelo de dados como em sua linguagem de consulta, implementando tais tipos e oferecendo pelo menos índices espaciais e algoritmos eficientes para operações espaciais. (ZIMBRÃO, 1999)
- **Predicado espacial** é uma operação de restrição sobre os atributos espaciais de um conjunto a fim de extrair os elementos que satisfaçam a uma condição.
- **Imagem Raster** é um reticulado regular de células retangulares ordenadas por linha, o que torna desnecessário o armazenamento das coordenadas de cada célula. Cada célula de uma estrutura raster possui um e apenas um valor em uma camada temática. (ZIMBRÃO, 1999)
- **Retângulo envolvente mínimo** ou **MBR** (*Minimum Bounding Rectangle*) refere-se ao menor retângulo alinhado com os eixos cartesianos que contém inteiramente o objeto representado. Também comumente referenciado como caixa envolvente e caixa limitante constitui uma aproximação simples e por tal motivo amplamente usada na literatura. A Figura 2.2 apresenta o mapa do Rio de Janeiro e o seu MBR.

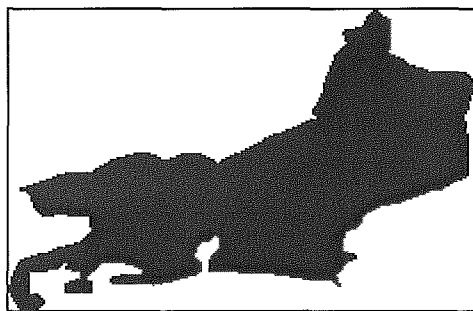


Figura 2.2 – Retângulo envolvente mínimo (MBR)

- Uma **polilinha** pode ser definida como um conjunto conexo de linhas poligonais, sendo estas últimas compostas por segmentos consecutivos. Cada linha poligonal é definida como uma seqüência de pontos onde cada ponto define com seu antecessor um segmento. Cada linha poligonal é chamada de parte. Portanto, uma polilinha é descrita através de suas partes e dos pontos que as definem. A Figura 2.3 apresenta a polilinha correspondente ao Rio Amazonas e seus afluentes.

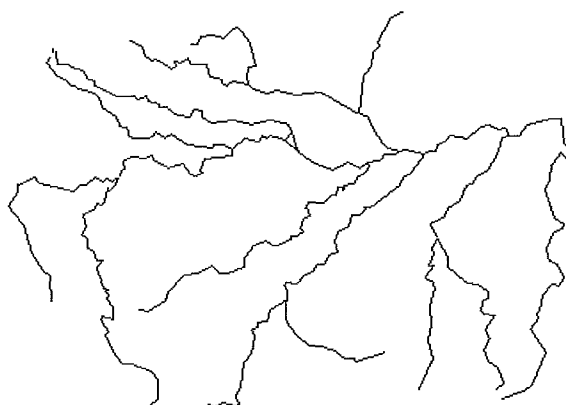


Figura 2.3 – Exemplo de polilinha

- Entende-se por **região ou imagem binária** aquela cujos elementos básicos podem assumir apenas dois valores distintos, como, por exemplo, preto ou branco. Uma imagem binária de uma região pode ser obtida considerando os pontos internos como pretos ou 1 e os externos como brancos ou 0, como exemplificado na Figura 2.4.

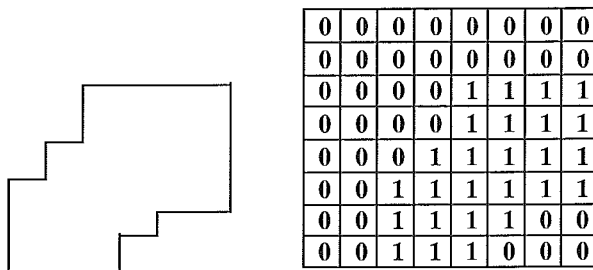


Figura 2.4 – Objeto e sua imagem binária

## 2.2. Junção espacial

Conforme apontado por BRINKHOFF et al. (1994), entende-se a junção espacial como um subconjunto do produto cartesiano de dois conjuntos, A e B, não necessariamente distintos, contendo respectivamente  $m$  e  $n$  elementos. Esse subconjunto é formado pelos elementos do produto cartesiano que atendam a um determinado predicado espacial.

Junções espaciais podem ser mais eficientemente avaliadas através da construção de índices sobre os conjuntos de dados, de forma que simultaneamente se percorra estes índices buscando por pares de objetos espaciais que se interceptam. Esta abordagem corresponderia à abordagem *sort-merge* dos bancos de dados relacionais, e é apontada por BRINKHOFF et al. (1993a) como sendo bastante eficiente, especialmente no caso de preexistência dos índices na base. Sem a utilização de índices no processamento da consulta teremos o tempo de processamento crescendo exponencialmente com o número de objetos envolvidos, já que para cada elemento de um conjunto todos os elementos do outro deverão ser testados. Além disso, o tamanho dos objetos muitas vezes pode impedir que todos os elementos necessários para a comparação sejam trazidos para a memória, implicando em grandes volumes de dados transferidos do disco para a memória e vice-versa (*swap*).

Tipicamente, um índice é composto de duas partes: a estrutura de índice que armazena apenas chaves para os dados, e a estrutura que armazena os dados em si. Desta forma, um índice espacial deve armazenar os objetos espaciais em uma estrutura espacial de acordo com uma chave geométrica (BRINKHOFF et al., 1994).

Devido à sua simplicidade, o MBR (*Minimum Bounding Rectangle* – menor retângulo envolvente) é a chave geométrica de uso mais difundido. Quando utilizamos MBRs, a complexidade de um objeto espacial é reduzida a quatro parâmetros que detêm as mais importantes características do objeto: sua posição e sua extensão. Apesar disso, conforme apontado por BRINKHOFF et al. (1994) e ZIMBRÃO et al. (1998), os objetos do mundo real são muito mal aproximados por MBRs – em particular, polilinhas são pessimamente aproximadas por MBRs.

Nesse cenário, é de fundamental importância para a avaliação efetiva de consultas um algoritmo eficiente para a realização de junções espaciais. BRINKHOFF et al. (1994) definem de um processador modular de junções espaciais. Tal modelo tem sido muito citado em trabalhos subsequentes, e por isso o trabalho aqui apresentado foi desenvolvido de forma a ser compatível com tal processador.

### ***2.3. Arquitetura para processamento de junção espacial***

Visando uma melhor eficiência, organização e estudo do processo de desenvolvimento de índices e filtros para dados espaciais, BRINKHOFF et al. (1994) propuseram uma arquitetura de múltiplos passos ou MSQP (*Multi-Step Query Processor*) para o processamento de junções espaciais (Figura 2.5), cujos passos serão detalhados mais à frente. O principal objetivo desta arquitetura é acelerar os passos mais custosos através da redução do número de objetos espaciais a serem comparados. Tal redução é feita utilizando filtros em passos anteriores. Neste trabalho são propostas e avaliadas aproximações a serem empregadas como filtros geométricos no segundo passo.

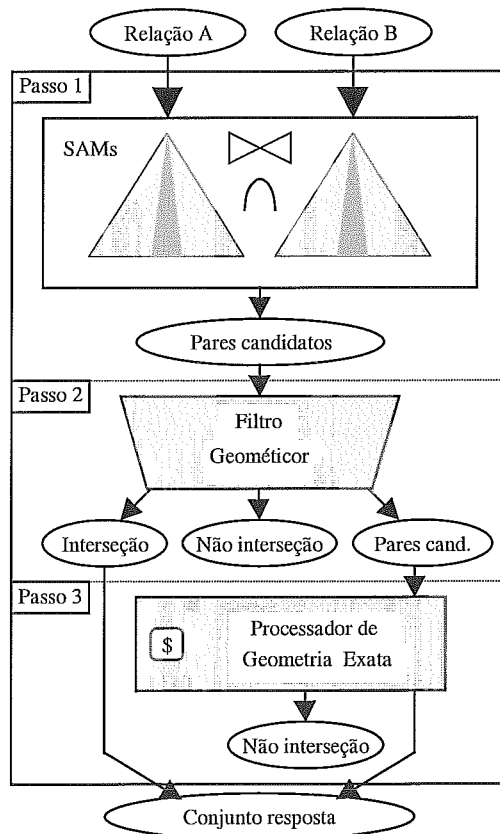


Figura 2.5 - Arquitetura de 3 passos.

A principal contribuição da arquitetura apresentada na Figura 2.5 é estabelecer uma modularização do processamento de junção espacial. O único ponto em comum entre módulos diferentes é o conjunto de objetos transferido de um para o outro, não importando a forma como este é computado. Este fato permite que diferentes algoritmos e estruturas de dados possam ser propostas de forma independente para cada etapa do processo.

O primeiro passo dessa arquitetura corresponde a um método de acesso espacial ou SAM (*Spatial Access Method*) que tem como objetivo restringir o espaço de busca. Os métodos de acesso tradicionalmente empregados usam MBRs dos objetos e por este motivo este passo não tem como saída o resultado da operação de junção, mas sim um conjunto de pares candidatos que corresponde a um superconjunto da solução. Esse passo é conhecido como junção de MBRs, e cabe ressaltar que qualquer que seja o método utilizado para



implementá-lo, o produto final deverá ser o mesmo conjunto, pois é composto pelos MBRs que se interceptam.

O segundo passo consiste em comparar os pares de candidatos resultantes do passo anterior através de um filtro geométrico. Um filtro geométrico usa uma representação aproximada e compacta do objeto, que procura guardar as principais características do mesmo, sendo alguns exemplos apresentados no capítulo 3. Como resultado da comparação dos filtros dos dois objetos candidatos, temos três possibilidades: os dois objetos pertencem à solução (*hit*); os dois objetos não pertencem à solução (*false hit*); e inconclusivo.

O terceiro passo tem como entrada os pares de objetos que tiveram a comparação de seus filtros geométricos inconclusiva. Este passo consiste em compará-los através da sua real representação. Este é o mais custoso requerendo tempo de CPU para computar a interseção exata e tempo de I/O para ler os objetos espaciais do disco, conforme resultados obtidos por BRINKHOFF et al. (1994). A quantidade de tempo gasta neste passo pode ser consideravelmente reduzida com melhores aproximações nos passos anteriores.

Assim, invariavelmente todos os algoritmos de junção espacial que têm sido propostos nos últimos anos realizam uma pré-junção sobre os MBRs dos polígonos, e somente então passam a investigar quais os pares desse conjunto resposta que farão parte do resultado da junção original. Muitos desses trabalhos param nesse passo, não chegando a realizar os passos 2 e 3 do MSQP: realizando apenas a junção de MBRs. Por outro lado, alguns métodos não implementam o passo 2, passando o conjunto produzido pela junção de MBRs diretamente para o passo 3.

#### ***2.4. Métodos de acesso espacial***

Esta seção apresenta métodos de acesso espacial candidatos a desempenhar o primeiro passo do MSQP. Duas classes de métodos recebem atenção especial devido a seu amplo uso na literatura: a Quadtree, apresentada na seção 2.4.1; e, a Árvore R, apresentada na seção 2.4.2. No final da seção, outros métodos são referenciados.

### 2.4.1 Quadtree

Segundo SAMET (1990) o termo Quadtree é usado para descrever uma classe de estruturas de dados hierárquica cuja propriedade comum é o princípio da decomposição recursiva do espaço. Estas estruturas, propostas pelo mesmo autor, podem se diferenciar nas seguintes características:

- Tipo de dado a ser representado
- Critério usado para o processo de decomposição
- A resolução (variável ou não)

As Quadtrees são usadas para os mais diversos tipos de dados como pontos, áreas, curvas, superfícies e volumes. A decomposição pode ser regular (divisão em partes iguais a cada nível) ou determinada por características do novo dado. Como exemplos temos a Quadtree de região, para o primeiro caso, e Quadtree de pontos, para o segundo. Detalhes destas duas estruturas serão apresentados mais à frente. Na área de computação gráfica, esta diferenciação de decomposição do espaço é referenciada respectivamente como hierarquias no espaço da imagem e hierarquias no espaço do objeto (SUTHERLAND et al., 1974). O último item, resolução, refere-se ao número de vezes que o processo de decomposição é aplicado, podendo ser pré-determinada ou obtida em função de características do novo dado. Uma quarta característica relevante para algumas aplicações pode ser acrescentada referindo-se a representação das fronteiras do objeto (curvas e superfícies) ou organização de seu interior (áreas e volumes).

Para exemplificar algumas das características apresentadas acima e melhorar entendimento do propósito da Quadtree, são apresentadas a seguir duas estruturas dessa classe: Quadtree de região e Quadtree de pontos.

A Quadtree de Região destina-se à representação de regiões binárias bidimensionais. Uma vez definidos os limites da região com base em sua borda esta estrutura baseia-se na subdivisão sucessiva do espaço em quatro quadrantes de mesmo tamanho. O critério usado para decomposição do espaço consiste em efetuar uma nova subdivisão em um quadrante enquanto este não for homogêneo, ou seja, possuir apenas

elementos iguais. Esta Quadtree pode ser classificada como uma estrutura de resolução variável.

Como exemplo de uma Quadtree de região considere a região apresentada na Figura 2.6.a e a sua imagem binária  $2^3 \times 2^3$  na Figura 2.6.b, onde 1 representa elementos dentro da região e 0 elementos fora da região. Os quadrantes resultantes do processo de decomposição são apresentados na Figura 2.6.c e a Quadtree correspondente na Figura 2.6.d.

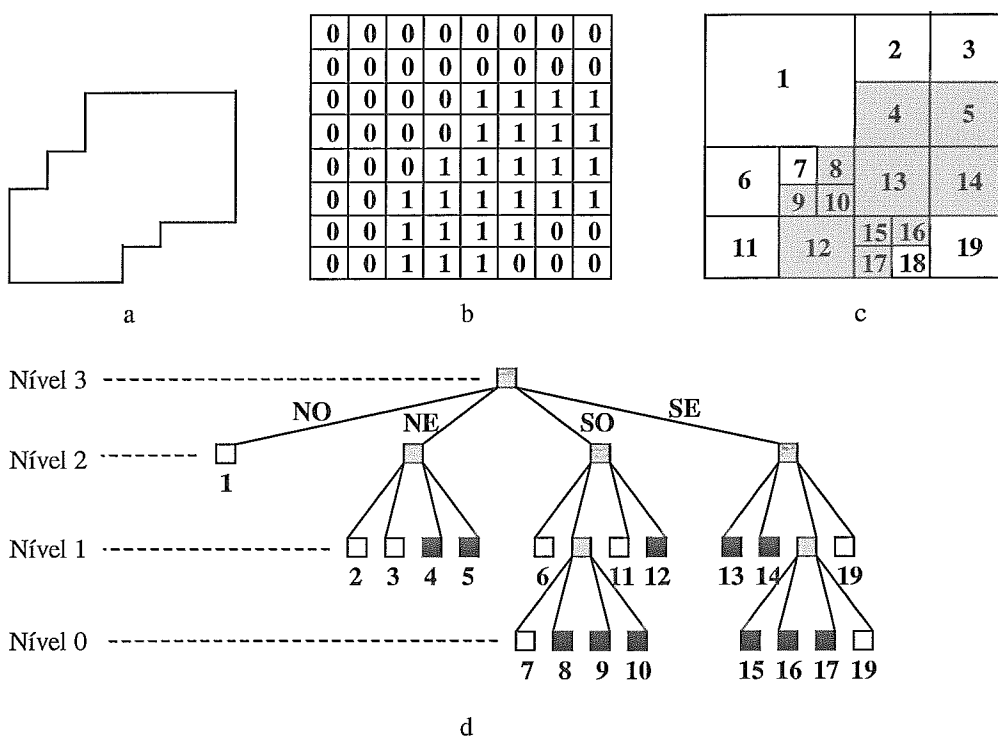


Figura 2.6 – Quadtree de região

Na representação da Figura 2.6.d, o nó raiz corresponde a toda a imagem binária da Figura 2.6.b. Cada um dos quatro filhos representa um quadrante resultante da subdivisão da região representada pelo nó, rotulados com NO, NE, SO e SE. Os nós folhas da árvore correspondem a regiões homogêneas onde, portanto, a subdivisão não foi necessária. Um nó folha é rotulado como preto ou branco dependendo se este representa uma porção dentro ou

fora da região representada. Os nós internos são rotulados como cinza. O nó raiz se encontra no nível N enquanto o nível 0 corresponde a um único elemento da imagem.

A Quadtree de pontos destina-se à representação de um conjunto de pontos num espaço bidimensional. A principal diferença em relação a Quadtree de região fica por conta da decomposição do espaço. Ao invés de regular, a decomposição ocorre em função da entrada, permitindo a formação de quadrantes retangulares. Esta estrutura é tida como uma adaptação da árvore binária de busca (KNUTH, 1973), onde cada ponto, com coordenadas  $X_p$  e  $Y_p$ , divide o seu quadrante em quatro novos quadrantes da seguinte forma:

- Quadrante NO:  $X < X_p$  e  $Y > Y_p$
- Quadrante NE:  $X > X_p$  e  $Y > Y_p$
- Quadrante SO:  $X < X_p$  e  $Y < Y_p$
- Quadrante SE:  $X > X_p$  e  $Y < Y_p$

A inserção de um novo ponto consiste em percorrer a árvore a partir da raiz até alcançar uma folha, a qual corresponde ao quadrante sem subdivisão que contém ponto, onde este será inserido. A descida na árvore é efetuada comparando, segundo as regras acima, as coordenadas do novo ponto com as do ponto do ramo sendo percorrido. Um novo ponto adicionado à estrutura sempre leva a criação de quatro novos quadrantes.

Como exemplo de uma Quadtree de pontos considere a Figura 2.7, onde os pontos foram inseridos na seguinte ordem: Teresópolis (1), Niterói (2), Rio de Janeiro (3), Friburgo (4), Búzios (5), Cabo Frio (6) e Petrópolis (7).

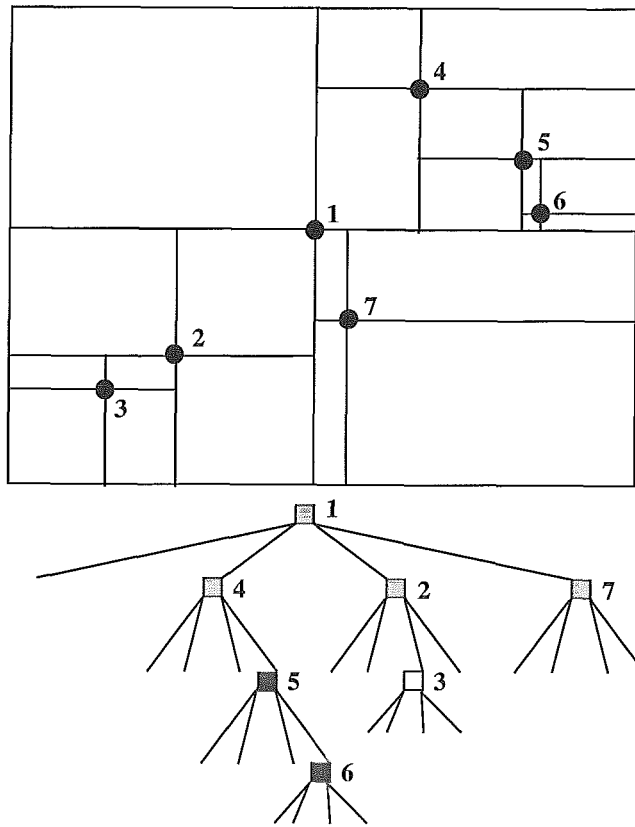


Figura 2.7 - Quadtree de pontos

O formato da árvore é altamente dependente da ordem de inserção dos pontos, assim como ocorre na árvore binária de busca.

Devido ao seu amplo e difundido uso ao longo dos anos<sup>1</sup>, estruturas pertencentes à classe das Quadrees surgiram cada qual com adaptações visando solucionar problemas específicos. Como exemplos de estruturas candidatas para armazenar os MBRs no primeiro passo do MSQP temos MX-CIF Quadtree<sup>2</sup>, MX-CIF Quadtree expandida e RR Quadtree as quais são apresentadas a seguir.

<sup>1</sup> uma Quadtree já era usada como método de acesso espacial por MORTON (1966).

<sup>2</sup> MX-CIF abreviação em inglês de Matrix – Caltech Intermediate Form.

A MX-CIF Quadtree, pertencente à família das CIF Quadtrees, foi criada independentemente por KEDEM (1982) e por ABEL et al. (1983) e associa cada retângulo com um nó da Quadtree correspondendo ao menor bloco que o contém inteiramente. Desta forma, podem existir retângulos associados a nós internos. O processo de decomposição pára quando um bloco não contém inteiramente nenhum retângulo. Um critério alternativo consiste em estabelecer um tamanho mínimo, de forma que o bloco só poderá ser subdividido se seu tamanho for maior que o limite. KEDEM (1982) sugere que o limite seja igual ao tamanho esperado dos retângulos.

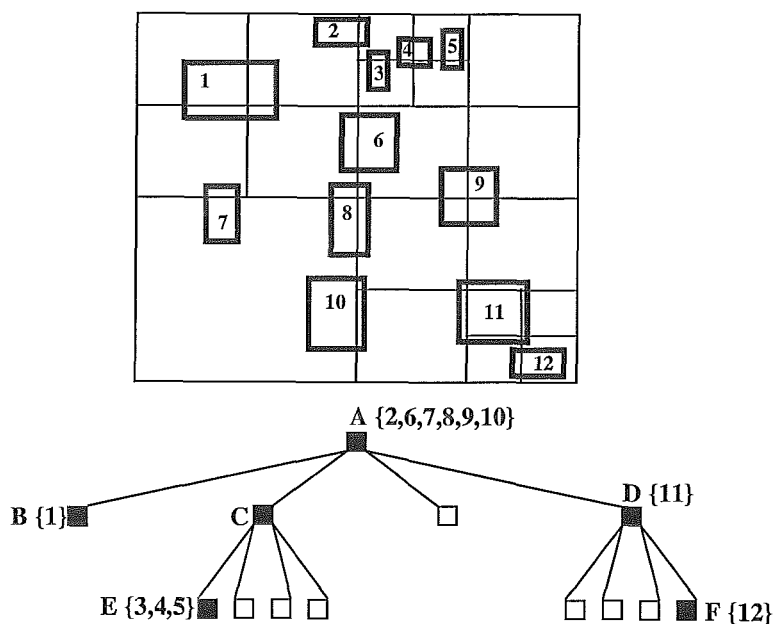


Figura 2.8 - MX-CIF Quadtree

A Figura 2.8 apresenta um exemplo da MX-CIF Quadtree. Note que um retângulo é associado a apenas um nó e não é considerado membro dos filhos deste. Como exemplo o retângulo 11 possui interseção com os nós D e F porém só está associado ao nó D. Por outro lado, mais de um retângulo pode ser associado a um nó, existindo diversas formas de organizá-los. Está organização pode variar desde nenhuma ordenação (ABEL et al., 1983) até formas mais elaboradas, como uma abordagem usando uma árvore binária para cada eixo descrita por KEDEM (1982).

A MX-CIF Quadtree associa a cada retângulo apenas um nó da árvore. Outra abordagem possível é a representação através de múltiplos blocos ou nós. A MX-CIF Quadtree expandida e RR Quadtree se enquadram nessa categoria.

A MX-CIF Quadtree expandida procura representar cada retângulo de uma forma mais compacta no que diz respeito à área. Proposta por ABEL (1985), esta representação segue os mesmos passos de uma MX-CIF Quadtree se diferenciado quando o menor bloco contendo o retângulo inteiramente é encontrado. Neste ponto um número pré-fixado de subdivisões é efetuado no bloco. O retângulo é representado pelo conjunto de blocos sem subdivisão da Quadtree que contém alguma porção do objeto. Em outras palavras, cada retângulo é representado pelo conjunto de nós folhas que contém alguma porção do objeto. Note que na MX-CIF Quadtree um retângulo pode ser associado a um nó interno. Um exemplo de MX-CIF Quadtree expandida é apresentado na Figura 2.9 juntamente com a MX-CIF Quadtree para o mesmo conjunto de retângulos. Nesse exemplo o número pré-fixado de subdivisões é igual a 1, ou seja, uma nova subdivisão é efetuada no bloco contendo o retângulo inteiramente.

A RR Quadtree criada por SHAFFER (1986) possui duas variantes denominadas  $RR_1$  e  $RR_2$  Quadtree. O que as diferencia é o critério de decomposição. Na  $RR_1$  Quadtree um bloco é subdividido até que contenha apenas um retângulo ou todos os retângulos possuam interseção entre si. Na  $RR_2$  Quadtree um bloco é subdividido até que contenha apenas um retângulo ou todos os retângulos façam parte de uma cadeia de interseção. A Figura 2.10 apresenta exemplos dessas representações.

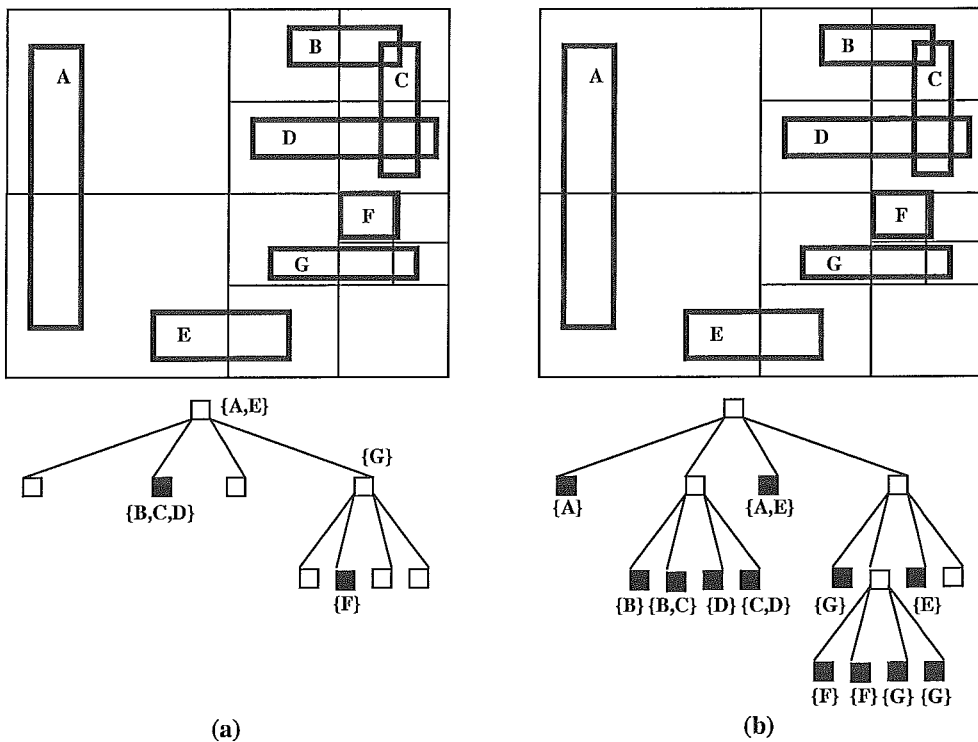


Figura 2.9 - (a) MX-CIF quadtree (b) MX-CIF quadtree expandida

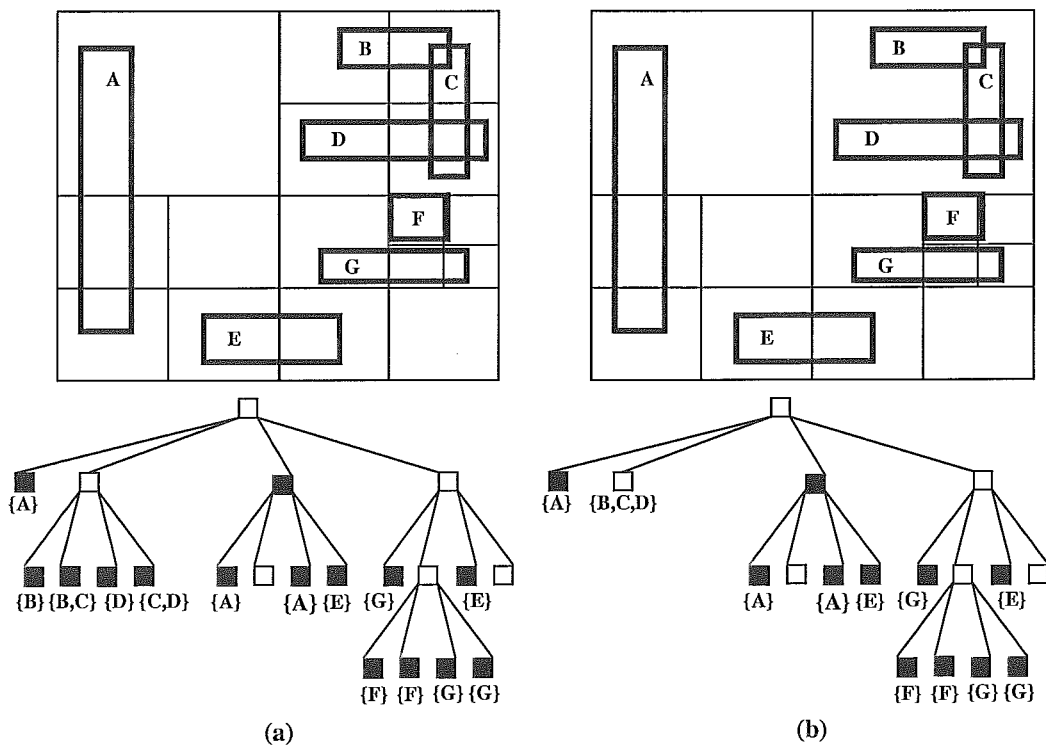


Figura 2.10 - (a) RR<sub>1</sub> quadtree (b) RR<sub>2</sub> quadtree



### 2.4.2. Árvore R

A árvore R (GUTTMAN, 1984) é uma estrutura de dados hierárquica balanceada, nos moldes das árvores B, sendo amplamente utilizada na representação de coleções de dados geométricos através de retângulos em  $n$  dimensões. As regras básicas para a formação de uma árvore R são semelhantes às da árvore B como se segue:

- Todos os nós folha aparecem no mesmo nível;
- Um nó folha contém coleções de pares (R,O) na qual R é o menor retângulo que envolve o objeto apontado por O;
- Um nó interno contém coleções de pares (R,P) no qual R é o retângulo envolvente mínimo dos objetos que estão no nó filho apontado por P.
- Diz-se que uma árvore R tem ordem (m,M) quando cada nó contém entre  $m$  e  $M$  entradas, onde  $m \leq \lceil M/2 \rceil$ , excetuando-se a raiz que tem pelo menos duas entradas a não ser que esta seja nó folha.

Freqüentemente nós correspondem a páginas do disco, e os parâmetros que definem a árvore são determinados de forma que o número de nós visitados durante a busca seja o menor possível. Uma árvore R não é uma divisão do espaço e sim um agrupamento de objetos o que permite que diferentes nós se sobreponham. Assim, um mesmo retângulo pode estar associado a um ou a vários nós o que leva a possibilidade da busca requerer a visita a vários nós antes de averiguar a presença ou ausência de um retângulo específico.

Como exemplo, considere os objetos e seus respectivos retângulos envolventes mínimos da Figura 2.11. Para  $M = 3$  e  $m = 1$  temos uma possível árvore R para a coleção. Nós folha são representados com letras indicando os retângulos envolventes mínimos dos objetos que eles representam, em tonalidade mais escura. Linhas pontilhadas e números assinalam nós internos. Observe que a árvore R não é única; sua estrutura é dependente da ordem em que os nós foram inseridos e possivelmente removidos da árvore.

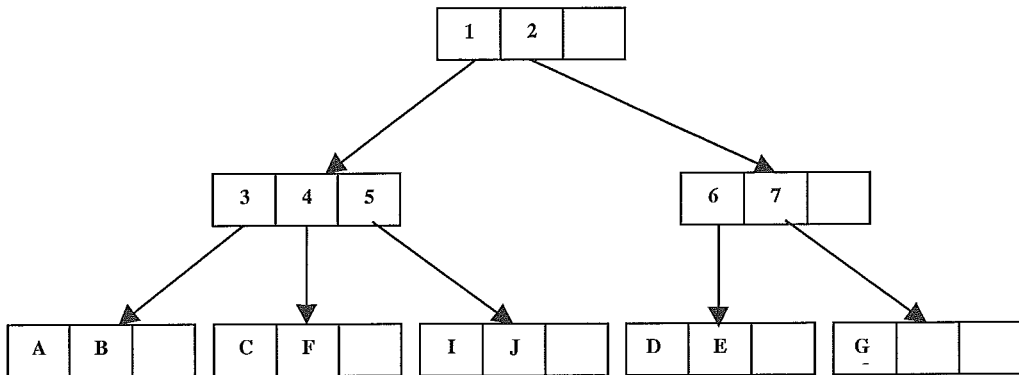
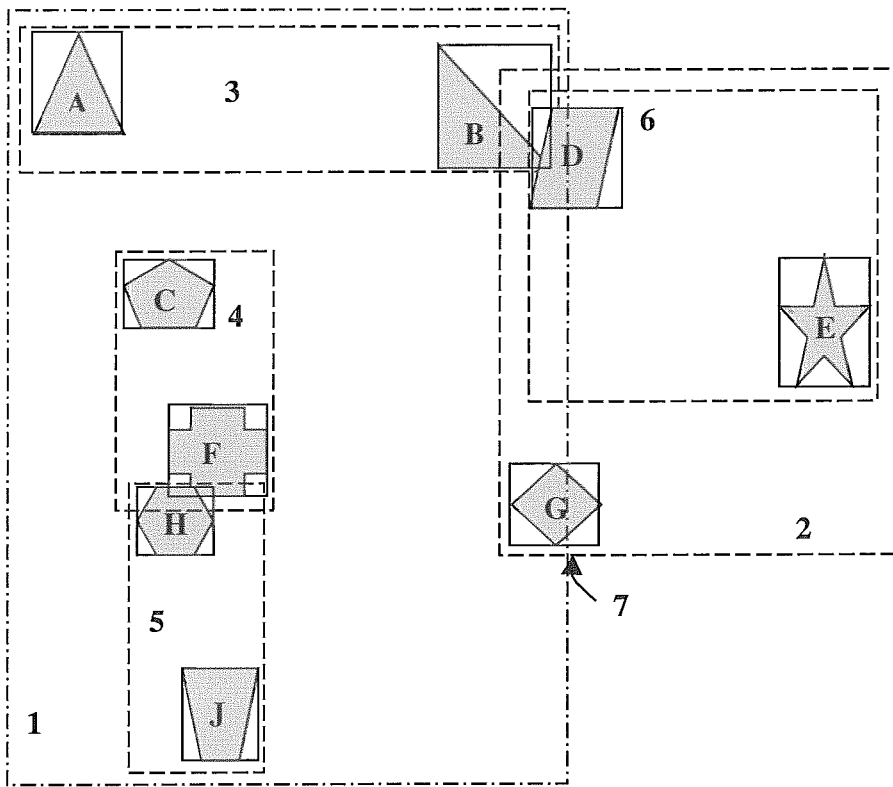


Figura 2.11 – Árvore R

A inserção de um novo objeto é semelhante a das árvores B. Novos retângulos são adicionados em nós folha. A determinação da folha apropriada é feita descendo a árvore a partir da raiz e a cada passo escolhendo a sub-árvore em que aumento de área do retângulo correspondente é mínimo. Uma vez determinado o nó folha, verifica-se se a inserção causará *overflow*. Se isto ocorrer, o nó é então dividido em dois, e as  $M+1$  entradas são

redistribuídas. A divisão é propagada para cima atualizando-se os MBRs das entradas do caminho da folha até a raiz. A árvore  $R$  utiliza como heurística para divisão de nós obter dois novos nós que possuam a menor área desperdiçada<sup>1</sup> possível.

A remoção de um retângulo  $R$  de uma árvore  $R$  ocorre através da localização do nó folha  $L$  contendo  $R$  e removendo  $R$  de  $L$ . Em seguida, é feito o ajuste dos retângulos das entradas dos nós no caminho de  $L$  até a raiz. Os nós que sofrem *underflow* são removidos e adicionados a um conjunto auxiliar, digamos  $U$ . Se a raiz sofre *underflow* e tem um filho, seu filho torna-se a nova raiz. As entradas do conjunto  $U$  são reinseridas em suas alturas originais.

A remoção pode também ser realizada analogamente à árvore  $B$ . Nós com *underflow* são redistribuídos em nós irmãos, podendo, além disso, também considerar para reinserção o irmão que sofrerá menor acréscimo de área. Entretanto, a reinserção é mais conveniente considerando o fato de que ela possibilita a árvore  $R$  refletir dinamicamente as mudanças na estrutura espacial ao invés de gradualmente degradar a árvore mantendo os retângulos sempre em suas mesmas posições.

Durante a busca numa árvore  $R$  um grande número de nós pode ser visitado pois o objeto buscado pode ter interseção com vários nós. Como exemplo, considere a busca do retângulo  $B$  (Figura 2.11). Começando pela raiz observamos que ambas as entradas 1 e 2 têm interseção com este retângulo, logo é necessário descer para as sub-árvores apontadas pelas mesmas. No nível seguinte, o nó apontado por 1 possui apenas a entrada 3 interceptando o objeto. O nó apontado por 2 possui a entrada 6 com interseção com o retângulo em questão sendo necessário também descer por este nó. A partir da entrada 3 vamos para um nó folha e neste alcançamos  $B$ . A partir da entrada 6 caminhamos para um nó folha que não contém  $B$ .

Existem outras maneiras para executar a divisão de nós. Uma delas é distribuir os MBRs entre os nós visando à redução do número de árvores visitadas durante a busca. Tal fato é obtido minimizando a área total de cobertura dos retângulos dos nós, ou seja, reduzindo a área de desperdício entre o MBR do nó e o conjunto de retângulos incluídos no

---

<sup>1</sup> Área desperdiçada de um nó é a área não coberta por MBR

mesmo. Uma outra alternativa é diminuir o número de nós visitados em buscas subseqüentes, o que pode ser alcançado minimizando a área comum entre ambos. A área de interseção entre retângulos em cada nível da árvore deve ser mínima. Muitas vezes os dois princípios são contraditórios como pode ser observado na Figura 2.12. A primeira alternativa aparece na Figura 2.12.a e a segunda na Figura 2.12.c.

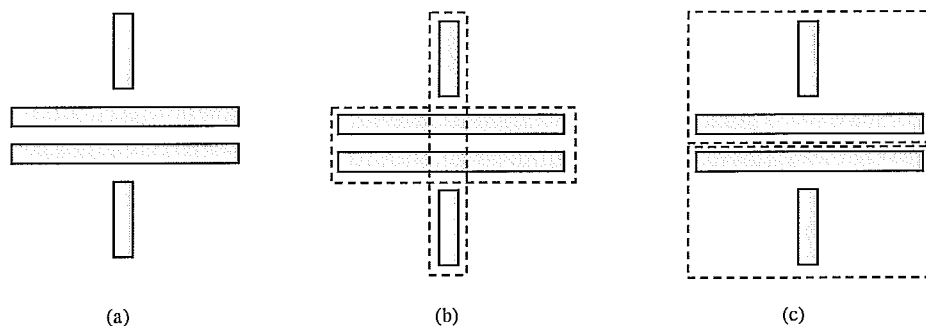


Figura 2.12 - (a) Quatro retângulos envolventes mínimos; (b) divisão minimizando a área de cobertura dos retângulos; (c) divisão minimizando a área de sobreposição entre os nós.

Outra heurística para divisão é minimizar o perímetro de cada MBR aproximando-os a quadrados. Quadrados são mais fáceis de agrupar do que retângulos. Pode-se também utilizar como heurística obter taxa máxima de utilização dos nós já que nós cheios diminuem o número de nós acessados durante a busca.

Uma variante das árvores R é a árvore R+, baseada no critério de evitar sobreposição entre os retângulos limitantes. Em particular, todos os retângulos limitantes, exceto os da folha, não possuem interseção. Cada retângulo envolvente mínimo dos objetos está associado com todos os retângulos limitantes que o interceptam. Como resultado existe mais de um caminho para alcançar um objeto. Conseqüentemente a altura da árvore é aumentada enquanto que o tempo de busca é acelerado.

A Figura 2.13 rerepresenta os objetos da Figura 2.11 considerando retângulos limitantes seguindo o critério da árvore R+. Observe que os retângulos B e F aparecem em dois nós diferentes e que não existe sobreposição entre retângulos limitantes de nós internos. Existem outras possibilidades para a representação da árvore R+ ilustrada.

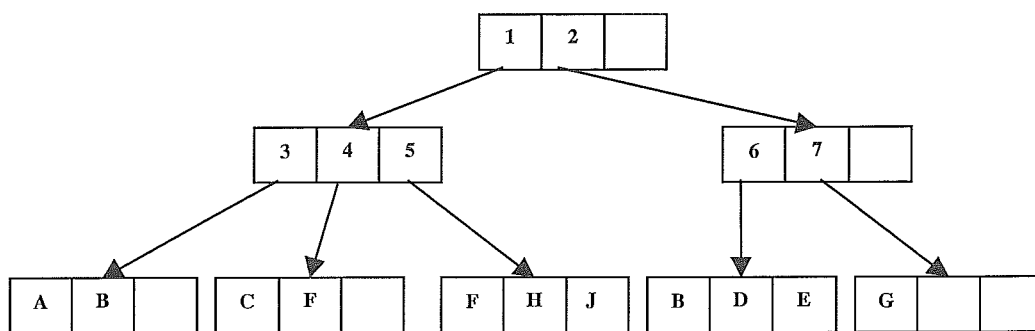
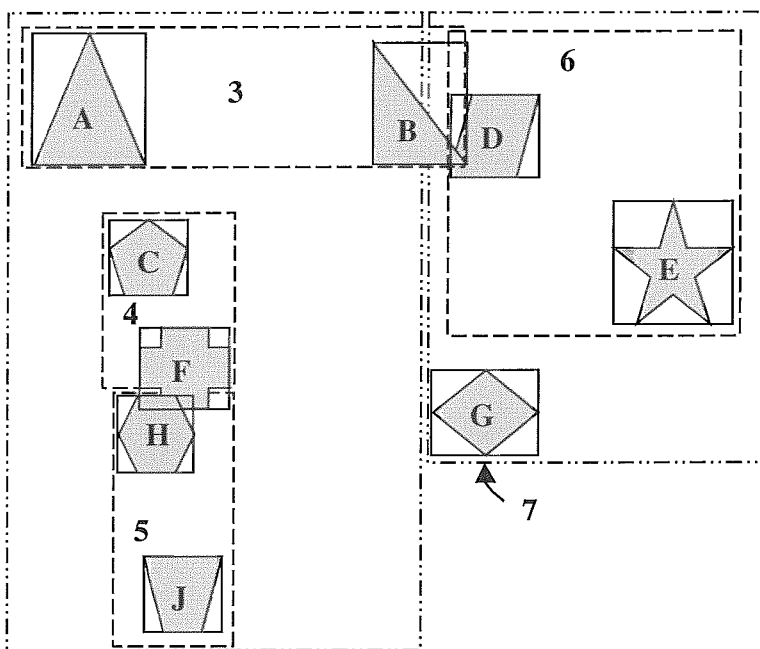


Figura 2.13 - Árvore R+

Outra variante da Árvore R é a Árvore R\*. A principal diferença entre estas duas estruturas se encontra no algoritmo de inserção de nós. Durante a inserção quando se atinge o penúltimo nível escolhe-se para inserção o nó folha que resultar em menor acréscimo de área total após a inserção. Se ocorrer empates, escolhe-se a folha com menor incremento de área. Se empatar novamente, escolher a folha com menor área.

Todas as representações de conjuntos de retângulos apresentadas são estruturas candidatas ao primeiro passo do MSQP, além de muitas outras presentes na literatura tais como: abordagens baseadas em *hash* (DEWITT et al., 1996, MING et al., 1995, 1996, ZIMBRÃO et al., 1997); algoritmos paralelos usando Árvores R\* (BRINKHOFF et al.,

1996); X-Tree (BERCHTOLD et al., 1996); separação por tamanho dos MBRs em uma árvore especial chamada de árvore de filtro (KOUDAS et al., 1997); algoritmo baseado em *plane-sweep* para realizar a junção quando nenhum índice existe sobre os conjuntos (ARGE et al., 1998); algoritmo para ser utilizado quando apenas um dos conjuntos possui índice, situação típica que ocorre quando uma consulta está encadeada com outra (MAMOULIS et al., 1999).

Como já mencionado na seção 2.3, um dos benefícios da modularização do MSQP é a possibilidade de testar e propor novas estruturas para cada um dos passos de forma independente. Assim, escolhemos, sem perda de generalidade, uma variante da Árvore R\* para a execução do primeiro passo, cujas características peculiares serão apresentadas mais à frente.

## ***2.5. Filtro geométrico***

Um filtro geométrico usa uma representação aproximada e compacta do objeto a qual guarda suas principais características. A utilização do filtro geométrico durante o processamento de consultas espaciais tem como objetivo reduzir o tempo de processamento obtendo uma resposta conclusiva através da comparação entre objetos mais simples (as aproximações dos objetos) sem ter que acessar o objeto real, o que pode ser custoso de acordo com a complexidade do mesmo.

Como exemplo, considere o filtro geométrico Retângulo Máximo Incluso ou MER (*Maximum Enclosed Rectangle*), que guarda informações do interior do objeto sendo aproximado através do maior retângulo que ele contém. A Figura 2.14 ilustra o MER aproximando uma da ilha do Hawai. Utilizando o MER, no segundo passo da arquitetura, podemos concluir que dois objetos se interceptam se as aproximações dos mesmos se sobrepõem.

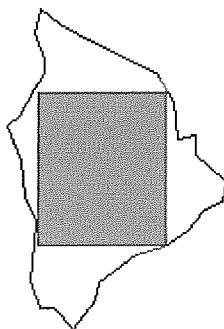


Figura 2.14 - MER para uma ilha do Hawaii.

O capítulo 3 é dedicado a maiores detalhes sobre filtros geométricos, uma vez que estes se referem ao principal assunto desta tese.

## 2.6. *Teste de geometria exata*

O terceiro passo da arquitetura de múltiplos passos é o mais custoso pois as representações reais dos objetos são comparadas, requerendo tempo de E/S para lê-las do disco e tempo de CPU para processar a interseção exata. Com o objetivo de otimizar sua execução, muitos algoritmos e estruturas espaciais foram desenvolvidos, entre os quais podemos destacar um algoritmo baseado em *scan line* (MONTEIRO et al., 2001a).

Este algoritmo foi desenvolvido com o intuito de verificar se duas polilinhas se interceptam. Possui três etapas.:

- **Primeira etapa:**

- 1) Para cada parte da polilinha faça:

- i. Ordenar os  $(n-1)$  segmentos da parte considerando o primeiro ponto de cada segmento e o sentido crescente de um dos eixos cartesianos;
- ii. Inserir os segmentos ordenados numa nova lista (formação de seqüências monotônicas);

- **Segunda etapa:**

- 2) Para cada seqüência monotônica faça:

- i. Inserir todos os segmentos num *heap* (SWARCFITER, 1994), agregando a cada um o indicador da polilinha e da sequência monotônica a qual pertence;

- **Terceira etapa:**

- 3) Criar lista vazia para armazenar elementos do *heap*, chamada de lista de comparações.
- 4) Enquanto existir elemento no *heap* faça
  - i. Retirar raiz do *heap* atribuindo a elemento atual;
  - ii. Para cada elemento da lista de comparações faça:
    - a. Se identificador de polilinha do elemento da lista é diferente do identificador de polilinha do elemento atual
      - A. Se existe interseção entre os segmentos destes elementos
        - i. **Retornar interseção como resultado** finalizando o processamento
    - iii. Se existir elemento na lista de comparações com mesmo identificador de sequência monotônica do elemento atual
      - a. Substituir este elemento por elemento atual.
    - iv. Senão
      - a. inserir elemento atual na lista de comparações

- 5) **Retornar não interseção como resultado**

O *heap*, utilizado na segunda parte do algoritmo, tem como objetivo obter ordenação com complexidade  $O(n \log n)$ . Assim, outra estrutura ou algoritmo pode ser empregado para executar a ordenação desde que seja garantida a complexidade de pior caso.



### 3. Filtros geométricos

O objetivo principal deste capítulo é apresentar os principais tipos de aproximações para filtros geométricos existentes e as dificuldades que elas possuem para a representação de polilinhas. Para uma melhor caracterização, os exemplos foram construídos baseados na junção espacial utilizando o predicado de interseção, principal tipo de junção espacial, como ressaltado por BRINKHOFF et al. (1994). Nos capítulos seguintes serão apresentadas a RIOS e a 5CDRS, propostas de melhores aproximações para polilinhas.

Filtros geométricos consistem na utilização, durante o processamento de consultas espaciais, de representações aproximadas e compactas dos objetos que armazenam suas principais características. Podemos rapidamente obter respostas conclusivas testando as aproximações, sem a necessidade de acessar as representações reais dos objetos, cuja complexidade pode resultar em um elevado custo de processamento. Com esta finalidade o filtro geométrico é utilizado no segundo passo da arquitetura MSQP, apresentado na seção 2.3. Ele permite um refinamento da consulta identificando pares que fazem parte da solução além daqueles que não se encontram neste conjunto, diminuindo assim o número testes a serem executados no terceiro passo.

A eficiência de um filtro geométrico está intimamente relacionada à aproximação por ele utilizado. Devido ao amplo uso, a maioria das aproximações referenciadas na literatura destina-se à representação de polígonos. Apesar dos bons resultados obtidos, utilizar aproximações para polígonos para representar polilinhas não tem se mostrado muito eficiente.

Este capítulo está dividido da seguinte maneira: na seção 3.1 serão ilustrados os tipos de aproximações existentes; na seção 3.2 apresentaremos a Assinatura *Raster* de Quatro cores, que é base para as aproximações para polilinhas que propomos; na seção 3.3 serão explicitados os problemas com a utilização das aproximações para polígonos para representar polilinhas; na seção 3.4, a divisão do espaço em células será apresentada, um dos principais conceitos relacionados à construção de aproximações *raster*; e na seção 3.5, um sumário do capítulo é apresentado.

### 3.1. Tipos de aproximações

Aproximações podem ser classificadas de acordo com a relação topológica que possuem com o objeto real. Segundo BRINKHOFF et al. (1993b) existem três tipos de aproximações: conservadoras, progressistas e generalizadoras.

Aproximações conservadoras contêm todo o objeto sendo representado, como ilustrado na Figura 3.1 (BRINKHOFF et al., 1993b). Na Figura 3.1.a temos o Retângulo Envoltente Mínimo ou MBR, que pode ser representado por apenas dois pontos (limite inferior esquerdo e limite superior direito). Devido a sua simplicidade, eficientes algoritmos e estruturas espaciais foram desenvolvidos empregando MBRs, os quais são amplamente utilizados no primeiro passo da arquitetura MSQP como evidenciado na seção 2.3.

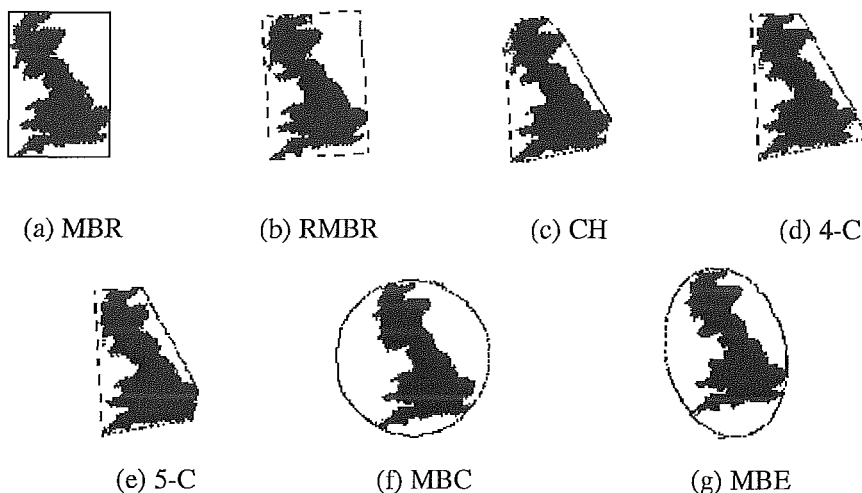


Figura 3.1 – Filtros geométricos

Existem outras aproximações conservadoras que têm a capacidade de representar polígonos de forma mais precisa do que MBR. Uma maior precisão é obtida minimizando a área coberta pela aproximação que não é preenchida pelo objeto, chamada de área falsa. Na Figura 3.1.b é apresentado o Retângulo Envoltente Mínimo Rotacionado ou RMBR (*Rotated Minimum Bounding Rectangle*) o qual é obtido adicionando um ângulo de rotação ao MBR. A Figura 3.1.c ilustra a Casca Convexa ou CH (*Convex Hull*) cuja aproximação é feita através do menor polígono convexo que contém todo o objeto. A utilização da CH pode resultar em uma aproximação tão complexa quanto o próprio objeto devido à

complexidade do polígono gerado. Com o intuito de garantir uma complexidade máxima foi desenvolvida uma aproximação limitando o número de vértices do polígono chamado de m-Cantos Envolvente Mínimo ou m-C (*m-Corner*). A Figura 3.1.d apresenta o 4-C e a Figura 3.1.e o 5-C, menor polígono com 4 e 5 vértices que envolvem o objeto respectivamente. A Figura 3.1.f ilustra o Círculo Envolvente Mínimo ou MBC (*Minimum Bounding Circle*), que aproxima o objeto através do menor círculo que o envolve. Na Figura 3.1.g é apresentada aproximação Elipse Envolvente Mínima ou MBE (*Minimum Bounding Ellipse*) que representa o objeto através da menor elipse envolvente.

Quando comparamos dois objetos através de suas aproximações conservadoras, concluímos que eles não se interceptam quando suas aproximações não se sobrepõem. Caso contrário, não podemos concluir interseção pois esta pode ocorrer nas áreas falsas. A Figura 3.2.a ilustra a identificação de não interseção entre Ruanda e o Lago Vitória (África), utilizando MBRs. No entanto, quando utilizamos o MBR para aproximar os mapas do Gabão e da bacia do Rio Congo (África), ao executarmos a comparação não conseguimos obter uma conclusão (Figura 3.2.b).

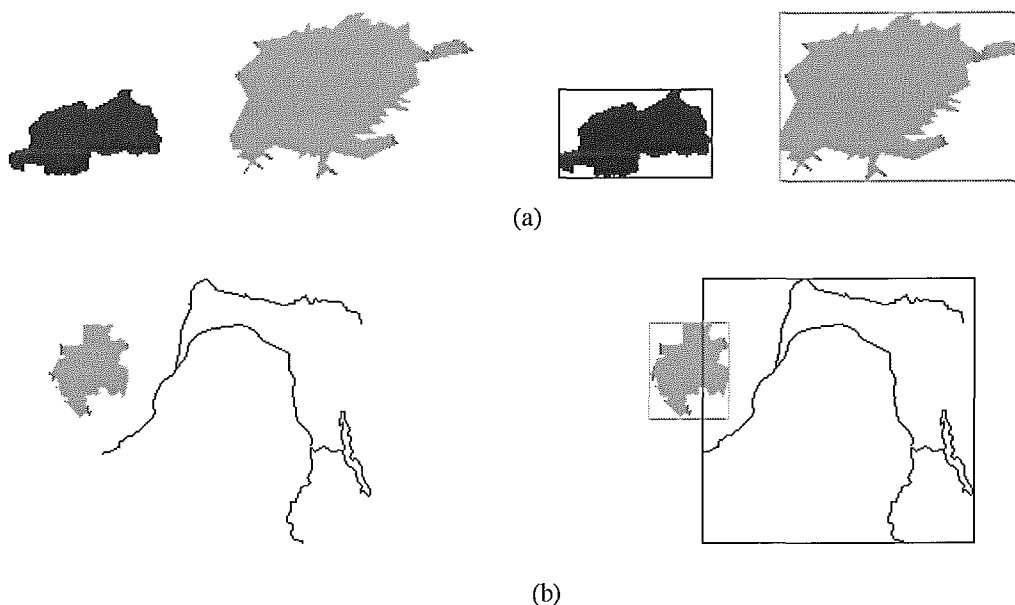


Figura 3.2 - Exemplo de comparação utilizando aproximação conservadora: (a) identificação de não interseção comparando MBRs; (b) MBR não é suficiente para identificar interseção.

Classificam-se aproximações como sendo progressistas aquelas que estão inteiramente contidas no objeto. Como exemplos temos o Retângulo Máximo Incluso ou MER (*Maximum Enclosed Rectangle*), apresentado na Figura 3.3.a, aproximando o Lago Vitória através do maior retângulo que este contém, e o Círculo Máximo Incluso ou MEC (*Maximum Enclosed Circle*), ilustrado na Figura 3.3.b, cuja aproximação é feita através do maior círculo que está inteiramente contido no objeto.

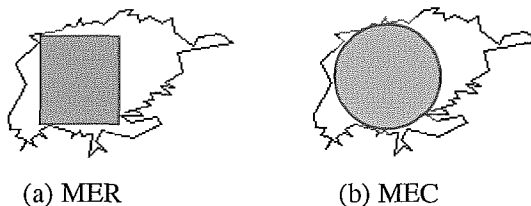


Figura 3.3 – Aproximações progressistas.

Se duas aproximações progressistas se sobrepõem, certamente os objetos se interceptam, enquanto que se o contrário acontece não podemos garantir a não interseção. A Figura 3.4.a exemplifica a identificação de interseção entre os mapas do Lago Vitória e Uganda (África), utilizando a aproximação MER. Já na Figura 3.4.b, apesar de ocorrer interseção entre o Lago Vitória e o Kênia (África), a aproximação MER não é capaz de concluir interseção.

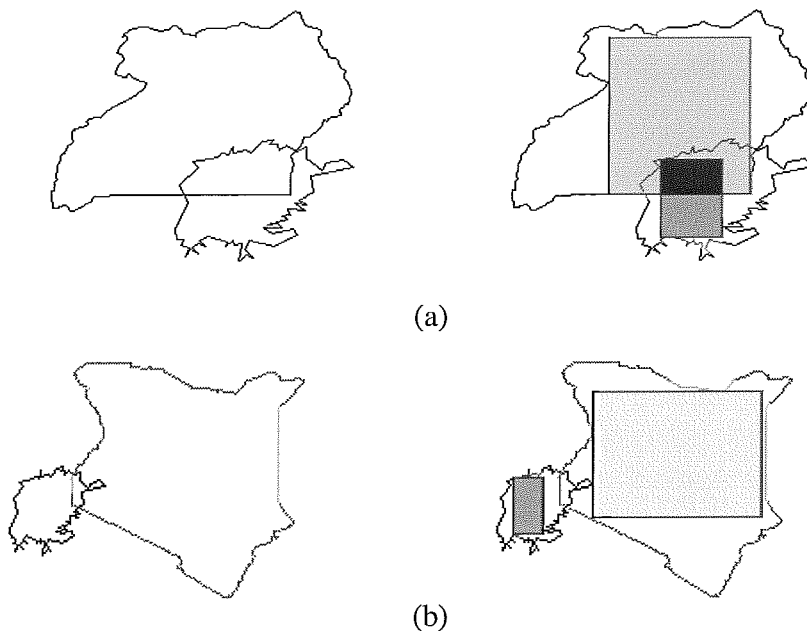


Figura 3.4 - Exemplo de comparação utilizando aproximação progressista: (a) identificação de interseção utilizando o MER; (b) MER não é suficiente para concluir interseção.

Aproximações generalizadoras são aquelas que tentam simplificar o contorno dos objetos. Em geral, não existe relação topológica entre a aproximação e o objeto real, ou seja, nem o objeto está inteiramente contido na aproximação nem a aproximação está contida no objeto. Isto faz com que a utilização de aproximações generalizadoras, em geral, não seja eficiente quando empregadas no filtro geométrico. Sua utilidade aparece, por exemplo, na apresentação de mapas. A Figura 3.5 ilustra uma aproximação generalizadora para o Lago Vitória.

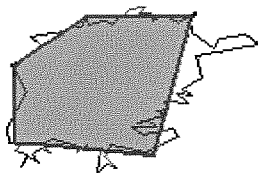


Figura 3.5 - Exemplo de aproximação generalizadora.

### 3.2. Assinatura Raster de Quatro Cores

A Assinatura *Raster* de Quatro Cores ou 4CRS (*Four Color Raster Signature*) proposta em (ZIMBRÃO,1998) é uma aproximação *raster* para representar polígonos. Apesar de ser classificada como sendo generalizadora, a 4CRS combina em uma única aproximação características das aproximações conservadoras e progressistas, tendo como consequência uma elevada eficiência quando utilizada para a construção de filtros geométricos para polígonos. Como a 4CRS se constitui na base para as abordagens das aproximações para polilinhas que serão propostas nos capítulos seguintes, dedicamos esta seção para apresentá-la. Maiores detalhes podem ser obtidos em (ZIMBRÃO, 1999).

A aproximação 4CRS é uma aproximação *raster* contendo  $m \times n$  células cada uma delas armazenando informação do percentual de área do polígono na célula. Existem quatro tipos de células possíveis (quatro cores) cada um representado por dois *bits* de informação, como ilustrado na Tabela 3.1. A Figura 3.6 apresenta a aproximação 4CRS para o Lago Vitória.

Tabela 3.1 – Valores para cada célula da aproximação 4CRS

Tipo de Célula	Representação (bits)	Descrição
Vazia	00	A célula não intercepta o polígono.
Pouco	01	A célula contém uma interseção de 50% ou menos com o polígono.
Muito	10	A célula contém uma interseção de mais de 50% com o polígono.
Cheia	11	A célula está totalmente ocupada pelo polígono.

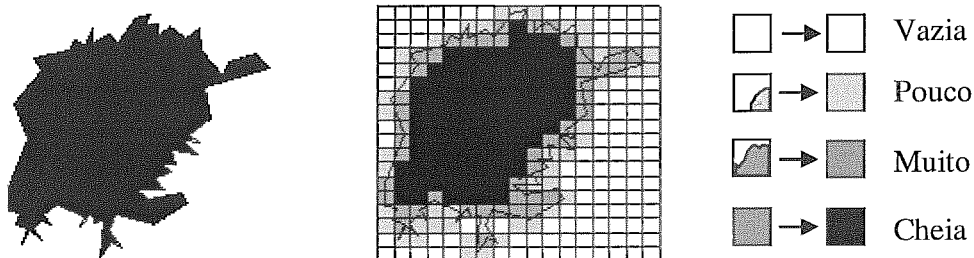


Figura 3.6 – Lago Vitória e sua aproximação 4CRS.

A 4CRS pode ser vista como sendo uma assinatura de *bits* do polígono podendo ser computada uma vez, armazenada e utilizada posteriormente. Seu pequeno tamanho permite que seja armazenada nos índices espaciais para utilização em consultas.

A verificação de interseção entre dois polígonos empregando a 4CRS é feita sobrepondo suas assinaturas e comparando cada par de células. Os resultados possíveis de comparações são apresentados na Tabela 3.2 sendo alguns casos ilustrados na Figura 3.7. Note que apenas três casos em dezesseis não permitem conclusão (Pouco x Pouco, Pouco x Muito e Muito x Pouco). Durante a comparação dos pares de células, basta obter uma resposta “Aceito” para concluir que os polígonos se interceptam. Entretanto, é necessário que todas as comparações resultem em “Rejeitado” para garantir que não existe interseção entre os polígonos. Qualquer outro resultado é inconclusivo, precisando comparar as representações reais dos objetos para alcançar uma resposta conclusiva.

Tabela 3.2 – Casos possíveis na comparação de um par de células 4CRS.

Tipo de célula × Interseção	Vazia	Pouco	Muito	Cheia
Vazia	Rejeitado	Rejeitado	Rejeitado	Rejeitado
Pouco	Rejeitado	<i>Inconclusivo</i>	<i>Inconclusivo</i>	Aceito
Muito	Rejeitado	<i>Inconclusivo</i>	Aceito	Aceito
Cheia	Rejeitado	Aceito	Aceito	Aceito

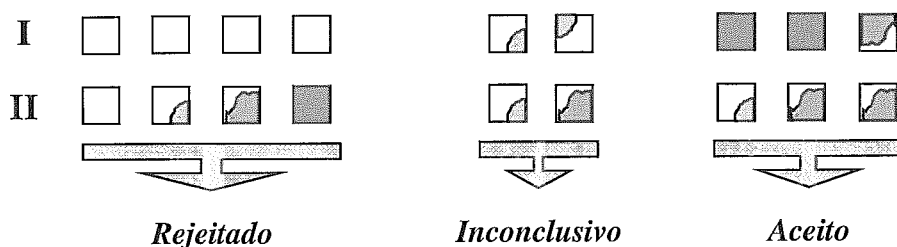


Figura 3.7 – Exemplos de comparações de células.

A 4CRS possui uma série de vantagens das quais podemos destacar a capacidade de compactação, tendo tamanho mínimo e máximo previsível; e o ótimo desempenho com o filtro para o segundo passo da arquitetura de múltiplos passos.

### *3.3. Problemas com a utilização das aproximações para polígonos na representação de polilinhas*

Aproximar polilinhas utilizando as aproximações descritas na seção anterior não se constitui na melhor abordagem possível. Um conjunto de características peculiares as polilinhas não são atendidas de forma satisfatória por aproximações especialmente desenvolvidas para polígonos. BRINKHOFF et al. (1993b) aponta três critérios que devem ser considerados para a utilização de aproximações no processamento de consultas: simplicidade, qualidade e tempo de construção. Nesta seção serão destacados os problemas resultantes da utilização das aproximações descritas na seção 3.2 na representação de polilinhas.

Uma aproximação deve ser simples, de forma que a busca e os algoritmos de teste sejam executados rapidamente. A qualidade da aproximação na representação do objeto real deve adicionar ao filtro geométrico uma precisão que permita alcançar o maior número de respostas conclusivas, tanto na identificação de pares que fazem parte do conjunto solução como aqueles que não se encontram no mesmo. O tempo de construção da aproximação deve determinar baixo custo durante a inserção ou atualização dos dados da base.

A qualidade de uma aproximação conservadora é avaliada de acordo com sua área falsa, parte da aproximação que não é preenchida pelo objeto. Quanto menor a área falsa melhor é a aproximação. Dado este fato, representar polilinhas através de aproximações conservadoras não é a melhor abordagem devido a grande área falsa obtida, como pode ser evidenciado na Figura 3.8. A Figura 3.8.a apresenta o mapa do rio Snake (Estados Unidos) e seu MBR. Observe a proporção de área falsa com relação ao espaço ocupado pelo objeto. A dificuldade de identificação de não interseção é evidenciada na Figura 3.8.b. Apesar do Rio Snake estar bastante distante da Rodovia Inter-Estadual (Estados Unidos), a pouca precisão da aproximação não permite concluir que os objetos não se interceptam. Deve ser ressaltado aqui que embora os exemplos explorem a área falsa obtida com a utilização de MBRs na representação de polilinhas, não é difícil notar que empregar outra aproximação conservadora também implicará em uma representação com elevada área falsa.



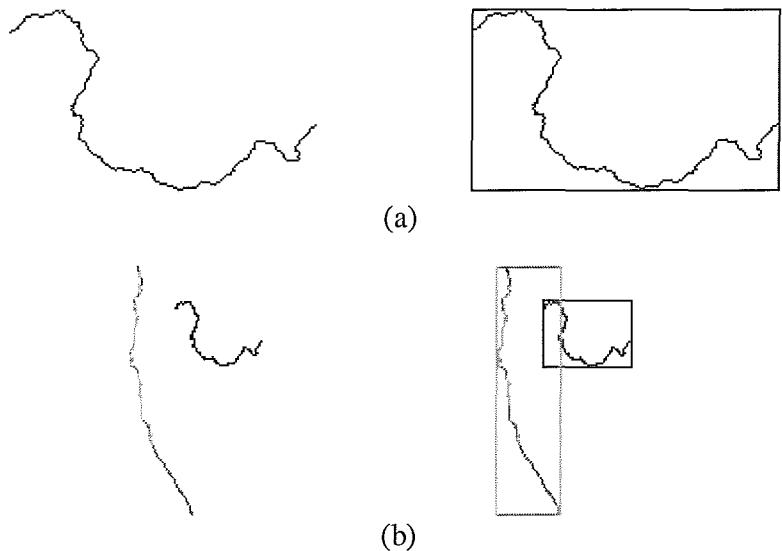


Figura 3.8 - Área falsa de aproximações conservadoras: (a) Representação de polilinha através de MBR; (b) Interseção dos MBRs das polilinhas mas não das polilinhas.

A representação de polilinhas através de aproximações progressistas também não é a mais adequada. Como, neste caso, a aproximação está inteiramente contida no objeto, a aproximação se degeneraria em um conjunto de pontos pertencentes a polilinha.

Avaliamos então a utilização de aproximações do tipo generalizadora. As qualidades da 4CRS nos inspirou em utilizá-la como ponto chave de pesquisa. Como resultado, empregar a 4CRS para aproximar polilinhas teria como resultado seus tipos de células reduzidos a dois: vazia e pouco. A aproximação de uma célula armazenaria apenas se esta é interceptada pela polilinha ou não tem interseção. Poderíamos chamar esta aproximação de Assinatura *Raster* de duas Cores ou 2CRS (*Two Color Raster Signature*) ilustrada na Figura 3.9.

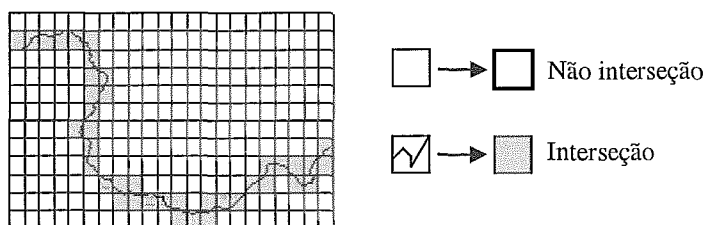


Figura 3.9 - Exemplo de aproximação 2CRS

A verificação de interseção entre duas polilinhas utilizando suas aproximações 2CRS teria não interseção como único resultado conclusivo. Mesmo comparando um par de células interceptadas não é possível caracterizar interseção. Dentro da célula as polilinhas podem descrever trajetórias que não se cruzam. Os casos possíveis para comparação são descritos na Tabela 3.2 e ilustrados na Figura 3.10. Observe que a única forma de concluir que duas polilinhas não se interceptam é obter “Rejeitado” como resultado de todas as comparações de pares de células. De outra forma, quando um resultado inconclusivo é obtido, não é mais possível concluir não interseção utilizando a aproximação. Neste caso é necessário comparar as representações reais dos objetos.

Tabela 3.2 – Casos possíveis na comparação de um par de células 2CRS.

Tipo de célula × Interseção	Não interseção	Interseção
Não interseção	Rejeitado	Rejeitado
Interseção	Rejeitado	<i>Inconclusivo</i>

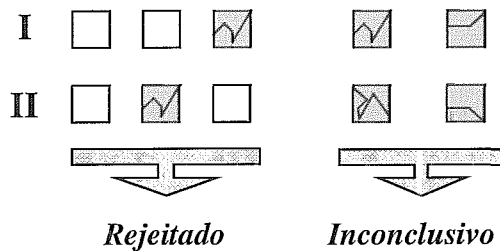


Figura 3.10 – Exemplos de comparações de células 2CRS.

Apesar do número de tipos de células para a aproximação 2CRS ser reduzido, podemos notar sua boa eficiência na representação de polilinhas. A 2CRS tem pequena área falsa, a qual ocorre apenas nas células interceptadas pela polilinha, e não é complexa, pois apenas dois tipos de células precisam ser computados. Como exemplo, é possível identificar que não há interseção entre o rio Snake e a rodovia Inter-Estadual utilizando a 2CRS (Figura 3.11). Não existem células interceptadas por ambas as polilinhas ao mesmo tempo. Devemos ressaltar aqui o fato de que as aproximações possuem células com tamanhos diferentes. Para executarmos a comparação é necessário, inicialmente, fazer uma mudança de resolução para que as células comparadas tenham o mesmo tamanho. Isto pode

ser obtido construindo-se uma nova aproximação para a de maior resolução (menor tamanho) agrupando células com o intuito de formar uma única célula de maior tamanho. Detalhes de como uma aproximação *raster* é construída com a finalidade de permitir mudança de resolução são apresentados na próxima seção.

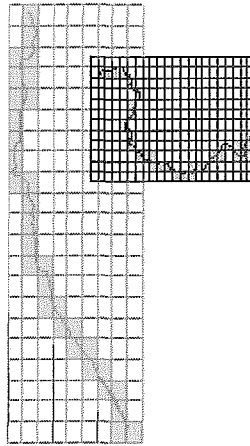


Figura 3.11 - Identificação de não interseção empregando a 2CRS

Quando utilizamos a 2CRS para verificar se o rio Sacramento tem interseção com a rodovia Inter-Estadual (Estados Unidos), o resultado é inconclusivo. Existem células atravessadas pelas duas polilinhas ao mesmo tempo (Figura 3.12) e, como evidenciado na Tabela 3.2, a verificação de interseção tem inconclusivo como resposta.

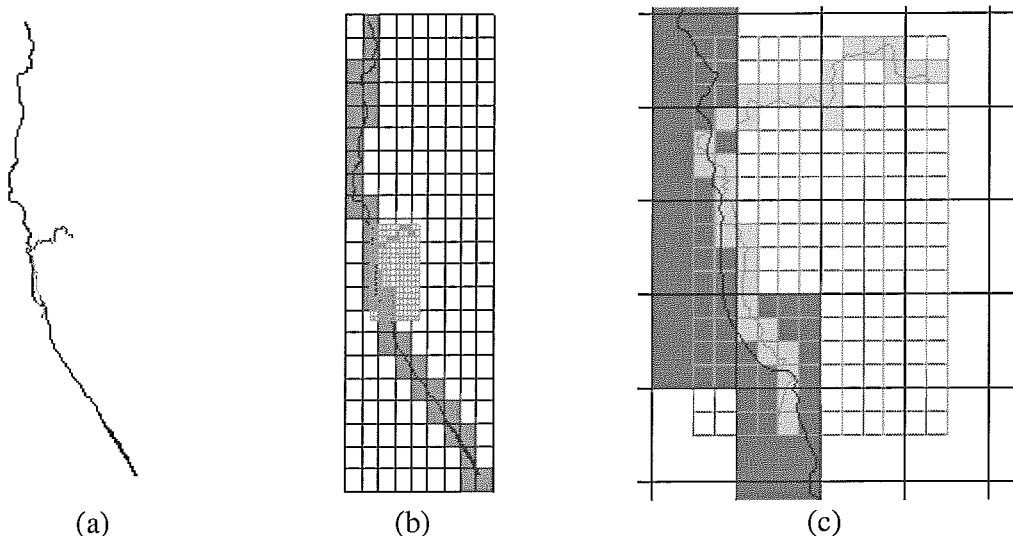


Figura 3.12 - (a) Mapa da rodovia Inter-Estadual e rio Sacramento;(b) Aproximações 2CRS; (c) Ampliação das células pertencentes ao MBR de interseção das aproximações.

A impossibilidade de concluir interseção entre polilinhas utilizando a 2CRS nos motivou a pesquisar aproximações mais precisas. Como resultado dos nossos trabalho implementamos duas propostas: a RIOS e a 5CDRS. A primeira será apresentada no capítulo 4, enquanto que a segunda no capítulo 5.

### 3.4. Divisão do espaço em células

A divisão do espaço em células, como ressaltado anteriormente, está intimamente relacionada à construção das aproximações *raster*. Em uma aproximação *raster*, cada célula armazena um número pequeno de informações do objeto extraídas de sua representação real permitindo uma representação simples e compacta. A aproximação é então empregada em filtros espaciais, otimizando o processamento das consultas, como apresentado no capítulo 2. Como características importantes destacamos que a divisão do espaço deve ser de fácil manipulação, ter número máximo de células definido, e ser capaz de, ao comparar aproximações de objetos distintos, garantir que suas relações de posicionamento sejam mantidas.

A manutenção dos posicionamentos relativos entre objetos é obtida dividindo-se o espaço globalmente enquadrando o objeto dentro das células, e não empregando uma divisão de acordo com cada um. Compactação e facilidade de manipulação são alcançadas

armazenando um número relativamente pequeno de parâmetros com informações suficientes para construir a grade rapidamente através da execução de um algoritmo simples. Finalmente, a garantia de número máximo de células é feita definindo o tamanho de cada célula da grade de acordo com a dimensão do objeto sendo representado, ou seja, objetos grandes terão células de tamanho maior do que objetos pequenos. Como consequência é necessário manipular objetos com grades em diferentes resoluções.

Todas estas características são obtidas dividindo-se o espaço em células com tamanho em potência de dois ( $2^n$ ,  $n$  inteiro) e definindo-se coordenadas para as arestas múltiplas da mesma potência na forma  $(2^na, 2^nb)$ , com  $a$  e  $b$  também inteiros. A Figura 3.13 ilustra a divisão do espaço desta maneira. Note que  $n$  varia de acordo com o tamanho da célula de cada objeto e que, com esta divisão, é possível agrupar um conjunto de células com tamanho menor para obter uma célula de maior tamanho. Tendo em mãos as coordenadas do  $MBR-2^k$  do objeto, a saber,  $e$  o tamanho da célula da grade, rapidamente esta pode ser construída.

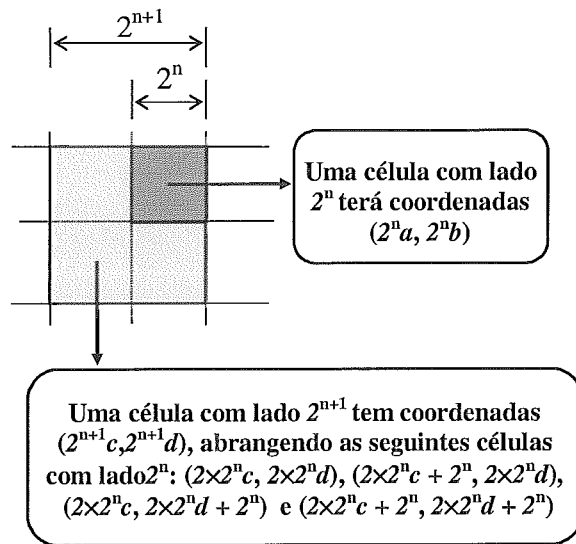


Figura 3.13 – Alinhamento dos cantos das células

O  $MBR-2^k$  do objeto corresponde a um  $MBR$  cujos vértices são da forma  $(2^ka_0, 2^kb_0)$  e  $(2^ka_m, 2^kb_n)$ , onde  $a_0, a_m, b_0, b_n$  e  $k$  são inteiros.  $k$  é escolhido de tal modo que  $(a_m - a_0)(b_n - b_0) < k + 1$ , sendo  $k$  o número máximo de células da grade de  $m \times n$  células, fixado no momento da geração. O  $MBR-2^k$  é calculado com base no  $MBR$  do polígono da seguinte

forma: inicialmente o maior  $k$  é escolhido de forma que  $2^k$  seja menor ou igual ao lado do MBR original. A partir daí, em repetições sucessivas,  $k$  é diminuído de 1 enquanto as condições anteriores valerem. Por fim,  $a_0$  e  $b_0$  são os maiores inteiros satisfazendo  $2^k a_0 \leq x$  mínimo e  $2^k b_0 \leq y$  mínimo do MBR original. Analogamente,  $a_m$  e  $b_n$  são os menores inteiros satisfazendo  $2^k a_m \geq x$  máximo e  $2^k b_n \geq y$  máximo do MBR original.

Os pontos  $2^k a_0, 2^k a_1, \dots, 2^k a_m$  definem um conjunto de linhas paralelas ao eixo vertical que chamaremos de  $X_0, X_1, \dots, X_m$ . De modo semelhante, definimos as linhas horizontais  $Y_0, Y_1, \dots, Y_n$ .

### 3.5. Sumário

Este capítulo apresentou os principais tipos de aproximações para filtros geométricos existentes e as dificuldades que elas possuem para a representação de polilinhas. Tais dificuldades nos motivaram à pesquisa de aproximações mais eficientes. Como resultado implementamos as aproximações *raster* RIOS e 5CDRS que serão apresentadas nos capítulos seguintes. Na seção 3.4 apresentamos a divisão do espaço em células, uma das principais características para construção de aproximações *raster*.

## 4. RIOS

Este capítulo apresenta a Assinatura *Raster* Usando Posições de Entrada e Saída ou RIOS (*Raster I/O Signature*), a primeira proposta desta tese, capaz de representar polilinhas de uma forma mais eficiente para o processamento de consultas espaciais do que as aproximações apresentadas no capítulo 3. Essa aproximação foi estudada e apresentada no relatório técnico (ZIMBRÃO et al., 2001).

A RIOS foi desenvolvida tendo como base a aproximação 4CRS, apresentada na seção 2.5, com o objetivo de identificar tanto interseções como não interseções no processamento de junção de polilinhas. Ao compararmos as aproximações *raster* de duas polilinhas, concluímos que elas não se interceptam quando elas não se cruzarem em nenhuma célula de seus mapas *raster* atravessadas por ambas ao mesmo tempo. Por outro lado, basta ocorrer interseção em uma única célula para concluir que as polilinhas se interceptam. Todavia, decidir que existe interseção não é uma tarefa fácil. Dentro da célula as polilinhas podem se dispor em várias trajetórias e em nenhuma delas se cruzarem.

A aproximação RIOS armazena, para cada célula da grade *raster*, as posições de interseção da polilinha com as bordas da célula, chamadas de posições de entrada e saída, para verificar se duas polilinhas se interceptam. A Figura 4.1.b exemplifica as posições de entrada e saída de uma célula atravessada por duas polilinhas, na qual 0 e 6 são as posições da primeira polilinha, enquanto que 2 e 11 são as posições da segunda. A RIOS baseia-se no fato de que se duas polilinhas se interceptam então, em geral, os segmentos definidos pelas posições de entrada e saída das mesmas com relação à célula que atravessam devem se cruzar. A única exceção ocorre quando os segmentos são idênticos, ou seja, na célula considerada, têm as mesmas posições de entrada e saída. Neste caso não podemos garantir que existe interseção.

As posições de entrada e saída são obtidas dividindo cada lado da célula em  $n$  partes iguais, onde  $n$  é potência de dois ( $2^m$ ). Cada parte é numerada, segundo o sentido horário, de 0 a  $4n - 1$ , começando na extremidade superior esquerda da célula. Na Figura 4.1 é apresentado, em linhas gerais, um exemplo de como as posições de entrada e saída podem ser empregadas para determinar se duas polilinhas se interceptam. Na Figura 4.1.a temos

uma célula que é atravessada por duas polilinhas. A Figura 4.1.b ilustra a divisão das arestas, com número de divisões igual a quatro, e suas respectivas posições de entrada e saída. Concluímos que as polilinhas têm interseção quando os segmentos definidos pelas posições de entrada e saída das células se cruzam (Figura 4.1.c).

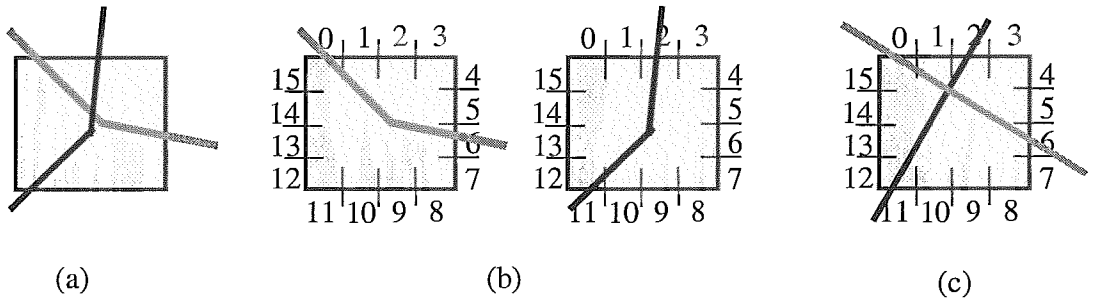


Figura 4.1 -Exemplo de divisões da aresta para células: (a) célula interceptada por duas polilinhas; (b) divisão das arestas para cada polilinha; (c) verificação de interseção.

Neste capítulo apresentaremos na seção 4.1 a composição da aproximação RIOS. Na seção 4.2, a sua geração será apresentada. Na seção seguinte iremos estudar o mapa de posições candidatas, estrutura utilizada durante o cálculo da aproximação. A seção 4.4 descreve o que é realizado para comparar aproximações em diferentes resoluções. Na seção 4.5 está o algoritmo para comparação enquanto que na seção 4.6 é descrita a forma de armazenamento. Finalmente, na seção 4.7, um sumário do capítulo é apresentado.

#### 4.1. Composição da assinatura

A assinatura de cada polilinha é composta por um cabeçalho e um mapa de *bits*. O cabeçalho possui informações de identificação, localização e resolução da grade (Tabela 4.1). O mapa de *bits* corresponde às aproximações RIOS para cada célula armazenando as seguintes informações (Tabela 4.2):

- Indicador de interseção da polilinha com a célula;
- Indicador de posições de entrada e saída válidas;
- Posição de entrada da polilinha na célula; e,
- Posição de saída da polilinha na célula.



Tabela 4.1 – Estrutura do cabeçalho para RIOS.

Bits (bytes)	Descrição	Tipo
32 (4)	Identificador da polilinha	Inteiro longo sem sinal
32 (4)	X mínimo ( MBR da polilinha )	Inteiro longo sem sinal
32 (4)	Y mínimo ( MBR da polilinha)	Inteiro longo sem sinal
32 (4)	X máximo ( MBR da polilinha)	Inteiro longo sem sinal
32 (4)	Y máximo ( MBR da polilinha)	Inteiro longo sem sinal
12 (1,5)	Nº Células no eixo X	Inteiro sem sinal
12 (1,5)	Nº Células no eixo Y	Inteiro sem sinal
8 (1)	Tamanho da célula representada em potência de 2	Inteiro sem sinal
<b>Total: 192 (24)</b>		

Tabela 4.2 – Estrutura da aproximação RIOS para uma célula.

Bits	Descrição	Tipo
1	Indicador de interseção da célula com a polilinha	Booleano
1	Indicador de assinatura válida	Booleano
m	Posição de entrada	Inteiro sem sinal
m	Posição de saída	Inteiro sem sinal
<b>Total: 2 + 2 M</b>		

O primeiro *bit* indica se a polilinha passa pela célula. O segundo *bit* indica se foi possível gerar posições de entrada e saída válidas. Os *bits* seguintes armazenam os respectivos valores das posições de entrada e saída.

O número de *bits*, utilizado para armazenar as posições de entrada ou saída, é obtido a partir do número de divisões de cada aresta. Seja  $n$  este número, a possibilidade para representação de uma posição é igual a  $4n$ , pois cada um dos quatro lados da célula é dividido  $n$  vezes. O valor de  $n$  é determinado como sendo potência de dois. O objetivo é garantir o maior número de divisões possíveis, de acordo com o número de *bits* utilizados para a representação. Isto permite uma melhor utilização do espaço de armazenamento. O número de *bits* para representar uma posição pode ser calculado como sendo igual a  $\log_2 4n$ . Se o número de divisões de cada aresta é por exemplo igual a 8, teremos  $4 \times 8 = 32$  possibilidades que podem ser representadas em  $\log_2 32 = 5$  *bits*. A Tabela 4.3 apresenta a aproximação para a célula da Figura 4.1.b interceptada pela polilinha com entrada e saída nas posições 0 e 6.

Tabela 4.3 – Exemplo de aproximação RIOS.

Valor inteiro (bits)	Descrição
1 (1)	Polilinha intercepta a célula
1 (1)	Assinatura é válida
0 (0000)	Posição de entrada
6 (0110)	Posição de saída
<b>Total: 10 bits</b>	

Constituindo-se uma aproximação *raster*, a RIOS armazena informações relevantes para cada célula da grade de seu mapa. Assim, quanto maior o número de células maior o espaço em disco ocupado pela aproximação mas melhor é sua precisão. Obviamente, diminuir o número de células implica em ter uma aproximação mais distante da representação real do objeto. Aliada a resolução da grade, outro fator preponderante na definição do tamanho da aproximação é a quantidade de informações armazenadas, neste caso, estamos considerando o número de *bits* utilizados para armazenar as posições de entrada e saída. Assim, para garantir o tamanho máximo é de suma importância definir tanto um limite para o número de células da grade como também para a divisão de cada aresta de cada célula.

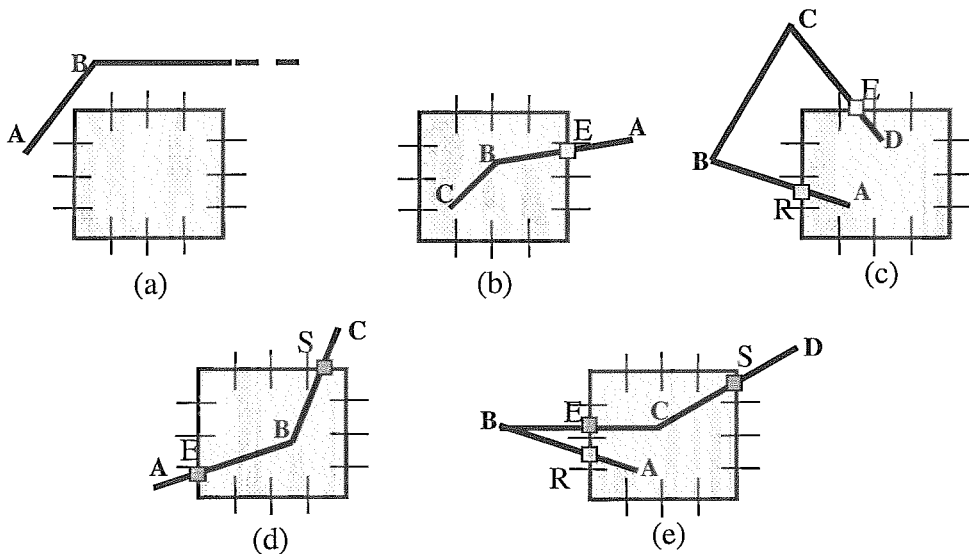
#### 4.2. Geração da aproximação

O algoritmo de geração da aproximação consiste em percorrer todas as células de uma grade de células armazenando as posições de entrada e saída referentes às interseções da polilinha com as bordas de cada célula. A seguir descreveremos os passos desse algoritmo. Maiores detalhes serão apresentados posteriormente para facilitar um melhor entendimento.

- 6) Obter  $MBR-2^n$  da parte da polilinha com base no MBR da mesma, conforme descrito na seção 3.4;
- 7) Gerar grade com  $p \times q$  células com arestas de tamanho igual a  $2^n$  de acordo com o  $MBR-2^n$ .
- 8) Para cada célula da grade calcular a aproximação da seguinte forma:
  - i. Se a célula não for atravessada pela polilinha:

- a. Ajustar aproximação da célula para *Não existe interseção*;
- ii. Se a célula for atravessada pela polilinha mas não existirem dois pontos de interseção tal que o caminho de um ponto ao outro esteja totalmente contido na célula:
  - a. Ajustar aproximação da célula para *Existe interseção e as posições de entrada e saída não são válidas*;
- iii. Se existirem dois pontos de interseção da polilinha com a célula tal que o caminho de um ponto ao outro esteja totalmente contido dentro da célula:
  - a. Ajustar aproximação da célula para *Existe interseção, as posições de entrada e saída são válidas, e armazenar as respectivas posições de entrada e saída*.

O cálculo de posições de entrada e saída é feito computando-se interseções a partir do primeiro ponto localizado fora da célula. Assim, conseguimos garantir que o trecho de interseção esteja inteiramente contido na mesma. Como exemplo, considere a Figura 4.4. Na Figura 4.4.a é ilustrado o caso trivial onde a célula não é atravessada pela polilinha. As Figuras 4.4.b e 4.4.c demonstram o caso em que não é possível determinar posições válidas. Na primeira, a interseção é computada a partir do ponto *A* e é definido apenas um ponto de interseção *E*. Na outra Figura, existem dois pontos de interseção mas o caminho descrito de um ponto ao outro não está inteiramente contido na célula. A obtenção de interseções válidas para determinação de posições de entrada e saída é ilustrada nas Figuras 4.4.d e 4.4.e. A célula da primeira Figura tem os pontos *E* e *S* como interseções calculadas a partir do ponto *A*. Enquanto que na Figura 4.4.e, o cálculo inicia-se a partir de *B* (primeiro ponto fora da célula), gerando como resultado os pontos *E* e *S*. Observe que se o cálculo fosse realizado a partir de *A*, *R* e *E* seriam escolhidos incorretamente como pontos de interseção válidos.



■ Interseção computada

□ Interseção ignorada

Figura 4.4– Exemplo do cálculo de posições de entrada e saída.

A partir dos pontos de interseção calculados, determinam-se as posições de entrada e saída. Para cada ponto é computado um deslocamento de acordo com a coordenada inferior esquerda da célula, a aresta de interseção e o tamanho de cada divisão de aresta. A posição de entrada ou saída é definida considerando a aresta interceptada e o deslocamento. Abaixo são apresentados os cálculos realizados para cada aresta seguidos de exemplos. Considere as seguintes definições:

- T: tamanho de cada aresta da célula,  $T=2^n$ , n inteiro sem sinal, cujos detalhes se encontram na seção 3.4 ("Divisão do espaço em células");
- k: número de divisões de cada aresta,  $k = 2^m$ , m inteiro sem sinal;
- t: tamanho de cada uma das k divisões de aresta,  $t = T/k = 2^{(n-m)}$ , t real sem sinal;
- p: posição de entrada ou saída sendo calculada, inteiro sem sinal variando de 0 a  $4k - 1$ ;
- P ( $P_x, P_y$ ): ponto de interseção;
- M ( $M_x, M_y$ ): ponto inferior esquerdo da célula;
- d: deslocamento, inteiro sem sinal.

Como exemplo, considere os valores para  $T = 8$  e  $k = 4$ , como consequência  $t = 2$ ;  $M = (32,40)$ .

– Fórmula para aresta superior:

- $d = \lfloor (P_x - M_x) / t \rfloor$
- $p = d$

Na Figura 4.5 temos um exemplo com  $P = (35,48)$ .

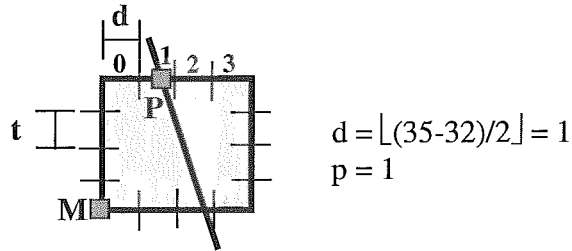


Figura 4.5 – Cálculo da posição de entrada.

– Fórmula para aresta da direita:

- $d = \lfloor (P_y - M_y) / t \rfloor$
- $p = 2k - d - 1$

A Figura 4.6 exemplifica o cálculo com  $P = (40,43)$ .

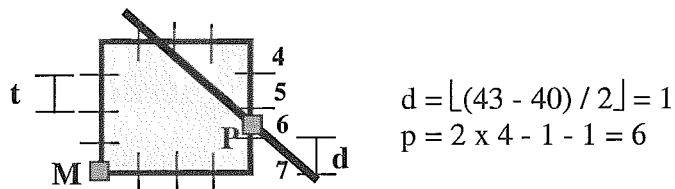


Figura 4.6 – Cálculo da posição de saída.

– Fórmula para aresta inferior:

- $d = \lfloor (P_x - M_x) / t \rfloor$
- $p = 3k - d - 1$

Na Figura 4.7 temos um exemplo com  $P = (37,40)$ .

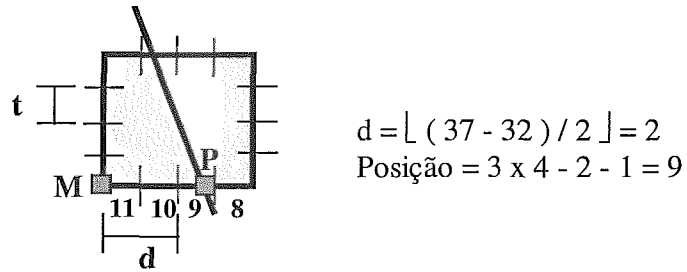


Figura 4.7 – Cálculo da posição de saída.

– Fórmula para aresta da esquerda:

- $d = \lfloor (P_y - M_y) / t \rfloor$
- $p = 3k + d$

A Figura 4.8 apresenta um exemplo do cálculo com  $P = (32, 45)$ .

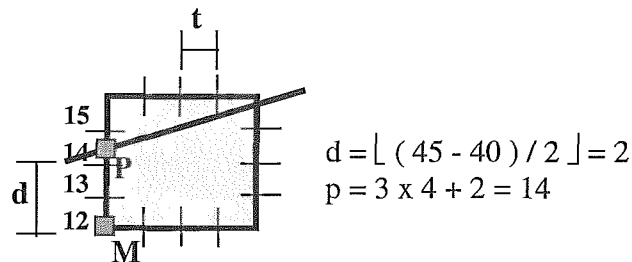


Figura 4.8 – Cálculo da posição de saída.

Um caso que deve ser ressaltado é a interseção da polilinha com os cantos da célula. Cada canto pode ser considerado como pertencente a duas arestas. Entretanto, apenas uma posição de entrada ou saída pode ser computada. Adotamos, sem perda de generalidade, a convenção apresentada na Tabela 4.4 e ilustrada na Figura 4.9. O objetivo foi obter o maior número possível de posições que mantivessem equivalência em arestas de células diferentes, conceito que será detalhado na próxima seção. Calcular entrada e saída em arestas adjacentes é extremamente útil durante a mudança de resolução, a ser apresentada na seção 4.4. Como resultado, as interseções que ocorrem com os cantos da esquerda têm posições em arestas adjacentes. Já interseções com os cantos da direita produzem posições válidas em arestas não adjacentes.

Tabela 4.4 – Convenção para interseção com os cantos.

Cantos	Convenção
Inferior esquerdo	Primeira posição à direita.
Superior esquerdo	Primeira posição à direita.
Inferior direito	Primeira posição acima.
Superior direito	Primeira posição abaixo.

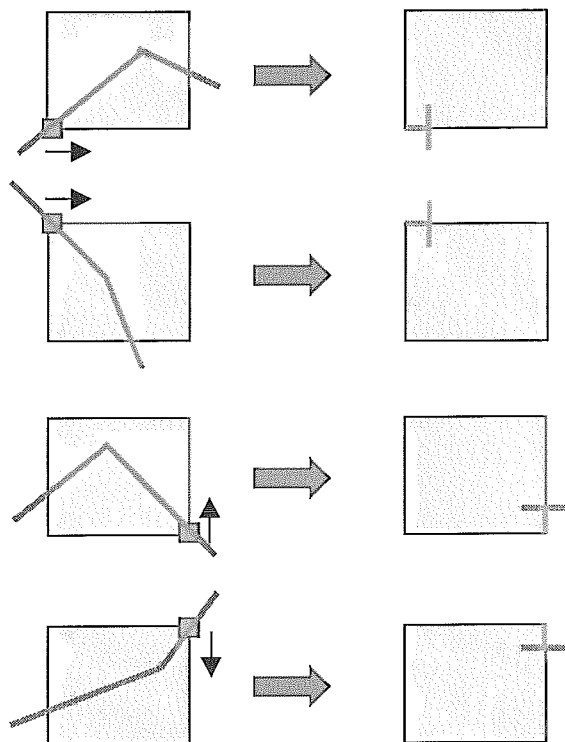


Figura 4.9 - Convenção para interseção com o canto da célula.

Na próxima seção será explicado o mapa de posições candidatas. Este é utilizado durante a geração da aproximação, com o objetivo de obter células aproximadas por partes contínuas da polilinha, fator preponderante para realizar mudanças de resoluções, que será apresentada na seção 4.4.

### 4.3. Mapa de posições candidatas

O mapa de posições candidatas é utilizado durante a geração da RIOS com o intuito de produzir aproximações usando partes contínuas da polilinha. O objetivo principal é minimizar a quantidade de informações perdidas quando são necessárias mudanças de resolução para comparar aproximações em resoluções diferentes. Para um melhor

entendimento do conceito de mapas de posições candidatas, primeiramente serão definidos posições correspondentes e posição candidata.

**Posições correspondentes** referem-se às posições de células vizinhas que ocupam o mesmo lugar no espaço. Na Figura 4.10 é apresentado um exemplo de posições correspondentes. Considerando que o número de divisões de cada aresta é igual a quatro, a posição 4 da célula à esquerda (aresta direita) é correspondente à posição 15 da célula da direita (aresta esquerda).

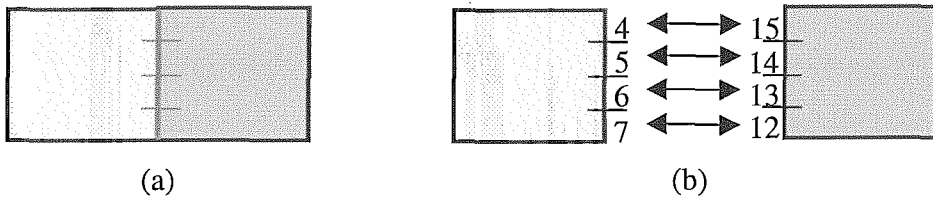


Figura 4.10 - Posições correspondentes para células com número de divisões das arestas igual quatro

**Posição candidata** é uma possível posição válida para uma célula, que ainda não teve aproximação calculada. Esta é posição de entrada ou saída para uma célula vizinha em sua posição correspondente. A Figura 4.11 ilustra posição candidata. Ao calcular a aproximação para a célula da esquerda, 5 é escolhida como posição de saída, enquanto que a célula adjacente, à sua direita, tem ajustado 14 como posição candidata.

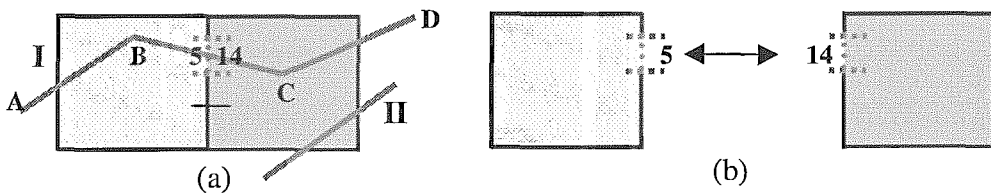


Figura 4.11 - Exemplo de posições candidatas

O **mapa de posições candidatas** armazena as posições candidatas de cada célula juntamente com os segmentos e as partes que as geraram. Na Figura 4.11, para a célula da direita, armazenaríamos que a posição 14 é uma posição candidata gerada pelo segmento BC da parte I. Utilizando o mapa de posições candidatas, assegura-se que células vizinhas que possuam posições de entrada ou saída possíveis em posições correspondentes, tenham estas escolhidas como posições válidas apenas se estiverem na parte contínua da polilinha e não por outra parte (Figura 4.12). Caso não seja possível definir uma aproximação



utilizando a posição candidata, esta é marcada como inválida. Impede-se assim que outra parte da polilinha a utilize, o que poderia quebrar a continuidade. Na Figura 4.12.a a aproximação é gerada utilizando a posição candidata, resultando em (intercepta, válida, 5, 14). A Figura 4.12.b expõe o caso em que a posição 14 é marcada como inválida por não ser possível gerar uma aproximação contínua utilizando-a com o segmento da parte I. Apesar da parte II ter posições de entrada e saída com a célula, a aproximação não pode ser gerada pois utilizaria a posição 14 que é inválida. A aproximação, então, se resume a (intercepta, inválida, -, -). A Figura 4.12.c exemplifica o caso em que a posição candidata é inválida mas não restringe o cálculo das posições de entrada e saída em outros pontos. A célula então seria aproximada como (intercepta, válida, 3, 8).

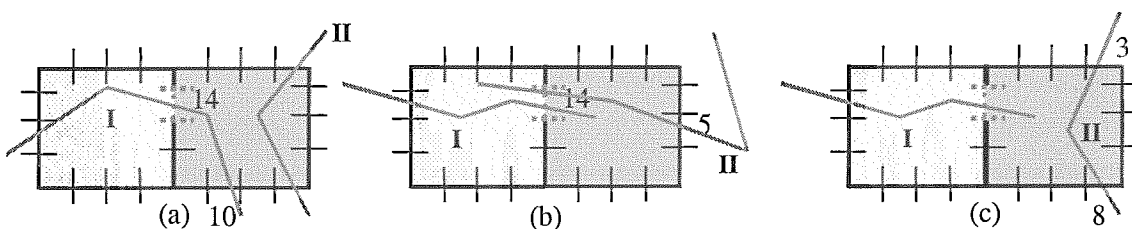


Figura 4.12 - Determinação de posições de entrada e saída: (a) a partir do segmento que gerou a posição candidata é possível gerar posições válidas; (b) não é possível gerar posições válidas com a posição candidata. (c) não é possível gerar posição válida com a posição candidata mas existem outras posições válidas.

#### 4.4. Mudança de resolução

De acordo com o algoritmo de geração, seção 4.2, polilinhas diferentes podem ter grades de células em diferentes resoluções pois o tamanho das células está intimamente relacionado ao limite do número máximo de células que a grade pode possuir. Por outro lado, para verificar se duas polilinhas se interceptam, é necessário que as aproximações a serem testadas tenham o mesmo tamanho de célula e que células que se sobreponham tenham mesmas coordenadas. O algoritmo de divisão do espaço em células, relatado na seção 3.4, garante tais características, com o tamanho das células sendo uma potência de dois ( $2^n$ ) e suas coordenadas múltiplas da mesma potência na forma  $(a2^n, b2^n)$ .

Para obter células de mesmo tamanho poderíamos pensar em duas abordagens: dividir as células da aproximação com maior tamanho diminuindo seu lado; ou agrupar células de menor tamanho. O primeiro caso não se constitui numa boa alternativa pois não é

possível definir posições de entrada e saída para células menores utilizando apenas as posições da célula maior. Neste caso, as aproximações para as células atravessadas pela polilinha se resumiriam a (intercepta, inválida, -, -). Como a possibilidade de comparação entre aproximações em diferentes resoluções é alta, muita informação relevante seria perdida durante a mudança de escala, além de perdermos a capacidade de identificar interseção. A alternativa adotada foi agrupar um conjunto de células com menor tamanho para formar uma célula maior. Para que este objetivo fosse alcançado com o mínimo de perda possível foi desenvolvido o mapa de posições candidatas, descrito na seção anterior.

O algoritmo para mudança de resolução consiste em percorrer as posições de entrada e saída que são equivalentes, nas células que serão agrupadas, tentando formar uma sequência que tenha início e fim em arestas da nova célula. Se esta sequência for obtida uma nova aproximação com posições válidas de entrada e saída é gerada. Caso contrário, a aproximação será (intercepta, inválida, -, -). A Figura 4.13 ilustra ambos os casos. Na Figura 4.13.a, seguindo as posições de entrada e saída que são equivalentes (setas), é possível determinar as posições da nova célula. Na Figura 4.13.b, a descontinuidade da polilinha nas células que serão agrupadas não permite gerar aproximação com posições válidas.

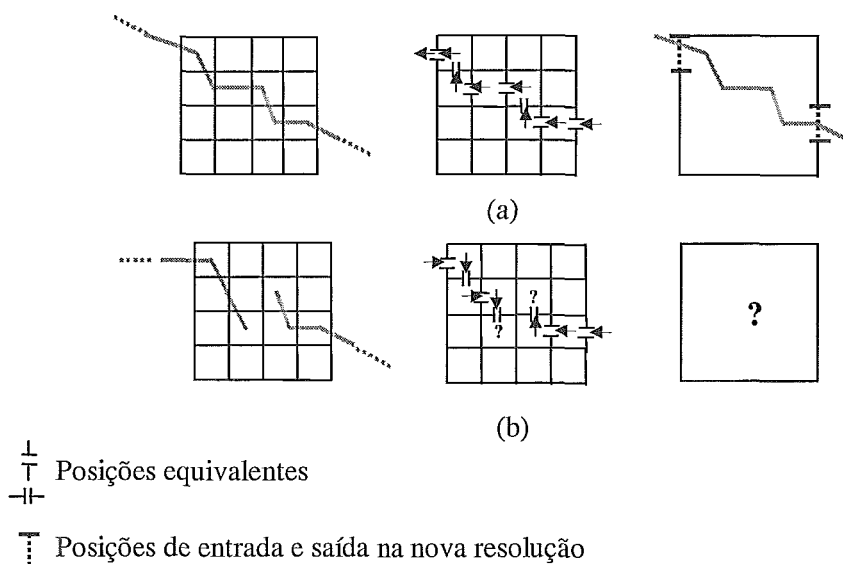


Figura 4.13 – Exemplos de mudança de resolução: (a) agrupamento de células com novas posições de entrada e saída; (b) posições equivalentes não formam sequência, não é possível determinar posições válidas.

O cálculo das posições de entrada e saída é feito de forma semelhante à geração das posições da aproximação, como descrito na seção 4.2. Entretanto, agora o cálculo é feito sobre posições da célula menor para gerar posições para uma célula maior, ao contrário de utilizar coordenadas referentes aos pontos de interseção. Posições são calculadas de acordo com a aresta interceptada e um deslocamento, o qual considera características da célula menor. Considere as seguintes definições:

- $T'$ : tamanho de cada aresta da célula menor,  $T'=2^{n'}$ ,  $n'$  inteiro sem sinal;
- $T''$ : tamanho de cada aresta da célula maior,  $T''=2^{n''}$ ,  $n''$  inteiro sem sinal;
- $L$ : número de arestas da célula maior que cabem em uma aresta da célula maior,  $L = T'' / T' = 2^{(n''-n')}$ , inteiro sem sinal.
- $k$ : número de divisões de cada aresta da célula,  $k = 2^m$ ,  $m$  inteiro sem sinal;
- $t'$ : tamanho de cada uma das  $k$  divisões de aresta da célula menor,  $t' = T' / k = 2^{(n'-m)}$ , número real;
- $t''$ : tamanho de cada uma das  $k$  divisões de aresta da célula maior,  $t'' = T'' / k = 2^{(n''-m)}$ , número real;
- $p'$ : posição de entrada ou saída da célula menor, inteiro sem sinal variando de 0 a  $4k - 1$ ;
- $p''$ : posição de entrada ou saída a ser calculada, inteiro sem sinal variando de 0 a  $4k - 1$ ;
- $P(i, j)$ : posição da célula menor no agrupamento de células;

Como ilustração, considere que a aresta da célula maior é 4 vezes maior do que a aresta da célula menor e que cada divisão de aresta ( $k$ ) é igual a 8. Com isto, é necessário um agrupamento de 4 x 4 células para formar uma única célula maior, como na Figura 4.14. A Figura 4.14.a ilustra o agrupamento de células. Na Figura 4.14.b a ampliação da célula menor localizada na posição P (3,3) do agrupamento é destacada e a Figura 4.14.c ilustra os dados da célula maior.

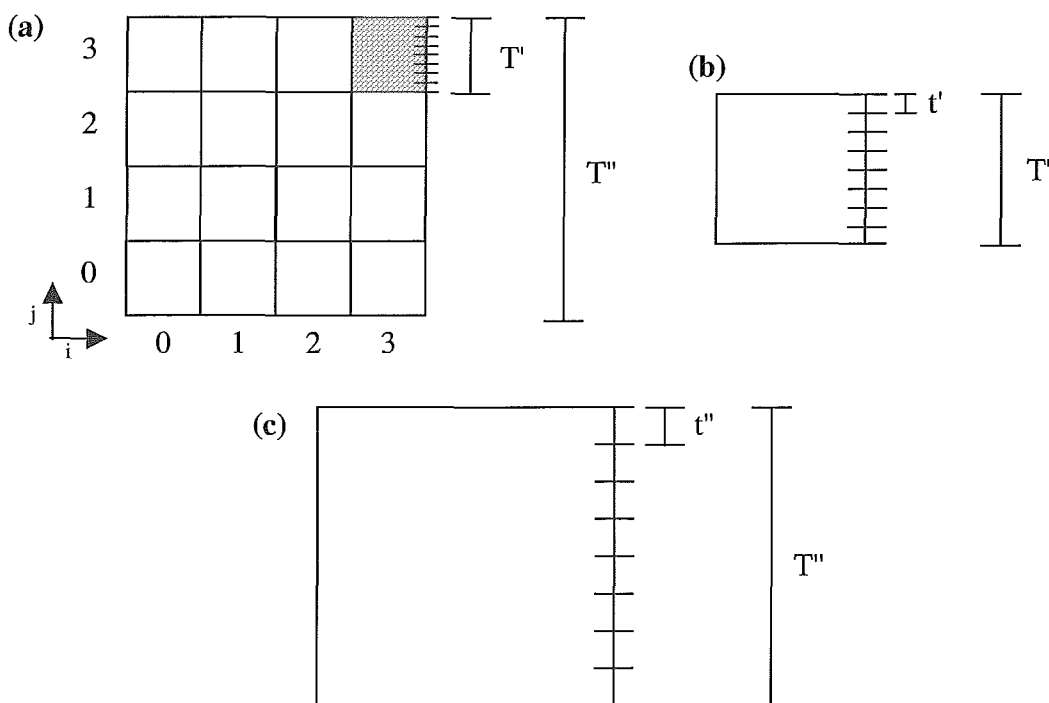


Figura 4.14 - Parâmetros para mudança de resolução: (a) agrupamento de 4 x 4 células menores para formar uma célula maior; (b) ampliação da célula menor, tamanhos da aresta e da divisão da mesma destacados; (c) nova célula, tamanhos da aresta e da divisão da mesma em destaque.

A fórmula de cálculo da nova posição é semelhante para cada uma das quatro arestas da célula maior. De acordo com a aresta considerada, calcula-se o deslocamento correspondente ao canto superior esquerdo da célula maior até a posição da célula menor e, em seguida, divide-se pelo tamanho de divisão de aresta da célula maior. Note que o cálculo das posições de entrada e saída da célula maior considera as células menores adjacentes às bordas da célula maior.

Considere como exemplo  $T' = 4$ ,  $T'' = 16$  e  $k = 8$ . Como consequência teremos  $L = 16/4 = 4$ ,  $t' = 4/8 = 0,5$  e  $t'' = 16/8 = 2$ .

– Fórmula para aresta superior:

- $p'' = \lfloor (i \cdot T' + p' \cdot t') / t'' \rfloor$

- $i$  : número inteiro de arestas menores deslocadas

- $i \cdot T'$ : deslocamento inteiro em tamanho de arestas menores até a célula onde a interseção ocorre.

- $p' \cdot t'$ : deslocamento na célula em questão

Na Figura 4.15 é apresentado um exemplo de mudança de resolução para a aresta superior com posição anterior igual a 5 na célula (2,3) do agrupamento.

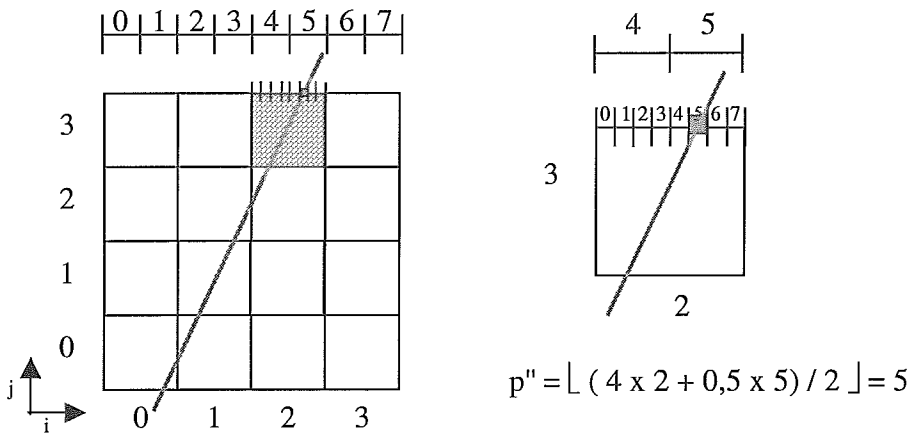


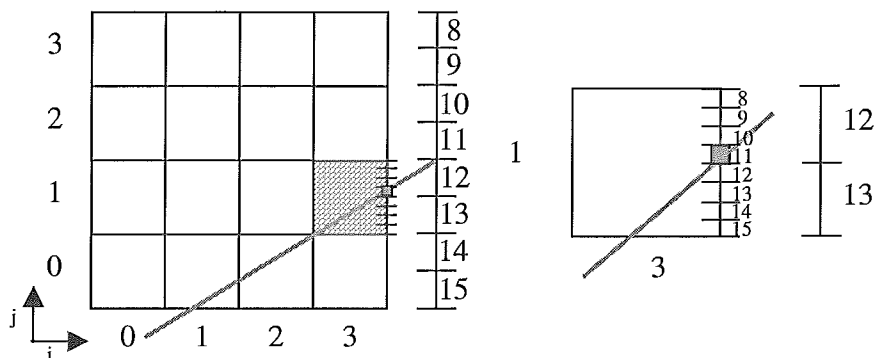
Figura 4.15 - Exemplo de cálculo de posição para aresta superior com mudança de resolução.

– Fórmula para a aresta da direita

- $p'' = \lfloor \{ T'' + T' \cdot [L - (j + 1)] + (p' - k) \cdot t' \} / t'' \rfloor$

- $L - (j + 1)$ : número inteiro de arestas menores deslocadas na aresta maior da direita. Observe que como o deslocamento é calculado de cima para baixo e  $j$  começa em 0 e cresce de baixo para cima, o deslocamento é o complemento de  $(j+1)$ .
- $T'' + T' \cdot [L - (j + 1)]$ : deslocamento inteiro em tamanho de arestas menores até a célula onde a interseção ocorre.
- $(p' - k)$ : número inteiro de posições deslocadas na aresta da direita da célula menor.
- $(p' - k) \cdot t'$ : deslocamento na aresta da direita da célula menor

Na Figura 4.16 é apresentado um exemplo de mudança de resolução para a aresta da direita com posição anterior igual a 11 na célula (3,1) do agrupamento.



$$p'' = \lfloor \{ 16 + 4 \times [4 - (1+1)] + (11 - 8) \times 0,5 \} / 2 \rfloor = \lfloor 25,5/2 \rfloor = 12$$

Figura 4.16 - Exemplo de cálculo de mudança de resolução para a aresta da direita.

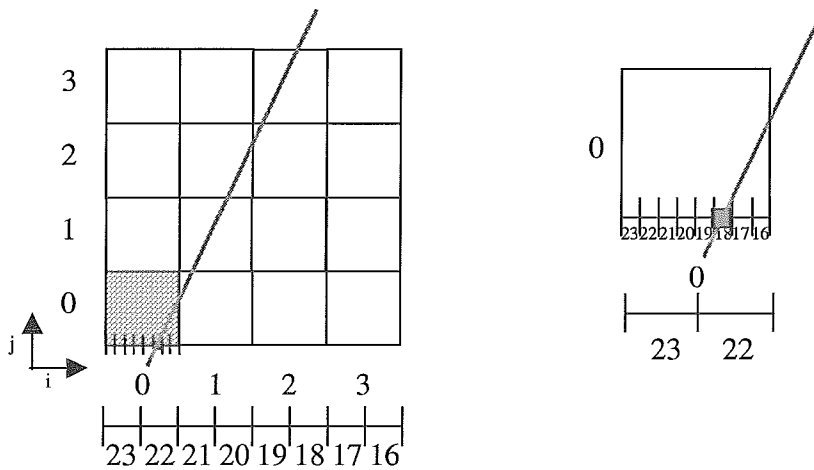
– Fórmula para a aresta de baixo

- $p'' = \lfloor \{ 2 T'' + T' \cdot [L - (i + 1)] + (p' - 2 k) \cdot t' \} / t'' \rfloor$

- $L - (i + 1)$ : número inteiro de arestas menores deslocadas na aresta maior de baixo. Observe que como o deslocamento é calculado da direita para a esquerda e  $i$  começa em 0 e cresce da esquerda para a direita, o deslocamento é o complemento de  $(i + 1)$ .

- $2 T'' + T' \cdot [L - (i + 1)]$ : deslocamento inteiro em tamanho de arestas menores até a célula onde a interseção ocorre.
- $(p' - 2 k)$ : número de posições inteiras deslocadas na aresta de baixo da célula menor.
- $(p' - 2 k) \cdot t'$ : deslocamento na aresta de baixo da célula menor.

Na Figura 4.17 é apresentado um exemplo de mudança de resolução para a aresta inferior com posição anterior igual a 18 na célula (0,0) do agrupamento.



$$p'' = \lfloor \{ 2 \times 16 + 4 \times [4 - (0 + 1)] + (18 - 2 \times 8) \cdot 0,5 \} / 2 \rfloor = \lfloor 45 / 2 \rfloor = 22$$

Figura 4.17 - Exemplo de cálculo de mudança de resolução para a aresta de baixo.

– Fórmula para a aresta da direita

- $p'' = \lfloor (3 T'' + j \cdot T' + (p' - 3 k) \cdot t') / t'' \rfloor$

- $j$  : número inteiro de arestas menores deslocadas na aresta da esquerda.
- $3 T'' + j \cdot T'$ : deslocamento inteiro em tamanho de arestas menores até a célula onde a interseção ocorre.

- o  $(p' - 3k)$ : número inteiro de posições deslocadas na aresta da esquerda da célula menor.
- o  $(p' - 3k) \cdot t'$ : deslocamento na aresta da esquerda da célula menor.

Na Figura 4.18 é apresentado um exemplo de mudança de resolução para a aresta da esquerda com posição anterior igual a 29 na célula (0,2) do agrupamento.

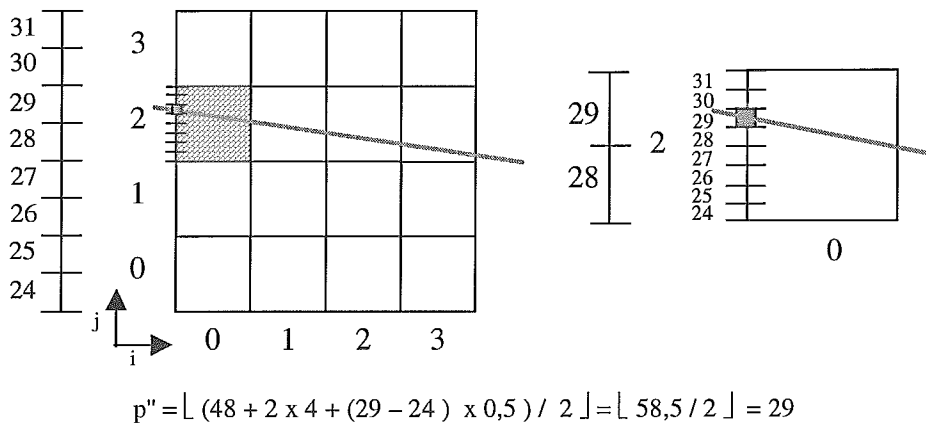


Figura 4.18 - Exemplo de cálculo de mudança de resolução para a aresta da direita.

#### 4.5. Algoritmo de comparação

A verificação se duas polilinhas se interceptam, testando suas aproximações RIOS, deve ser realizada no MBR de interseção de seus MBRs. Apenas serão testadas as células das grades das aproximações pertencentes as duas polilinhas ao mesmo tempo. Além disso, as células comparadas devem estar na mesma resolução. Sendo assim pode ser necessário realizar a mudança de resolução na aproximação com menor tamanho de célula antes de executar a comparação. Outro fato a ser ressaltado é que, durante a comparação das células, assim que um resultado “Aceito” é alcançado podemos parar o teste e afirmar que as polilinhas se interceptam. A Tabela 4.5 apresenta os resultados possíveis ao compararmos um par de células. Abaixo segue o algoritmo de comparação:

- 1) Calcular o MBR de interseção dos MBRs das polilinhas (Figura 4.19.b);
- 2) Se os tamanhos das células das aproximações forem diferentes



- i. Aplicar a mudança de resolução nas células do MBR de interseção da aproximação com menor tamanho de célula;
- 3) Inicializar resultado como *Rejeitado*;
- 4) Para cada par de células do MBR de interseção faça;
- i. Se uma das células é inconclusiva e a outra é inconclusiva ou tem posições de entrada e saída válidas
    - a. Ajustar resultado como *Inconclusivo*;
  - ii. Senão Se as duas células têm posições de entrada e saída válidas
    - a. Testar as posições para verificar interseção
      - A. Se existir interseção
        - i. Ajustar resultado como *Aceito*;
        - ii. Sair do laço
      - B. Senão
        - i. Ajustar resultado como *Inconclusivo*;
- 5) Retornar resultado como resultado final da comparação;

Tabela 4.5 - Tipos possíveis ao compararmos um par de aproximações RIOS.

Tipo de Célula X Interseção	Vazia	Inconclusiva	Posições de E/S
Vazia	Rejeitado	Rejeitado	Rejeitado
Inconclusiva	Rejeitado	Inconclusivo	Inconclusivo
Posições E/S	Rejeitado	Inconclusivo	<i>Comparar posições</i>

A fórmula de comparação de posições de entrada e saída pode ser descrita da seguinte maneira:

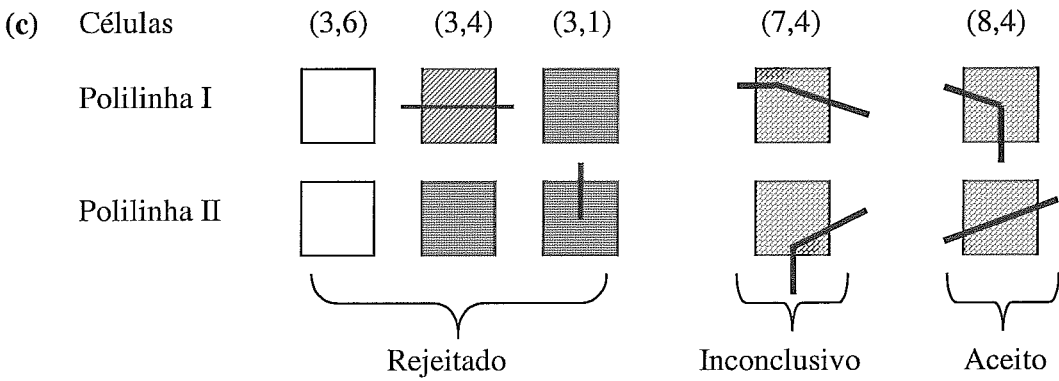
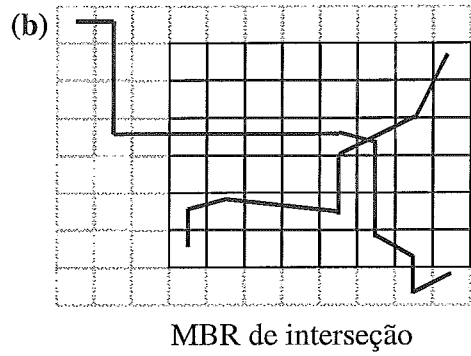
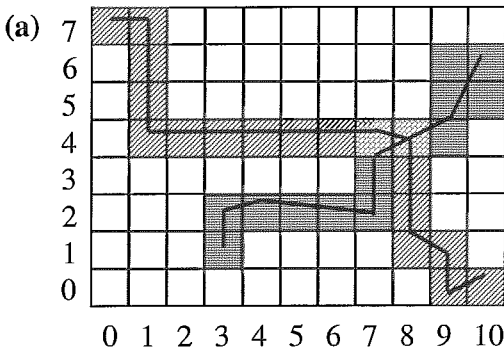
Seja  $E_1, S_1$  e  $E_2, S_2$  as posições de entrada e saída de duas células,  $E_1 \leq S_1$  e  $E_2 \leq S_2$ ; a numeração das posições segue o sentido horário.

**Se  $((E_2 > E_1) \text{ e } (E_2 < S_1) \text{ e } (S_2 > S_1))$  Então Aceito**

**Senão Se  $((E_2 < E_1) \text{ e } (S_2 > E_1) \text{ e } (S_2 < S_1))$  Então Aceito**

**Senão Inconclusivo.**

A Figura 4.19 ilustra a comparação realizada entre duas polilinhas quando é utilizada a RIOS como aproximação. A Figura 4.19.a apresenta as polilinhas e suas grades destacando as células interceptadas, a numeração dos eixos é utilizada para explicar exemplos de comparações. A Figura 4.19.b mostra o MBR de interseção, apenas as células que ele contém serão comparadas. A Figura 4.19.c exemplifica a comparação entre tipos diferentes de células. Ao compararmos, por exemplo, o par de células localizadas na posição (3,6), que são do tipo vazia para as duas aproximações, o resultado obtido é rejeitado. Mesmo resultado é alcançado quando comparamos uma célula com posições de entrada e saída válidas a uma célula vazia, como ocorre na posição (3,4); ou uma célula vazia e uma inconclusiva, posição (3,1). Quando é executada a comparação para o par de células da posição (7,4) o resultado é inconclusivo, as posições de entrada e saída das aproximações não são suficientes para garantir que existe ou não interseção. Ao compararmos as posições de entrada e saída do par de células da posição (8,4) concluímos interseção. Neste momento pode-se parar a comparação e concluir que as polilinhas se interceptam.



4.19 - Exemplo de comparação.

## 4.6. Armazenamento

O armazenamento é feito de forma a guardar apenas as informações necessárias para ler a aproximação do disco para a memória. Inicialmente são armazenados os dados do cabeçalho (Tabela 4.1.). Em seguida, a aproximação referente a cada célula da grade é armazenada. Em memória, o número de *bits*  $n$  necessário para armazenar a aproximação de uma célula é igual a  $2 + 2m$  *bits* como descrito na Tabela 4.2., onde  $m = \log_2 4k$  e  $k$  é o número de divisões de cada aresta. Quando a célula não é interceptada por polilinha, armazena-se apenas um *bit* com valor zero. Se a polilinha atravessar a célula e não for possível calcular as posições de entrada e saída dois *bits* são armazenados. O primeiro com valor um, indicando interseção, e o segundo com valor zero, indicando que as posições de entrada e saída são inválidas. Caso contrário, todos os *bits* são armazenados: interseção (1), válida (1) e as posições de entrada e saída ( $2m$ ). Para armazenar, por exemplo, 8 células vazias é necessário apenas 1 *byte*, 1 *bit* por aproximação.

## 4.7. Sumário

Este capítulo apresentou a Assinatura *Raster* Usando Posições de Entrada e Saída (RIOS), uma das propostas desta tese de representação para polilinhas a ser utilizada em índices geométricos durante o processamento de consultas espaciais. No capítulo 6 serão destacadas as vantagens e desvantagens em empregar a RIOS como aproximação. Entre as vantagens destacamos a capacidade de identificação de interseção devido ao armazenamento das posições de entrada e saída da polilinha para as células atravessadas pela mesma. Uma desvantagem é o fato do tamanho da aproximação poder ser elevado devido às informações serem armazenadas para todas as células da grade e ao número de divisões de aresta para armazenamento das posições de entrada e saída.

As desvantagens da RIOS nos motivaram à pesquisa de uma nova aproximação, a Assinatura *Raster* Direcional de Cinco Cores (5CDRS), outra proposta desta tese para aproximar polilinhas, a qual apresentamos no capítulo seguinte.

O capítulo 6 foi então dedicado a apresentar resultados experimentais e a exibir comparações entre as aproximações RIOS e 5CDRS.

## 5. 5CDRS

Este capítulo apresenta em detalhes a Assinatura *Raster* Direcional de Cinco Cores ou 5CDRS (*Five Colors Directional Raster Signature*), uma proposta de aproximação para polilinhas a ser utilizada no processamento de consultas espaciais. Esta aproximação foi desenvolvida em conjunto com MONTEIRO (2001b) e originou o artigo (ZIMBRÃO et al., 2000) que foi submetido, aprovado e apresentado no XV Simpósio Brasileiro de Banco de Dados de 2000 além do relatório técnico (MONTEIRO et al., 2001a).

A 5CDRS foi desenvolvida tendo como base a aproximação 4CRS para polígonos (ZIMBRÃO, 1998) com o objetivo de ser capaz de identificar tanto interseção como também não interseção durante o processamento da junção de polilinhas. A 5CDRS é uma variação da aproximação RIOS apresentada na seção anterior. Ela atende de maneira muito mais eficiente aos requisitos de espaço de armazenamento quando comparada com a aproximação RIOS, tendo grande poder de compactação. Esta compactação é obtida reduzindo-se a quantidade de informações armazenadas, o que traz como consequência um menor poder de identificação de interseção entre polilinhas. Todavia, a 5CDRS mostrou-se muito eficiente quando aplicada no segundo passo da arquitetura MSQP. Resultados experimentais apresentados em (MONTEIRO et al., 2001a) demonstram que mais de 50% dos pares candidatos são identificados neste passo não sendo necessário avaliar o predicado com as representações reais dos objetos no passo seguinte. No próximo capítulo serão apresentados resultados experimentais, além de comparações entre a 5CDRS e a RIOS.

Assim como a RIOS, a aproximação 5CDRS é construída sobre uma grade de  $m \times n$  células armazenando uma assinatura de *bits* para as mesmas. Ao contrário da RIOS, a 5CDRS guarda informações apenas para as células atravessadas pela polilinha e não para todas. São armazenadas as direções de deslocamento da polilinha nas células da grade. Tal fato permite uma grande compactação. Resultados experimentais são apresentados no capítulo 6. A Figura 5.1 apresenta uma polilinha formada por duas partes, e seus vetores de deslocamento. De 1 a 8 estão numerados os vetores da primeira parte e de 1 a 6 os da segunda. Note que os tipos possíveis de vetores são: cima, baixo, direita e esquerda.

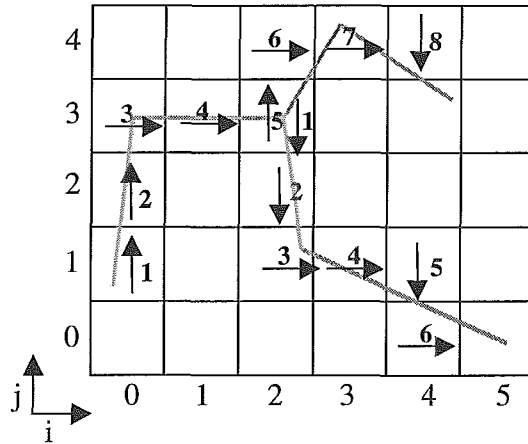


Figura 5.1 - Polilinha e seus vetores de deslocamento

Neste capítulo veremos na seção 5.1 a composição da aproximação. Na seção 5.2 apresentaremos sua geração. O armazenamento será apresentado na seção 5.3 enquanto que na seção 5.4 o algoritmo de comparação das aproximações será estudado. Na seção 5.5 é apresentado o algoritmo para mudança de resolução. Finalizaremos, apresentando um sumário do capítulo na seção 5.6.

### 5.1. Composição da aproximação

A aproximação 5CDRS é formada por um cabeçalho e um mapa de *bits*, da mesma forma que a RIOS. O cabeçalho possui informações de identificação, localização, resolução da grade e tamanho da aproximação. O mapa de *bits* corresponde às aproximações para cada célula da cadeia atravessada pela polilinha.

A aproximação é construída seguindo o caminho descrito pela polilinha na grade. Um vetor é gerado para cada deslocamento. Isto traz como consequência a necessidade de identificar qual vetor deve ser armazenado quando uma bifurcação é atingida. A solução encontrada foi construir uma aproximação independente para cada parte da polilinha, eliminando bifurcações. Tal fato é evidenciado na Figura 5.1. Caminhando a partir da célula (0,1), ao atingir a célula (2,3) um ponto de ramificação é identificado. Neste momento, escolhe-se continuar para cima, enquanto que o deslocamento para baixo dá origem a outro caminho independente. É interessante aqui ressaltar que os objetos reais, que foram utilizados como fonte para resultados experimentais (capítulo 6), estavam

armazenados de forma semelhante. Cada parte é formada por um conjunto de segmentos contínuo, sem ramificações. Pontos comuns a partes diferentes mantêm a conexão entre elas. Tais fatos facilitaram a concepção de um algoritmo para aproximar cada parte.

A geração de aproximações independentes para cada parte levantou outra questão importante: se as aproximações das partes deveriam ser armazenadas juntas constituindo-se numa representação global da polilinha, ou seria melhor empregar uma concepção local, armazenando cada aproximação de parte de forma independente juntamente com o identificador da polilinha.

Inicialmente foi concebida a aproximação de caráter global, armazenando as representações de todas as partes da polilinha. Como o número de partes é variável foi adicionado ao cabeçalho um novo campo para armazenar seu valor, o qual está destacado na segunda linha da Tabela 5.1.

Tabela 5.1 – Estrutura do cabeçalho

Bits (bytes)	Descrição	Tipo
32 (4)	Identificador da polilinha	Inteiro longo sem sinal
32 (4)	<b>Número de partes</b>	<b>Inteiro longo sem sinal</b>
32 (4)	X mínimo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	Y mínimo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	X máximo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	Y máximo (MBR da polilinha)	Inteiro longo sem sinal
12 (1,5)	Nº de células no eixo X	Inteiro sem sinal
12 (1,5)	Nº de células no eixo Y	Inteiro sem sinal
8 (1)	Tamanho da célula representada em potência de 2	Inteiro sem sinal
<b>Total: 224 (28)</b>		

A aproximação de cada parte é composta pela posição (i,j) da célula da grade onde está localizado seu ponto inicial (Figura 5.1), número de vetores e os vetores propriamente ditos (Tabela 5.2). Como existem quatro tipos diferentes de vetores (cima, baixo, direita e esquerda) a representação de cada um é formada por dois *bits* (Tabela 5.3). O número de *bits* N necessários para armazenar a aproximação da parte pode ser calculado pela fórmula  $N = 24 + 2n$ , onde n é o número de vetores da aproximação.

Tabela 5.2 – Estrutura da assinatura de cada parte

Bits	Descrição	Tipo
8	Localização horizontal da célula do ponto inicial	Inteiro sem sinal
8	Localização vertical da célula do ponto inicial	Inteiro sem sinal
8	Número de vetores ( $n$ )	Inteiro sem sinal
$n^2$	Vetores de deslocamento	
<b>Total: <math>24+n^2</math></b>		

Tabela 5.3 – Vetores de deslocamento

Bits	Descrição
00	Cima
01	Direita
10	Baixo
11	Esquerda

Contudo, o armazenamento global mostrou-se inadequado, tanto o número de partes que uma polilinha pode conter é variável quanto o número total de vetores de cada aproximação de parte. Isto dificulta muito o controle do tamanho máximo das aproximações, característica imprescindível para o armazenamento das assinaturas em estruturas de índices espaciais. Cada assinatura de parte pode conter no mínimo vinte e quatro (24) *bits* considerando que esta é formada por apenas uma célula não possuindo vetores. O número de *bits* necessário para armazenamento da célula inicial e o número de vetores são fixos (vinte e quatro *bits*). Estes parâmetros podem ser considerados como sendo o cabeçalho da assinatura da parte. Desse modo, o número máximo de assinaturas de partes que poderiam ser armazenadas seria igual ao número de *bits* correspondente ao tamanho máximo possível dividido pelo tamanho do cabeçalho da parte. Além disso, quanto maior o número de partes, menor o espaço para armazenamento dos vetores, diminuindo a precisão da representação. Como resultado perderíamos a independência entre as aproximações, já que o aumento do tamanho de uma implica na diminuição da outra para manutenção de critérios que restringem o espaço de armazenamento. Como consequência uma nova estrutura de armazenamento foi desenvolvida.

A aproximação da polilinha foi então reformulada para armazenamento local e não global, ou seja, armazenar a aproximação de uma parte de forma independente das demais, e não a da polilinha como um todo. Com isto o cabeçalho foi alterado, o indicador da parte sendo representada e o tamanho da assinatura da mesma foram adicionados enquanto que o campo para armazenamento do número de partes foi removido (Tabela 5.4). A estrutura da



assinatura de *bits* utilizada para guardar a aproximação da parte não foi alterada. Com esta nova representação foi obtido um controle maior sobre o tamanho, a única variação possível concentrou-se na quantidade de vetores gerados. Este, por sua vez, pode ser facilmente controlado diminuindo a resolução da grade, ou seja, quando o número de vetores ultrapassa o limite de espaço, o tamanho da célula da grade é aumentado e a aproximação é recalculada.

Tabela 5.4– Estrutura do cabeçalho

Bits (bytes)	Descrição	Tipo
32 (4)	Identificador da polilinha	Inteiro longo sem sinal
8(1)	Identificador da parte	Inteiro longo sem sinal
32 (4)	Tamanho da assinatura	Inteiro sem sinal
32 (4)	X mínimo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	Y mínimo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	X máximo (MBR da polilinha)	Inteiro longo sem sinal
32 (4)	Y máximo (MBR da polilinha)	Inteiro longo sem sinal
12 (1,5)	Nº Células no eixo X	Inteiro sem sinal
12 (1,5)	Nº Células no eixo Y	Inteiro sem sinal
8 (1)	Tamanho da célula representada em potência de 2	Inteiro sem sinal
<b>Total: 232 (29)</b>		

Além disso, quando a aproximação é global, o MBR armazenado no cabeçalho corresponde ao MBR da polilinha, enquanto que a assinatura local armazena o MBR de cada parte. Na maioria das vezes o segundo é bem menor do que o primeiro, no pior caso o MBR da parte será igual ao MBR da polilinha. Como consequência a representação por partes permite diminuir a área de indecisão melhorando a capacidade de decisão do primeiro passo, na arquitetura MSQP. Outra vantagem é que a construção de aproximações independentes permite que o tamanho da célula da grade seja diferente para cada parte da polilinha, ocasionando representações mais precisas em maiores resoluções. A Figura 5.3.a apresenta uma polilinha dividida em três partes e a grade de células para uma aproximação de caráter global. A Figura 5.3.b ilustra a grade definida para cada parte da polilinha.

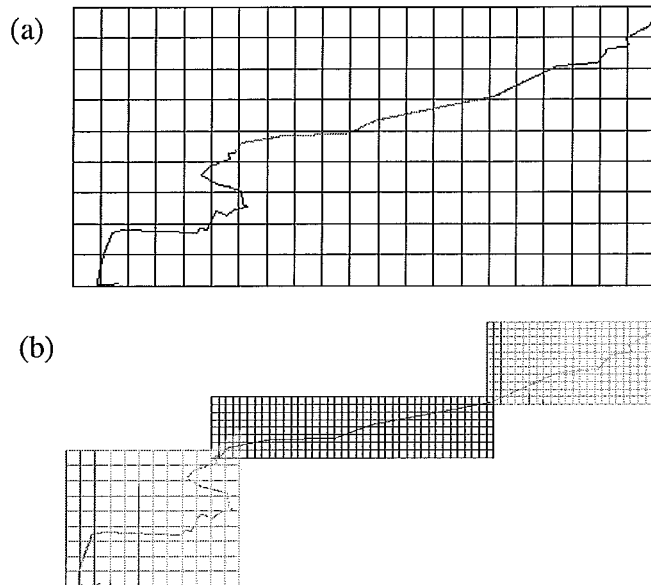


Figura 5.3 - Polilinha com grade para cada uma de sua parte

Maior precisão também é obtida no conjunto resposta da consulta, ao invés deste ser composto pelas polilinhas que se interceptam, ele será formado pelas partes das polilinhas que têm interseção.

## 5.2. Geração da aproximação

Uma assinatura de *bits* é computada para cada parte da polilinha, como ressaltado na seção anterior. A seguir descreveremos o algoritmo para geração da aproximação. Maiores detalhes serão apresentados posteriormente para facilitar um melhor entendimento. Observe que o algoritmo deve ser executado para todas as partes da polilinha, pois uma aproximação é gerada para cada parte.

- 1) Obter  $MBR-2^n$  da parte da polilinha com base no MBR da mesma, conforme descrito na seção 3.4;
- 2) Gerar grade com  $p \times q$  células com arestas de tamanho igual a  $2^n$  de acordo com o  $MBR-2^n$ ;
- 3) Determinar a célula da grade onde se localiza o primeiro ponto da parte;

- 4) Marcar primeiro ponto como ponto atual, próximo ponto como próximo ponto da parte, célula inicial como célula atual e ponto de entrada como inválido;
- 5) Enquanto próximo ponto for diferente do último ponto da parte faça:
  - A. Determinar a célula onde se localiza o próximo ponto e marcá-la como célula destino.
  - B. Enquanto a célula atual for diferente da célula destino faça:
    - i. Calcular interseções do segmento, formado pelo ponto atual e pelo próximo ponto da parte, com as bordas da célula atual;
    - ii. Selecionar o ponto que não é ponto de entrada;
    - iii. Calcular vetor de deslocamento (cima, baixo, esquerda e direita) de acordo com a célula interceptada pelo ponto selecionado;
    - iv. Ajustar célula atual para nova célula de acordo com o vetor calculado;
    - v. Ajustar ponto de entrada para o ponto selecionado;
  - C. Ajustar ponto atual para próximo ponto
  - D. Ajustar próximo ponto para próximo ponto da parte

Uma série de detalhes relacionados ao cálculo dos vetores foi omitida na descrição do algoritmo, estes serão destacados a seguir.

Um detalhe que deve ser considerado a parte ocorre quando o ponto de interseção com a célula localiza-se sobre uma de suas bordas. Pertence a célula os pontos localizados sobre as arestas inferior e esquerda. O canto superior esquerdo, o canto inferior direito e o canto superior direito não pertencem à célula, como pode ser evidenciado na Figura 5.4.

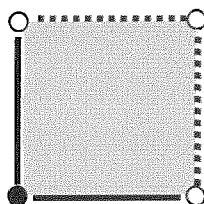


Figura 5.4 – Convenção para ponto sobre as bordas

De acordo com a aresta interceptada pelo ponto de interseção selecionado definem-se os vetores de deslocamento como é apresentado na Tabela 5.5.

Tabela 5.5 – Deslocamento obtido de acordo com a aresta de interseção.

Localização do ponto de interseção	Deslocamento
Aresta direita	Direita
Aresta esquerda	Esquerda
Aresta superior	Cima
Aresta inferior	Baixo

Em particular, deve-se considerar o caso em que ocorre interseção com os cantos da célula. O tratamento para cada um segue a convenção apresentada na Tabela 5.6 e ilustrada na Figura 5.4. Como o canto superior esquerdo pertence à célula de cima o deslocamento computado é para cima (Figura 5.5.a), e esta passa a ser a nova célula. De forma semelhante, o canto inferior direito, segundo a convenção, pertence à célula da direita determinando um deslocamento para a direita (Figura 5.5.b). Já o canto superior direito pertence à célula da diagonal, que deve ser considerada como próxima célula. Como os tipos de deslocamentos existentes não permitem alcançar esta célula em apenas um passo, foi adotado como convenção dois deslocamentos: cima e direita (Figura 5.5.c). O último caso a ser tratado é a interseção com o canto inferior esquerdo, o único que faz parte da célula atual, não indicando diretamente qual deve ser a próxima célula. A solução adotada foi determinar a próxima célula de acordo com a relação entre o próximo ponto da parte e o canto em questão (Tabela 5.7), observando que o deslocamento deve ser para uma célula que garantidamente tenha interseção com o segmento. O objetivo da convenção é determinar deslocamentos que não levem o processamento do algoritmo a um ciclo sem fim. Desse modo, se o próximo ponto localiza-se no quadrante superior esquerdo, é computado um vetor de deslocamento para a esquerda (Figura 5.5.d). Caso o próximo

ponto esteja localizado no quadrante inferior esquerdo, a próxima célula a ser considerada é a da diagonal, como não é possível atingi-la através de um único vetor, por convenção, são gerados dois vetores: esquerda e baixo (Figura 5.5.e). Se o próximo ponto se encontra no quadrante inferior direito gera-se um vetor para baixo (Figura 5.5.f). Caso o canto da célula seja um ponto de final de segmento, significa que a célula destino foi alcançada. Neste caso nenhum vetor de deslocamento é gerado e armazena-se o próprio canto como ponto de entrada na célula (Figura 5.5.g).

Tabela 5.6 – Convenção para definição do vetor de deslocamento quando interseção ocorre com cantos da célula.

Localização do ponto de interseção	Deslocamento
Canto superior direito	Cima e direita
Canto superior esquerdo	Cima
Canto inferior direito	Direita
Canto inferior esquerdo	Coordenada da extremidade final do segmento é analisada de acordo com a Tabela 5.7

Tabela 5.7 – Deslocamento obtido quando a interseção ocorre com o canto inferior esquerdo.

Coordenada do próximo ponto ( $X_p, Y_p$ ) X Coordenada do canto inferior esquerdo da célula atual ( $X_m, Y_m$ )	Deslocamento ou ação
$Y_p \geq Y_m$ e $X_p \leq X_m$	Esquerda
$Y_p \leq Y_m$ e $X_p \leq X_m$	Esquerda e Baixo
$Y_p \leq Y_m$ e $X_p \geq X_m$	Baixo
$Y_p \geq Y_m$ e $X_p \geq X_m$	Ponto Entrada = ( $X_m, Y_m$ )

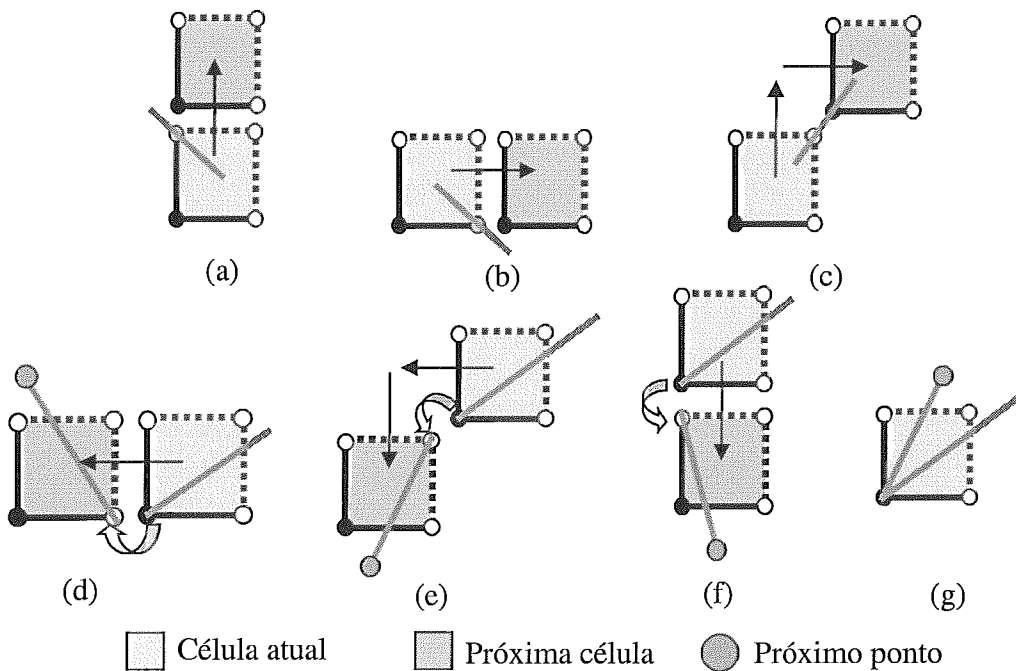
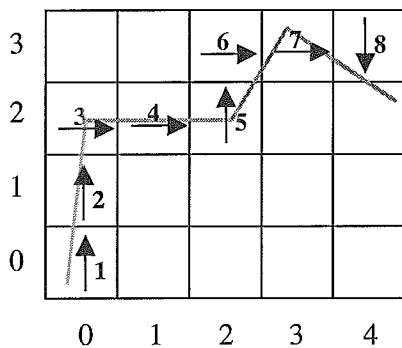


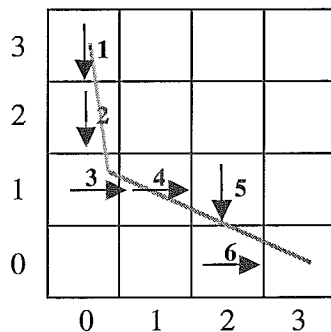
Figura 5.5 – Convenção para interseção com os cantos da célula

A Figura 5.6 exemplifica o cálculo do mapa de *bits* da assinatura da polilinha da Figura 5.1, que é formada por duas partes. No caso da primeira parte, armazena-se que a célula inicial localiza-se na célula (0,0) da grade e os vetores armazenados são: cima, cima, direita, direita, cima, direita, direita e baixo.



Mapa de bits para primeira parte:

- Célula inicial: (0,0)
- Vetores:  $\uparrow \uparrow \rightarrow \rightarrow \uparrow \rightarrow \rightarrow \downarrow$



Mapa de bits para segunda parte:

- Célula inicial: (0,3)
- Vetores:  $\downarrow \downarrow \rightarrow \rightarrow \downarrow \rightarrow$

Figura 5.6 – Cálculo do mapa de bits para a polilinha da figura 5.1

### 5.3. Armazenamento

Armazena-se em disco uma estrutura praticamente idêntica à aproximação vetorial gerada em memória, descrita na seção anterior. Após a geração da aproximação vetorial, primeiramente armazena-se o cabeçalho (Tabela 5.4). Em seguida o mapa de *bits*: célula inicial, número de vetores e vetores (Tabela 5.2). A diferença que aparece com relação à memória é que como cada vetor está associado a um número de dois *bits*, armazenaremos quatro vetores num único *byte*. Os dois primeiros *bits* do *byte* correspondendo ao primeiro vetor, o terceiro e o quarto *bits* ao segundo vetor e assim sucessivamente. Como consequência conseguimos diminuir os espaço ocupado pelos vetores em disco.

### 5.4. Algoritmo de comparação

A geração da aproximação 5CDRS e seu armazenamento em disco são feitos através de vetores de deslocamento. Estes permitem uma elevada compactação da aproximação como foi ressaltado anteriormente. Entretanto, com o objetivo de acelerar o processamento da comparação, esta não é executada diretamente sobre os vetores. Ao invés disso, uma grade de  $m \times n$  células é construída de acordo com os valores armazenados no cabeçalho e os vetores propriamente ditos. Detalhes referentes ao algoritmo que gera esta nova grade são apresentados na próxima seção. Nesta seção estudaremos sua estrutura e o algoritmo para comparação entre as grades geradas.

Cada célula da grade armazena um valor indicando o tipo de interseção que ela possui com relação a polilinha, de acordo com os vetores de deslocamento. Cinco tipos de células podem ser armazenados: horizontal e vertical, horizontal, vertical, inconclusivo e vazia (Tabela 5.8). O tipo horizontal e vertical é armazenado quando a polilinha intercepta a célula nas arestas da direita e da esquerda e também nas arestas inferior e superior aos pares, ou seja, após interceptar a aresta inferior, a próxima aresta interceptada é a superior ou vice e versa e, após interceptar a aresta da esquerda, a aresta seguinte é a da direita ou vice e versa (Figura 5.7.a). O tipo horizontal ocorre quando as arestas atravessadas são apenas a da esquerda e a da direita (Figura 5.7.b) e o vertical quando as arestas interceptadas são a inferior e a superior (Figura 5.7.c). O tipo de célula inconclusiva é armazenado quando a célula é atravessada pela polilinha, mas nenhum dos tipos anteriores é identificado (Figura 5.7.d). Quando a polilinha não atravessa a célula diz-se que a célula é

vazia (Figura 5.7.e). Observe que quando ocorre na mesma célula um tipo de interseção conclusiva (horizontal, vertical ou horizontal e vertical), mas também existe o tipo inconclusivo, o tipo armazenado é o que permite conclusão. Tal fato pode ser evidenciado na terceira célula das Figuras 5.7.a, 5.7.b e 5.7.c.

Tabela 5.8 - Tipos de célula para a 5CDRS

Tipo de Célula	Descrição
Horizontal e Vertical	Interseção da polilinha tanto horizontalmente quanto verticalmente à célula
Horizontal	Interseção da polilinha horizontalmente à célula
Vertical	Interseção da polilinha verticalmente à célula
Inconclusiva	Outros tipos de interseções que não permitem conclusão
Vazia	Célula sem interseção com a polilinha

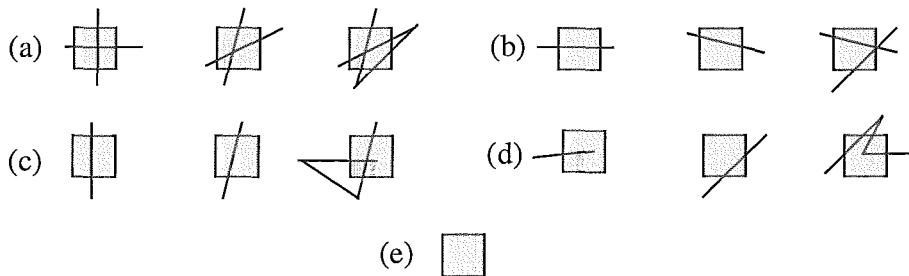


Figura 5.7 - Tipos de células para 5CDRS: (a) horizontal e vertical; (b) horizontal; (c) vertical; (d) inconclusiva; e (e) vazia

Com o objetivo de ilustrar a grade gerada pela 5CDRS, considere o exemplo apresentado na Figura 5.8. Esta mostra os principais rios da bacia amazônica (Figura 5.8.a), em seguida, as grades geradas para todas as aproximações das partes juntamente com um desenho da aproximação calculada (Figura 5.8.b). Finalmente, a Figura 5.8.c exibe a aproximação sem as células da grade.



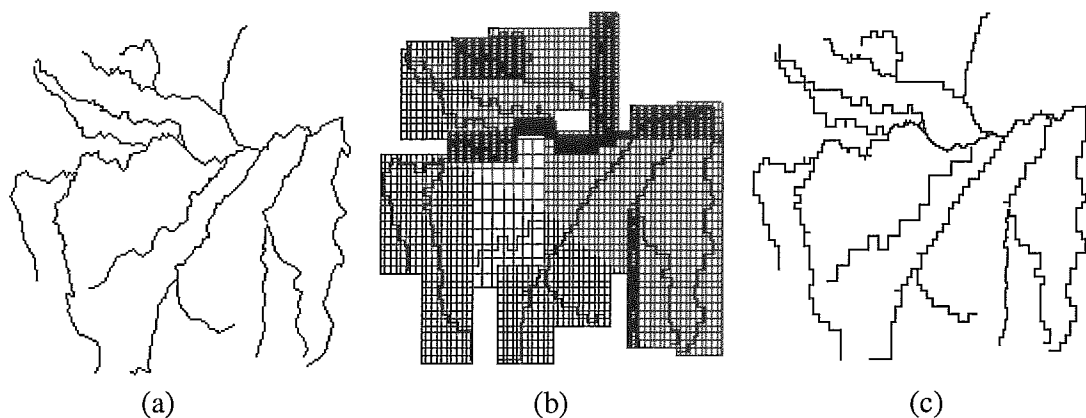


Figura 5.8 - Exemplo da aproximação 5CDRS para comparação.

Ao compararmos duas células, existem três casos possíveis de resposta: rejeitado, quando as células não possuem interseção; aceitos, quando existe interseção da polilinha dentro da célula; e inconclusivo, quando não podemos definir se as polilinhas se interceptam. Observando a Tabela 5.9 temos ao todo que 36% dos casos são rejeitados, 28% aceitos e 36% inconclusivos, ou seja, 64% dos casos de comparação são conclusivos e 36% dos casos inconclusivos. A Figura 5.9 ilustra alguns exemplos de comparações entre células de tipos diferentes.

Tabela 5.9 – Casos possíveis ao compararmos pares de células

Tipo de célula X Interseção	Vazia	Inconclusiva	Vertical	Horizontal	Horizontal e vertical
Vazia	Rejeitada	Rejeitada	Rejeitada	Rejeitada	Rejeitada
Inconclusiva	Rejeitada	<b>Inconclusiva</b>	<b>Inconclusiva</b>	<b>Inconclusiva</b>	<b>Inconclusiva</b>
Vertical	Rejeitada	<b>Inconclusiva</b>	<b>Inconclusiva</b>	<i>Aceita</i>	<i>Aceita</i>
Horizontal	Rejeitada	<b>Inconclusiva</b>	<i>Aceita</i>	<b>Inconclusiva</b>	<i>Aceita</i>
Horizontal e Vertical	Rejeitada	<b>Inconclusiva</b>	<i>Aceita</i>	<i>Aceita</i>	<i>Aceita</i>

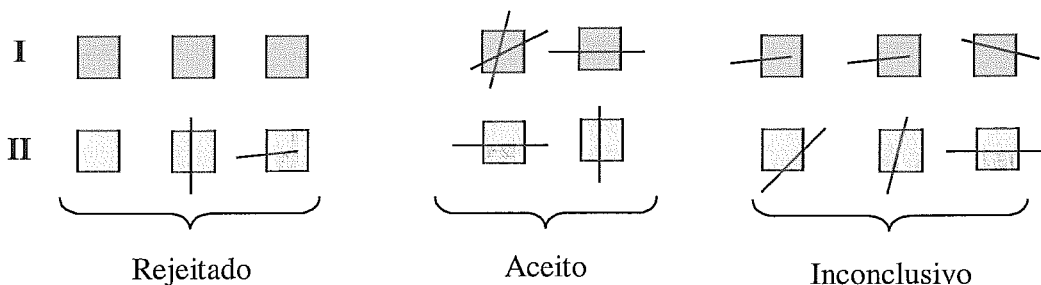


Figura 5.9 – Exemplo de comparação entre células de tipos diferentes.

A verificação se duas polilinhas se interceptam, testando suas aproximações 5CDRS, assim como ocorre com a RIOS, deve ser realizada no MBR de interseção de seus MBRs. Apenas serão testadas as células pertencentes as duas polilinhas ao mesmo tempo. Além disso, as células comparadas devem estar na mesma resolução. Sendo assim pode ser necessário realizar a mudança de resolução na aproximação com menor tamanho de célula antes de executar a comparação, o algoritmo de mudança de resolução será apresentado na próxima seção. Durante a comparação das células, assim que um resultado “*Aceito*” é alcançado podemos parar o teste e afirmar que as polilinhas se interceptam. Abaixo segue o algoritmo de comparação. A Figura 5.10 ilustra características importantes do algoritmo, além de apresentar exemplos de comparações de pares de células.

- 1) Calcular o MBR de interseção dos MBRs das polilinhas (Figura 5.10.b);
- 2) Se os tamanhos das células das aproximações forem diferentes
  - i. Aplicar a mudança de resolução nas células do MBR de interseção da aproximação, com menor tamanho de célula;
- 3) Inicializar resultado como *Rejeitado*;
- 4) Para cada par de células do MBR de interseção faça;
  - i. Se nenhuma das células é vazia;
    - a. Se uma das células é inconclusiva
      - A. Ajustar resultado como *Inconclusivo*;
    - b. Senão Se as duas células são horizontais

A. Ajustar resultado como *Inconclusivo*;

c. Senão Se as duas células são verticais;

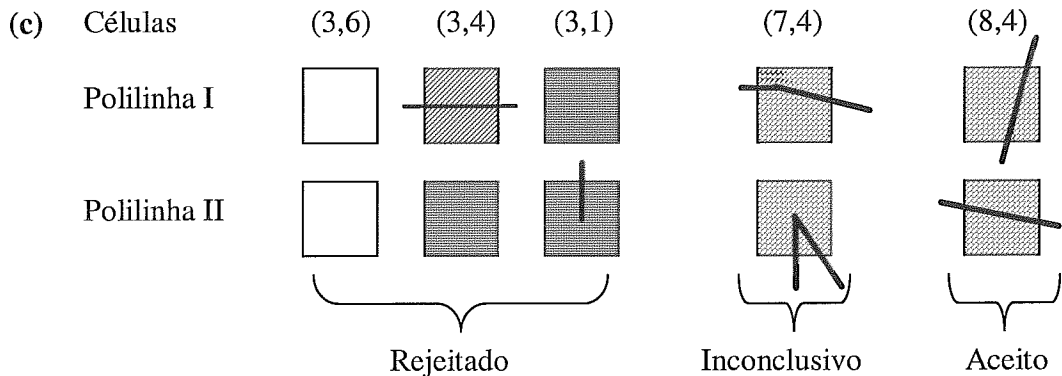
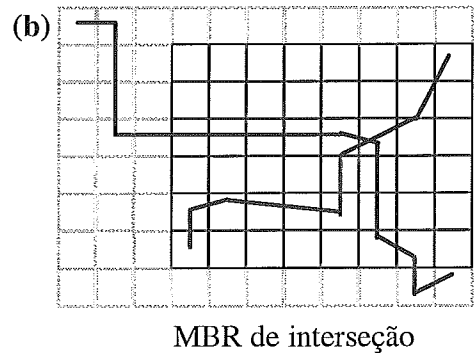
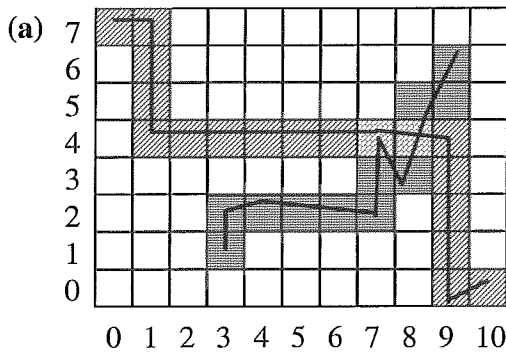
A. Ajustar resultado como *Inconclusivo*;

d. Senão

A. Ajustar resultado como *Aceito*;

B. Sair do laço;

5) Retornar resultado como resultado final da comparação;



5.10 - Exemplo de comparação.

### 5.5. Mudança de resolução

Quando comparamos duas aproximações que possuem células em tamanho diferentes, é necessário compatibilizar através de uma mudança de resolução. A geração da grade de células devido à mudança de resolução é realizada diretamente sobre os vetores direcionais. Da mesma forma que ocorre na mudança de resolução da aproximação RIOS,

não é possível determinar o tipo de célula da grade para células menores, tendo como base apenas os vetores direcionais e não informações reais da polilinha. Logo, a mudança de resolução é aplicada na aproximação com maior resolução (menor tamanho de célula). O algoritmo que será relatado a seguir é o mesmo utilizado para gerar a grade de células quando a mudança de resolução não é necessária. Detalhes omitidos serão apresentados em seguida.

- 1) Geração de uma grade de células a partir do MBR-2<sup>n</sup> armazenado na aproximação. O tamanho das células é definido como sendo o tamanho original, armazenado no cabeçalho da aproximação, ou o novo tamanho para mudança de resolução;
- 2) Inicializar todas as células da grade com o tipo vazio;
- 3) Ajustar célula atual como a célula inicial da aproximação;
- 4) Ajustar ponto atual como o canto inferior esquerdo da célula atual;
- 5) Ajustar tamanho como o tamanho da aresta da célula original;
- 6) Marcar aresta de entrada como inválida;
- 7) Para cada um dos vetores da aproximação faça
  - i. Se vetor igual a *cima*
    - a. Ajustar próximo ponto como ponto atual acrescido de tamanho em sua coordenada y;
  - ii. Senão Se vetor igual a *baixo*
    - a. Ajustar próximo ponto como ponto atual subtraído de tamanho em sua coordenada y;
  - iii. Senão Se vetor igual a *direita*
    - a. Ajustar próximo ponto como ponto atual acrescido de tamanho em sua coordenada x;
  - iv. Senão Se vetor igual a *esquerda*

- a. Ajustar próximo ponto como ponto atual subtraído de tamanho em sua coordenada x;
- v. Ajustar nova célula como a célula que o próximo ponto pertence
- vi. Se célula diferente da célula atual
  - a. Ajustar aresta de entrada como sendo a aresta interceptada pelo segmento formato pelo ponto atual e o próximo ponto;
  - b. Reavaliar situação da célula atual (Tabela 5.10);
  - c. Ajustar célula atual como nova célula;
- vii. Ajustar ponto atual como próximo ponto;

A reavaliação de uma célula é feita considerando-se sua situação atual e arestas de entrada e saída. Se a aresta de entrada for direita e a de saída for esquerda ou vice-versa, uma interseção horizontal é assinalada. Se as arestas forem a de cima e a de baixo, uma interseção vertical é assinalada. Em todos os outros casos é marcada uma interseção inconclusiva. A interseção assinalada e a situação atual da célula são avaliadas e a nova situação é armazenada. As possíveis combinações são apresentadas na Tabela 5.10.

Dessa forma, se temos uma célula assinalada com interseção horizontal e encontramos interseção vertical, quando sua situação é reavaliada, marca-se que a célula possui interseção tanto vertical como horizontal.

Tabela 5.10 – Avaliação da interseção assinalada junto com a situação atual.

Situação atual	Interseção assinalada		
	Inconclusivo	Horizontal	Vertical
Vazio	Inconclusivo	Horizontal	Vertical
Inconclusivo	Inconclusivo	Horizontal	Vertical
Horizontal	Horizontal	Horizontal	Horizontal e Vertical
Vertical	Vertical	Horizontal e Vertical	Vertical
Horizontal e Vertical	Horizontal e Vertical	Horizontal e Vertical	Horizontal e Vertical

## 5.6. Sumário

Este capítulo apresentou a Assinatura *Raster* Direcional de Cinco Cores (5CDRS), outra proposta desta tese para representação de polilinhas a ser utilizada em índices geométricos durante o processamento de consultas espaciais. A principal motivação para a pesquisa e desenvolvimento desta aproximação foi reduzir o espaço de armazenamento necessário para a aproximação RIOS, a qual foi estudada no capítulo anterior, sem perda significativa de informação. Este objetivo foi alcançado e, como resultado das pesquisas, um conjunto de novas idéias foi obtido. Entre estas devemos citar a geração da aproximação por parte e não para toda a polilinha, e a determinação de aproximações apenas para as células da grade atravessadas pela polilinha e não para todas as células da grade através dos vetores direcionais encadeados.

O próximo capítulo é dedicado à apresentação das diferenças entre as aproximações RIOS e 5CDRS, além dos resultados experimentais obtidos. Estes resultados evidenciam que a aplicação da 5CDRS no segundo passo da arquitetura MSQP é muito eficiente, sendo superior aos resultados obtidos por outras aproximações da literatura.

## 6. Resultados Experimentais

Neste capítulo serão discutidas comparações teóricas entre as aproximações RIOS e 5CDRS, apresentadas nos capítulos 4 e 5 respectivamente. Serão também abordados resultados experimentais resultantes da junção de polilinhas com polilinhas utilizando a arquitetura MSQP (BRINKHOFF et al., 1994) empregando a 5CDRS como aproximação para o filtro geométrico em seu segundo passo.

A seção 6.1, deste capítulo, apresenta comparações entre as aproximações RIOS e 5CDRS. A seção 6.2 detalha as estruturas utilizadas em cada passo da arquitetura MSQP. A seção 6.3 apresenta os dados utilizados como fonte para os testes além dos arquivos de aproximações e árvores geradas. Na seção 6.4, são realizadas análises sobre a execução das junções. Finalmente, na seção 6.5 é apresentada uma nova abordagem para a 5CDRS, como uma proposta para trabalho futuro.

### 6.1. Comparação entre as aproximações RIOS e 5CDRS

Esta seção visa apresentar as principais características que diferenciam a aproximação RIOS da aproximação 5CDRS, as quais são destacadas na Tabela 6.1.

Tabela 6.1 – Comparações entre RIOS x 5CDRS

RIOS	5CDRS
Aproximação armazenada para cada célula da grade	Aproximação armazenada apenas para as células interceptadas pela polilinha
Guarda informações das posições de entrada e saída da célula	Guarda informações relativas aos sentidos das direções horizontal e vertical do deslocamento
Tamanho relativamente grande	Alto poder de compactação
Maior identificação de interseção	Menor identificação de interseção
Aproximação de caráter global	Aproximação de caráter local
MBR da aproximação é o da polilinha	MBR da aproximação é o da parte
Conjunto resposta composto de polilinhas	Conjunto resposta composto de partes de polilinhas

A aproximação RIOS armazena uma assinatura de *bits* para cada célula da grade enquanto que a 5CDRS gera uma assinatura apenas para as células atravessadas pela polilinha. Na primeira, uma aproximação é gerada para cada célula da grade. Já na 5CDRS,

segue-se o caminho descrito pela polilinha armazenando as informações de interseção de acordo com as direções do seu deslocamento. Dessa forma, células que não são atravessadas não têm aproximação calculada. Além disso, ao invés de serem armazenadas as posições de entrada e saída, a 5CDRS armazena apenas o deslocamento da célula atual para a seguinte, o qual pode ocorrer nas direções horizontal, vertical ou horizontal e vertical, no caso em que a polilinha atravessa a célula mais de uma vez nestes sentidos. A redução na quantidade de informação armazenada permite diminuir consideravelmente o tamanho da aproximação 5CDRS em disco. Entretanto, como consequência, a capacidade de identificação de interseções pela 5CDRS é reduzida. Ela pode ser vista como sendo uma aproximação RIOS cujas posições de entrada e saída se resumiriam apenas às arestas da célula.

A aproximação RIOS é computada para toda a polilinha, podendo ser caracterizada como sendo uma aproximação de caráter global. Já a 5CDRS possui características mais localizadas, pois a aproximação é construída para cada parte da polilinha. A 5CDRS permite uma maior precisão com relação à posição e extensão do objeto, pois o MBR armazenado é o da parte e não o da polilinha. Isto determina um maior poder de decisão no primeiro passo. Além disso, a comparação de aproximações definidas para partes permite obter como respostas as partes que estão se interceptando, aumentando a precisão do conjunto solução.

## ***6.2. Estruturas utilizadas na arquitetura MSQP***

A arquitetura MSQP foi implementada em um módulo em C++ para testes e geração dos resultados experimentais.

No primeiro passo da arquitetura foi utilizada uma adaptação da Árvore R\* que possibilitasse armazenar nos nós folha aproximações com tamanhos variáveis. Os nós internos mantiveram a mesma concepção que a Árvore R\* original. Nas folhas, o número máximo de entradas foi substituído pela capacidade máxima de armazenamento. Assim, o *split* passou a considerar a soma dos tamanhos das aproximações armazenadas e não mais a quantidade das mesmas. Na criação da árvore, informa-se tanto o número de entradas máximo para nós internos quanto os tamanhos mínimo e máximo que devem ser considerados para os nós folha. Os números máximo e mínimo de entradas, usados pelos



algoritmos da Árvore R\*, são calculados em função desses tamanhos. Basta determinar o tamanho mínimo igual ao tamanho máximo para manter as características da árvore R\* original, armazenando aproximações com tamanhos fixos.

Como aproximação para o filtro geométrico, a ser aplicado no segundo passo, foi utilizada a 5CDRS para os dois conjuntos participantes da junção. Enquanto que o terceiro passo foi implementado utilizando o algoritmo *scan line*, descrito na seção 2.6.

O módulo implementado permite indicar se os terceiro e segundo passos devem ser executados. Este fato possibilita executar a junção em diversas maneiras, possibilitando atingir as conclusões que serão apresentadas a seguir.

Também foi implementado um módulo para visualização dos dados e suas aproximações. Este módulo permitiu validar os resultados de uma melhor maneira, podendo compará-los de forma visual. Nos anexos apresentamos telas exibindo os mapas dos dados utilizados nos experimentos, a visualização das aproximações 5CDRS destes dados, além da grade de células utilizada para construção das aproximações e uma ampliação da mesma.

## **6.2. Descrição dos dados**

Esta seção destina-se à descrição dos dados utilizados como fonte de testes além das aproximações e árvores geradas.

### **6.3.1. Dados dos objetos reais**

Os dados utilizados para testes consistem em polilinhas referentes a fluxos de rios, estradas, rodovias e ferrovias localizadas nos Estados Unidos e Amazônia brasileira. Todos os dados são reais e apresentam complexidades e volumes variados, descritos na Tabela 6.2. Nos anexos são apresentadas telas do sistema de visualização que implementamos exibindo os dados utilizados. As estradas e os rios dos Estados Unidos são apresentados nos anexos A e B (figuras A.1 e A.2). Os rios, estradas e ferrovias de Nova Jersey são ilustrados nos anexos C, D e E (figuras C.1, D.1 e E.1). As estradas e rios do Canadá nos anexos F e G (figuras F.1 e G.1). Finalmente, nos anexos H e I são ilustradas as estradas e os rios da Amazônia brasileira (H.1 e I.1).

Tabela 6.2 - Características dos conjuntos de dados de polilinhas.

Conjunto de dados		Nº objetos	Nº segmentos	Média de segmentos
EUA	Rios	123	6373	52
	Estradas	634	11334	18
	Nova Jersey – Rios	12386	336265	27
	Nova Jersey – Ferrovias	1243	18461	15
	Nova Jersey – Ruas	488670	799265	2
Canadá	Rios	146	6500	45
	Estradas	119	493	4
Amazônia	Rios	12827	312858	24
	Rodovia	1601	12481	8

Os dados obtidos estavam no formato *shape*, proprietário do ArcView (software desenvolvido e comercializado pela Esri). Eles foram extraídos e convertidos para um arquivo contendo os dados dos objetos em formato texto. O arquivo resultante da extração foi utilizado tanto para geração da aproximação como também para acessar os dados do objeto real durante a execução do terceiro passo da arquitetura MSQP.

É importante aqui ressaltar que existe uma grande correlação entre os tipos de dados utilizados como base. Tanto rios e estradas como rios e ferrovias estão intrinsecamente relacionados de forma que, uma estrada ou uma ferrovia é construída acompanhando os leitos dos rios determinando baixa quantidade de interseções entre os dados. Além disso, existe uma certa proximidade entre os objetos, fazendo com que grande parte de seus MBRs tenha interseção sem que haja interseção entre os objetos reais. Isto é importante para testar a capacidade da aproximação e identificação de não interseção. Os resultados obtidos foram bastante promissores como pode ser observado na seção 6.4. Observe também que o arquivo de ruas de Nova Jersey tem complexidade extremamente baixa contendo em média dois segmentos por objeto. Isto ocorre porque a maioria das cidades dos Estados Unidos possui suas edificações dispostas em blocos o que determina ruas formadas por muito poucos segmentos.

### 6.3.2. Aproximações e Árvores

Arquivos de aproximações 5CDRS foram gerados para todos os arquivos base, apresentados na seção anterior. A Tabela 6.3 ilustra a capacidade de compactação da aproximação de acordo com a complexidade dos objetos (número médio de segmentos por objeto). Quanto mais complexo é o objeto, maior é o poder de compactação da aproximação. Tal fato pode ser evidenciado observando que o objeto menos complexo teve uma aproximação que corresponde à cerca de 55% do objeto real, enquanto que para os objetos mais complexos, rios dos Estados Unidos e do Canadá, a proporção da aproximação com relação ao objeto é cerca de 5%.

As aproximações foram geradas com um limite para a grade de no máximo 350 células. Além deste, o tamanho da aproximação foi limitado em 95 *bytes* o que limita o número de vetores em no máximo 256. Limites maiores tanto para número de vetores como para o tamanho da aproximação serão pesquisados como trabalho futuro desta tese permitindo definir os melhores limites para a aproximação.

Tabela 6.3 - Tamanhos dos arquivos base, aproximações e tempo de geração das aproximações

Arquivos de dados		Base (Kb)	5CDRS		5CDRS / Base(%)
			(Kb)	(seg)	
EUA	Rios	115	6	0,15	5,22
	Estradas	213	26	0,37	12,21
	Nova Jersey – Rios	6279	497	9,724	7,92
	Nova Jersey – Ferrovias	365	49	0,702	13,42
	Nova Jersey – Estradas	30959	17132	127,463	55,34
Canadá	Rios	117	7	0,17	5,98
	Estradas	13	5	0,06	38,46
Amazônia	Rios	6202	501	9,3	8,08
	Rodovia	287	64	0,6	22,30

Para ilustrar a representação de polilinhas utilizando a aproximação 5CDRS adicionamos aos anexos telas do módulo de visualização para os dados utilizados. Para cada conjunto de dados temos quatro telas distintas: tela com os dados reais; tela com as aproximações; tela com os dados reais, aproximações e grade *raster* utilizada para construção das aproximações e tela com uma ampliação de parte da tela anterior. As figuras

dos anexos A e B ilustram as estradas e os rios dos Estados Unidos. As figuras dos anexos C, D e E apresentam as estradas, ferrovias e rios de Nova Jersey. Os anexos F e G apresentam figuras contendo as estradas e os rios do Canadá. Finalmente, as figuras dos anexos H e I ilustram as estradas e os rios da Amazônia.

Após a geração das aproximações, duas árvores R\* foram geradas. Uma delas armazena as aproximações nas folhas, a qual é utilizada quando o segundo passo da arquitetura MSQP é executado. Outra é gerada sem as aproximações, armazenando o identificador do objeto e seu MBR. Esta é utilizada quando são processados os passos 1 e 3, sem a execução do passo 2. Assim mantemos os critérios de tamanho mínimo e máximo para o número de entradas dos nós e, neste caso, estes são iguais tanto para nós internos como para nós folha. Isto determina árvores com alturas e espaços de armazenamento diferentes como pode ser evidenciado na Tabela 6.4.

Tabela 6.4 – Tamanho das árvores com e sem aproximação.

Conjuntos de dados		Rtree com aproximações (kb)	Rtree sem aproximações (kb)
EUA	Rios	12	10
	Estradas	44	32
	Nova Jersey - Rios	808	568
	Nova Jersey - Ferrovias	76	58
	Nova Jersey - Ruas	28186	22502
Canadá	Rios	16	12
	Estradas	12	10
Amazônia	Rios	790	568
	Rodovia	102	76

#### 6.4. Análise sobre a execução de junções

Nesta seção serão apresentados detalhes dos resultados experimentais realizados para expor a eficiência obtida com a utilização da aproximação 5CDRS na junção de polilinhas. O ambiente utilizado como teste foi um PC Pentium II 300 MHz com 128Mb de

memória RAM. As junções foram realizadas entre dados pertencentes à mesma região e são apresentadas na Tabela 6.5.

Tabela 6.5 - Junções executadas

Conjuntos		Junção
EUA	Rios x Estradas	A
EUA - Nova Jersey	Ruas x Ferrovias	B
	Rios x Ferrovias	C
	Rios x Ruas	D
Canadá	Rios x Estradas	E
Amazônia	Rios x Rodovias	F

Na execução do primeiro passo da arquitetura MSQP um conjunto de pares candidatos a fazer parte da solução é transmitido para o passo seguinte. Este conjunto é composto por pares que têm interseção de MBR, o que não é suficiente para garantir que existe interseção entre os objetos reais. Assim, seguem, como candidatos para o segundo passo, os pares que têm interseção (*hits*) e aqueles que não se interceptam (*false hits*). A Tabela 6.6 apresenta o número de pares candidatos resultantes do primeiro passo, o número total de pares cujos objetos se interceptam e o percentual deste com relação ao primeiro. A Figura 6.1 ilustra os resultados obtidos na Tabela 6.6, exibindo o percentual dos pares candidatos, que seguem do primeiro para o segundo passo e que não têm interseção, e o percentual dos pares que realmente se interceptam. Apenas os pares que têm interseção fazem parte do conjunto resposta e, como pode ser observado no gráfico, seu percentual é muito menor do que o dos pares que não se encontram na solução. O objetivo do segundo passo é refinar o resultado do primeiro passo, identificando pares que fazem parte do conjunto solução além de pares que não se encontram neste conjunto. Desse modo, o número de pares que seguem para o terceiro passo é reduzido. Como este é o passo mais custoso, conseqüentemente o tempo total de processamento da consulta é reduzido.

Tabela 6.6 - Análise para o primeiro passo

Conjuntos	Nº pares candidatos provenientes do 1º passo	Interseções	
		Total	(%)
A	823	167	20,29
B	322624	10776	3,34
C	7380	1376	18,64
D	507526	31412	6,19
E	145	33	22,76
F	4804	145	3,02

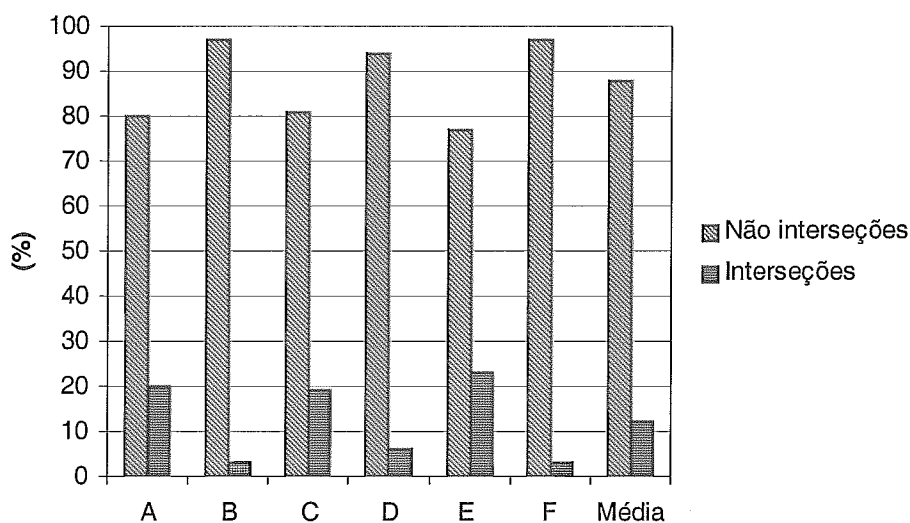


Figura 6.1 - Gráfico ilustrando os percentuais dos pares candidatos, que passam do primeiro para o segundo passo, que não têm interseção e que têm interseção.

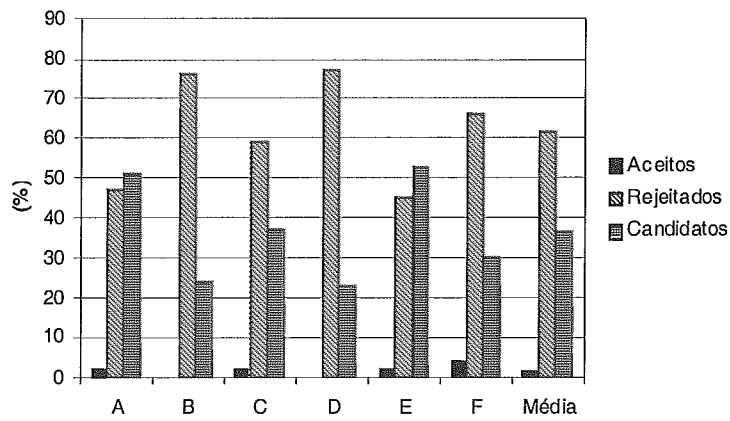
Os pares que seguem para o segundo passo são testados pelo filtro geométrico resultando em pares aceitos, rejeitados ou inconclusivos. São considerados aceitos os pares que têm interseção identificada pela comparação das aproximações enquanto que são rejeitados os pares cujas aproximações não possuem interseção. Os demais são pares inconclusivos e são transmitidos como pares candidatos para o passo seguinte. Na Tabela 6.7 são apresentados os resultados do segundo passo.

Tabela 6.7 – Resultados obtidos com a execução do segundo passo

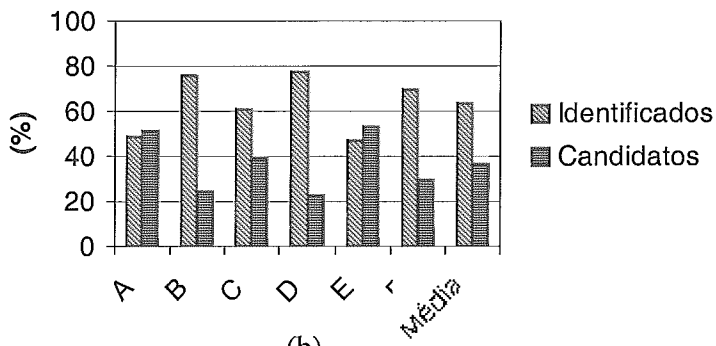
Junção	Aceitos	Rejeitados	Candidatos	Aceitos	Rejeitados	Capacidade
	(%)	(%)	(%)	Identificados	Identificados	total de
				(%)	(%)	decisão (%)
A	1,94	47,14	50,91	9,58	59,15	49,09
B	0,00	75,87	24,13	0,00	78,49	75,87
C	1,53	59,49	38,98	8,21	73,12	61,02
D	0,00	77,17	22,83	0,00	82,26	77,17
E	2,07	44,83	53,10	9,09	58,04	46,90
F	3,64	65,90	30,45	19,13	81,41	69,55
Média	1,53	61,73	36,74	7,67	72,08	63,26

As três primeiras colunas apresentam o percentual de respostas obtidas após a execução do segundo passo. A quarta coluna apresenta o percentual de interseções identificadas em relação ao total existente. A quinta coluna apresenta o percentual de pares que não se interceptam identificados, sobre o total das não interseções existentes no conjunto de pares provenientes do primeiro passo. Na Figura 6.2.a é apresentado um gráfico construído com os dados das três primeiras colunas da Tabela 6.7. Nele podemos observar que a 5CDRS tem maior capacidade de identificação de não interseção do que de interseção. A capacidade de identificar não interseção é bastante elevada tendo média de 72%. A quantidade de informações utilizada para identificar interseções se resume aos vetores de deslocamento, que se dispõem em poucas possibilidades para representações das trajetórias que a polilinha descreve na célula. Além disso, deve-se considerar que os dados utilizados para testes não são muito complexos em sua maioria, possuindo poucos segmentos por objeto, além de terem pouca interseção por natureza, como ressaltado na seção anterior. Apesar destas características não nos favorecerem, obtivemos ótimos resultados como pode ser evidenciado na sexta coluna da Tabela 6.7 e na Figura 6.2.b. O gráfico da Figura 6.2.b demonstra a grande capacidade de redução de pares candidatos que seguem para o terceiro passo. Observe que em apenas dois casos tivemos um número menor de pares identificados. Mesmo assim o percentual de decisão é maior do que o das outras aproximações da literatura, quando utilizadas neste mesmo passo com conjuntos de dados a elas favoráveis, excetuando-se a 4CRS, que foi a base para nossas propostas. A

última tem capacidade de redução em mais de 60% (ZIMBRAO, 1999). A 5CDRS, em média, reduz em 63% dos pares provenientes do primeiro passo que seguem para o terceiro.



(a)



(b)

Figura 6.2 – (a) Gráfico com os percentuais dos que têm interseção, não se interceptam e os que passam como candidatos do segundo para o terceiro passo. (b) Gráfico ilustrando a capacidade de decisão do segundo passo.

O ganho de tempo com a utilização da aproximação 5CDRS no segundo passo da arquitetura MSQP é evidenciado nas tabelas a seguir. Nestas são apresentados os percentuais de tempos obtidos com os experimentos realizados. A Tabela 6.8 exibe o percentual de tempo quando a junção é realizada empregando os três passos da arquitetura. O gráfico da figura 6.3 enfatiza os resultados desta tabela. Nele podemos observar que o segundo passo é o que tem menor impacto de tempo enquanto que o passo com maior percentual de tempo é o terceiro. O primeiro passo apresenta tempo considerável nas junções B, D e F. Isto ocorreu porque o número de pares testados no primeiro passo é muito maior do que os testes realizados nos passos seguintes pois a quantidade de pares inconclusivos do primeiro passo é muito pequena, como pode ser evidenciado na Tabela



6.9. Além disso, como ressaltado anteriormente, os objetos comparados apresentam complexidades muito baixas. As ruas de Nova Jersey têm média de 2 segmentos por objeto. Isto faz com que objetos pouco complexos sejam comparados nos passos seguintes.

Tabela 6.8 - Percentual de tempo de execução nos 3 passos

Junção	Passo 1 (%)	Passo 2 (%)	Passo 3 (%)
A	1,82	3,18	95,00
B	37,50	12,42	50,09
C	19,84	6,95	73,21
D	43,20	5,31	51,49
E	8,00	4,00	88,00
F	32,46	6,12	61,43
Média	23,80	6,33	69,87

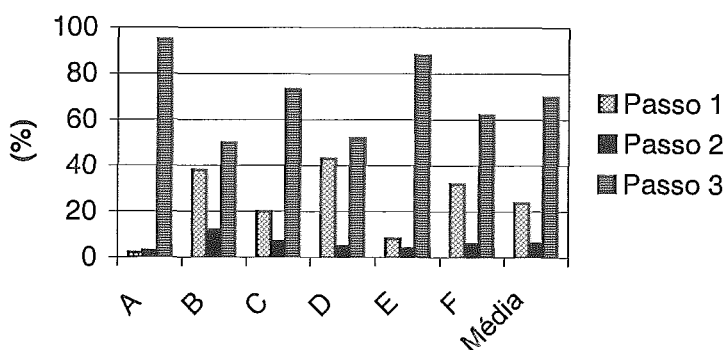


Figura 6.3 – Gráfico com os percentuais de tempo de execução nos 3 passos.

Tabela 6.9 – Resultados das comparações de MBR do primeiro passo executando os passos 1, 2 e 3.

Junção	Comparações de MBR	Pares candidatos	Percentual de candidatos
A	5630	823	14,62
B	5496355	322624	5,87
C	269512	7380	2,74
D	22577247	507526	2,25
E	934	145	15,52
F	441679	4804	1,09

Continuando nossa análise, na Tabela 6.10 temos o percentual de tempo de execução das junções sem o passo 2. Neste caso, o passo três é, sem dúvida, o que consome maior percentual de tempo, como ressaltado no gráfico da Figura 6.4 construído a partir da Tabela 6.10. Apesar das árvores utilizadas no primeiro passo, executando os passos 1 e 3, serem diferentes das árvores utilizadas no mesmo passo quando os passos 1, 2 e 3 são executados, as quantidades de testes de MBRs são muito semelhantes nos dois tipos de execução (tabelas 6.9 e 6.11). Isto salienta ainda mais a relevância que o passo três possui no tempo total de execução e a importância do segundo passo. O último, sendo eficiente e reduzindo a quantidade de pares que seguem para o terceiro passo, reduz consideravelmente o tempo total de processamento. A Tabela 6.12 e a Figura 6.5 demonstram estes fatos, apresentando o tempo total de processamento quando executamos os três passos e quando apenas os passos 1 e 3 são executados. Obtemos um ganho médio de 38% com a execução do segundo passo.

Tabela 6.10 - Percentual de tempo de execução nos passos 1 e 3

Junção	Passo 1 (%)	Passo 3 (%)
A	1,83	98,17
B	17,48	82,52
C	9,53	90,47
D	13,67	86,33
E	4,29	95,71
F	23,75	76,25
Média	11,76	88,24

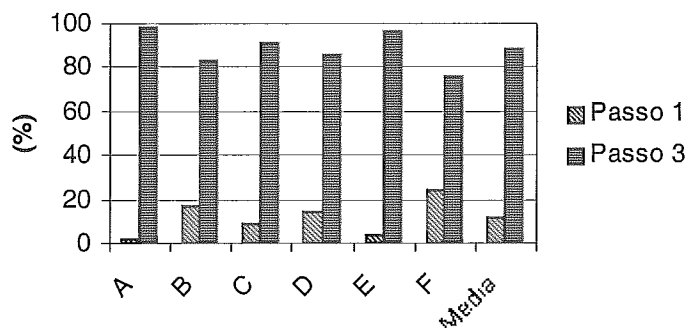


Figura 6.4 – Gráfico com os percentuais de tempo de execução dos passos 1 e 3.

Tabela 6.11 – Resultados das comparações de MBR do primeiro passo executando os passos 1 e 3.

Junção	Comparações de MBR	Pares candidatos	Percentual de candidatos
A	5541	823	14,85
B	5208744	322624	6,19
C	224445	7380	3,29
D	18005965	507526	2,82
E	839	145	17,28
F	389564	4804	1,23

Tabela 6.12 –Tempo total de execução.

Junção	Passo 1, 2, e 3 (seg)	Passo 1 e 3 (seg)	Ganho de tempo (%)
A	4,40	8,20	46,34
B	354,10	642,10	44,85
C	20,04	29,53	32,15
D	940,08	2021,57	53,50
E	0,50	0,70	28,57
F	16,20	20,20	19,80
Média	222,55	453,72	37,54

Com o objetivo de uma melhor explanação dos dados de teste até aqui apresentados, considere os fatos que se seguem. O passo mais custoso da arquitetura MSQP é o terceiro passo, como apontado por BRINKHOFF et al. (1994). A complexidade dos objetos reais, que devem ser lidos do disco para a memória principal, pode desencadear uma grande quantidade de acessos a disco, aumentando consideravelmente o tempo de execução da junção. Além disso, o algoritmo empregado na sua execução pode requerer elevado tempo de CPU, aumentando ainda mais seu tempo de processamento. Deve ser ressaltado que foi utilizada uma variante do algoritmo *plane-sweep* para polilinhas (*scan line*) nos testes. BRINKHOFF et al. (1994) aponta este algoritmo como sendo bastante ineficiente podendo

ser substituído por outros, tais como SID (detecção simbólica de interseções) apresentado em HUANG (1997), com resultados muito mais eficientes.

Desse modo, medir apenas o tempo de execução dos diferentes passos não é suficiente para ilustrar o que realmente está ocorrendo. Como medidas adicionais podem ser considerados o percentual de pares candidatos que são filtrados no segundo passo (Tabela 6.7) e os acessos a disco realizados no passo 3 (Tabela 6.13). Esta tabela evidencia a grande diminuição de acessos a disco quando há a execução do segundo passo, cuja redução chega a atingir 75% de redução quando o segundo passo é executado.

Tabela 6.13 - Número de acessos a disco no 3º passo

Junção	Nº de acessos a disco no passo 3		Redução (%)
	Passos 1, 2 e 3	Passo 1 e 3	
A	2217	4188	47
B	44426	177517	75
C	2724	6835	60
D	138520	557019	75
E	95	171	44
F	1116	3165	65
Média	31516,33	124815,83	61

## 6.5. Sumário

Este capítulo apresentou comparações entre a RIOS e a 5CDRS, enfatizando suas vantagens e desvantagens, além de resultados experimentais empregando a 5CDRS no segundo passo da arquitetura MSQP. Foram realizadas comparações executando os 3 passos da arquitetura, empregando a 5CDRS no passo 2, e executando apenas os passos 1 e 3. A 5CDRS mostrou-se muito eficiente. O segundo passo permitiu em média: identificar 63% dos pares candidatos provenientes do primeiro passo; reduzir 37% de tempo de execução das junções; e reduzir 61% dos acessos a disco. O próximo capítulo é dedicado à apresentação das conclusões desta tese.

## 7. Conclusão

Este capítulo apresenta as conclusões obtidas com os trabalhos realizados. Esta tese teve como principal objetivo propor aproximações *raster* para representação de polilinhas e avaliar sua eficiência quando aplicada em filtros geométricos no segundo passo da arquitetura MSQP (BRINKHOFF et al., 1994).

Duas propostas foram apresentadas: a RIOS e a 5CDRS. A primeira caracterizada pelo armazenamento das posições de entrada e saída da polilinha em cada célula da grade *raster*. A segunda guarda informações dos sucessivos deslocamentos da polilinha na grade (vetores direcionais). Ambas têm a capacidade de identificar tanto interseções como também ausência de interseções entre polilinhas. Existem várias diferenças entre elas; a primeira seção do capítulo 6 foi dedicada a essa caracterização. Pode-se ressaltar a maior quantidade de informações armazenadas pela RIOS em relação à 5CDRS que permite maior identificação de interseções em conjuntos complexos. Entretanto isto acarreta a necessidade de maior espaço para armazenamento da aproximação. A 5CDRS, com o armazenamento de menos informações, tem menor capacidade de identificação de interseções. Todavia, armazenar apenas vetores de deslocamento tem como consequência uma grande compactação. Deve ser enfatizado que, quanto mais complexos são os objetos, mais compacta é a aproximação, fato caracterizado nos resultados experimentais do capítulo anterior.

No capítulo 6 foram apresentadas as principais diferenças existentes entre as aproximações e um conjunto de testes empregando a 5CDRS foi realizado, devido às características da 5CDRS. A arquitetura de três passos de BRINKHOFF et al. (1994) foi utilizada como estrutura para testes empregando a 5CDRS no segundo passo. Os resultados obtidos caracterizaram a eficiência desta proposta, apesar dos dados não terem características que a favoreceriam, como ressaltado no capítulo 6. Apesar de não ter bom desempenho para identificação de interseção, a 5CDRS teve ótimo resultado para concluir não interseção. Em média, 72% dos pares candidatos que não têm interseção, provenientes do primeiro passo, são identificados no segundo passo. A capacidade total média de filtragem foi de 63%. Assim, 63% dos pares candidatos provenientes do primeiro passo

foram filtrados no segundo, não sendo necessário executar o teste de geometria exata para eles. BRINKHOFF et al. (1993) relatam que as aproximações geométricas existentes têm média de 46% de redução quando aplicadas no segundo passo da MSQP, o que enfatiza ainda mais a eficiência da nossa aproximação.

Além de servir como um guia para o estudo da utilização de aproximações para representar polilinhas, a presente tese também pode ser utilizada para o entendimento do processamento de consultas espaciais, já que estruturas e algoritmos relacionados são aqui destacados.

Como trabalhos futuros temos a avaliação dos melhores limites para o número máximo de células da grade e do número máximo de vetores para a 5CDRS. Estes estão relacionados ao custo de armazenamento e processamento, à capacidade de conclusão obtida e às características dos objetos a serem aproximados. Outra proposta interessante é desenvolver uma aproximação que agregue as melhores características da RIOS e da 5CDRS. Uma das abordagens seria armazenar as posições de entrada e saída da polilinha nas células atravessadas, ao invés de guardar apenas os vetores de deslocamento. Isto armazenaria um maior conjunto de informações para a detecção de interseção e manteria a concepção de armazenar informações apenas para as células atravessadas permitindo menor custo de armazenamento.

## 8. Bibliografia

- ABEL, D. J., 1985, “Some elemental operations on linear quadtrees for geographic information systems”, *Computer Journal*, v. 28, n. 1 (Feb), pp. 73-77.
- ABEL, D. J., SMITH, J. L., 1983, “A data structure and algorithm based on a linear key for a rectangle retrieval problem”, *Computer Vision, Graphics, and Image Processing*, v. 24, n. 1 (Oct.), pp. 1 – 13.
- ARGE, L., et al., 1998. “Scalable Sweeping Based Spatial Join”. In: *24th International Conference on Very Large Databases*, New York City, NY, USA, August, 1998.
- BERCHTOLD, S., KEIM, D. A., KRIEGEL, H. P., 1996. “The X-Tree: An Index Structure for High-Dimensional Data”, In: *Proceedings of 22nd International Conference on Very Large Data Bases*, Bombay, India 1996.
- BRINKHOFF, T., KRIEGEL, H. P., SCHNEIDER, R., et al., 1994. “Multi-step Processing of Spatial Joins”. In: *Proceedings of the 1994 ACM-SIGMOD Intl. Conference*, Minneapolis, USA, May 1994.
- BRINKHOFF, T., KRIEGEL, H. P., SCHNEIDER, R., 1993b. “Comparison of Approximations of Complex Objects Used for Approximation-based Query Processing in Spatial Database Systems”. In: *Proceedings of 9th International Conference on Data Engineering*, Vienna, Austria, Mar 1993.

- BRINKHOFF, T., KRIEGEL, H. P., SEEGER, B., 1993a. "Efficient processing of Spatial Joins Using R-Trees". In: *Proceedings of the 1993 ACM-SIGMOD Intl. Conference*, Washington, DC, USA, May 1993.
- BRINKHOFF, T., KRIEGEL, H. P., SEEGER, B., 1996. "Parallel Processing of Spatial Joins Using R-Trees", In: *Proceedings of 12th International Conference on Data Engineering*, New Orleans, LA, USA 1996.
- DEWITT, D., PATEL, J. M., 1996. "Partition Based Spatial-Merge Join". In: *Proceedings of the 1996 ACM-SIGMOD Conference*, Montreal, Canada, June 1996.
- GUTTMAN, A., 1984. "R-Trees: A Dynamic Index Structure for Spatial Searching". In: *Proceedings of the ACM SIGMOD Intl. Conf on Management of Data*, Boston, MA, USA, May 1984.
- HUANG, Y. W., JONES, M. C., RUNDESTEINER, E. A., 1997. "Improving Spatial Intersect Joins Using Symbolic Intersect Detection". In: *Proceedings of the 5th International Symposium on Advances in Spatial Databases, SSD'97*, Berlin, Germany, July 1997.
- KAUSHIK, S., RUNDENSTEINER, E. A. 1998. "SVIQUER: A Spatial Visual Query and Exploration Language". In: *9th International Conference on Database and Expert Systems Applications - DEXA*, pp. 290-299, Vienna, Austria, August 1998.
- KEDEM, G., 1982, "The quad-CIF tree: a data structure for hierarchical on-line algorithms", In: *Proceedings of the Nineteenth Design Automation Conference*, pp. 352-357, Las Vegas, June.



KNUTH, D. E., 1973, *The Art of Computer Programming*. 2 ed. New York, Boston, MA, USA.

KOUDAS, N., SEVCIK, K., 1997. "Size Separation Spatial Join". In: *Proceedings of ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, USA, May 1997.

MAMOULIS, N., PAPADIAS, D., 1999. "Integration of Spatial Join Algorithms for Joining Multiple Inputs". In: *Proceedings of the ACM SIGMOD Conference on the Management of Data*, Philadelphia, PA, USA, May 1999.

MING, L. L., RAVISHANKAR, C. V. "Generating Seeded Trees From Data Sets". In: *Proceedings of the 4th International Symposium on Large Spatial Databases*, Portland, ME, USA, August, 1995.

MING, L. L., RAVISHANKAR, C. V. "Spatial Hash-Joins". In: *Proceedings of the 1996 ACM SIGMOD Conference*, Montreal, Canada, June 1996.

MONTEIRO, R. S., AZEVEDO, L. G., ZIMBRAO, G., SOUZA, J. M., 2001a, *Polyline Join Evaluation Using Raster Approximation*, ES-550/110, Programa de Engenharia e Sistemas e Computação.

MONTEIRO, R. S., 2001b, *Junções de Polilinhas e Polígonos com Uso de Filtros Raster*, Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- MORTON, G. M., “A computer oriented geodetic data base and a new technique in file sequencing”, IBM Ltd., Ottawa, Canada, 1966.
- SAIF, MELP, 1994. *Spatial Archive and Interchange Format (SAIF): Formal Definition Release 3.1*, Reference Series Volume 1 – Surveys and Resource Mapping Branch – Ministry of Environment, Lands and Parks (MELP) – Province of British Columbia – Canada, April 1994.
- SAMET, H., 1990. *The Design and Analysis of Spatial Data Structure*, 2 ed, NYC, Addison-Wesley Publishing Company, 1990.
- SHAFFER, C. A., 1986, *Application of alternative quadtree representations*. Ph. D. dissertation, University of Maryland, College Park, MD, USA.
- SQL, 1995. *X3H2-95-084/DBL:Yow-004, (ISO-ANSI Working Draft) Database Language SQL (SQL3)*, Jim Melton Editor, March 1995.
- SUTHERLAND, I. E., SPROULL, R. F., SCHUMACKER, 1974, “A characterization of ten hidden-surface algorithms”, *ACM Computing Surveys*, v. 6, n. 1 (Mar), pp. 1-55.
- SWARCFITER, J. L., MARKENZON, L., 1994. *Estruturas de Dados e seus Algoritmos*, 2 ed., Livros Técnicos e Científicos Editora, Rio de Janeiro, 1994.
- ZIMBRÃO, G., 1999, *Avaliação de Junções em Bancos de Dados Espaciais*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

ZIMBRÃO, G., SOUZA, J. M., 1998. “A Raster Approximation For Processing of Spatial Joins”. In: *Proceedings of the 24th International Conference on Very Large Databases, VLDB'98*, NYC, USA, August 1998.

ZIMBRÃO, G., SOUZA, J. M., 1997. “Realização Eficiente de Junções Espaciais Utilizando Hash-Join”. In: *Anais do XII Simpósio Brasileiro de Banco de Dados*, Fortaleza, CE, Brasil, Outubro, 1997.

ZIMBRÃO, G., SOUZA, J. M., MONTEIRO, R. S., AZEVEDO, L. G., 2000. “Filtro Raster para Junção de Polilinhas”. In: *Anais do XV Simpósio Brasileiro de Banco de Dados*, João Pessoa, Brasil, Outubro 2000.

ZIMBRAO, G., MONTEIRO, R. S., AZEVEDO, L. G., SOUZA, J. M., 2001, *Polyline Join Evaluation Using Raster Aproximation, ES-544/01*, Programa de Engenharia e Sistemas e Computação.

## ANEXOS

Nos anexos que se seguem são apresentadas telas do módulo de visualização exibindo os mapas dos dados utilizados nos experimentos, as aproximações geradas para estes dados, além das grades de células utilizadas para geração das aproximações juntamente com as aproximações e os objetos reais. Devido à quantidade de informação da figura exibindo as grades, assinaturas e os objetos, apresentamos uma tela contendo uma ampliação de uma parte desta figura.

Nos anexos A e B são apresentadas às telas correspondentes às estradas e aos rios dos Estados Unidos. Os anexos C, D e E contêm as telas para as estradas, ferrovias e rios de Nova Jersey. Os anexos F e G apresentam as telas das estradas e dos rios do Canadá. Finalmente, os anexos H e I exibem as telas das estradas e dos rios da Amazônia brasileira.

## ANEXO A

Neste anexo são apresentadas as telas das estradas dos Estados Unidos (Figura A.1); suas aproximações 5CDRS (Figura A.2); as estradas com suas aproximações e suas grades *raster* (Figura A.3) e uma ampliação de parte desta tela (Figura A.4).

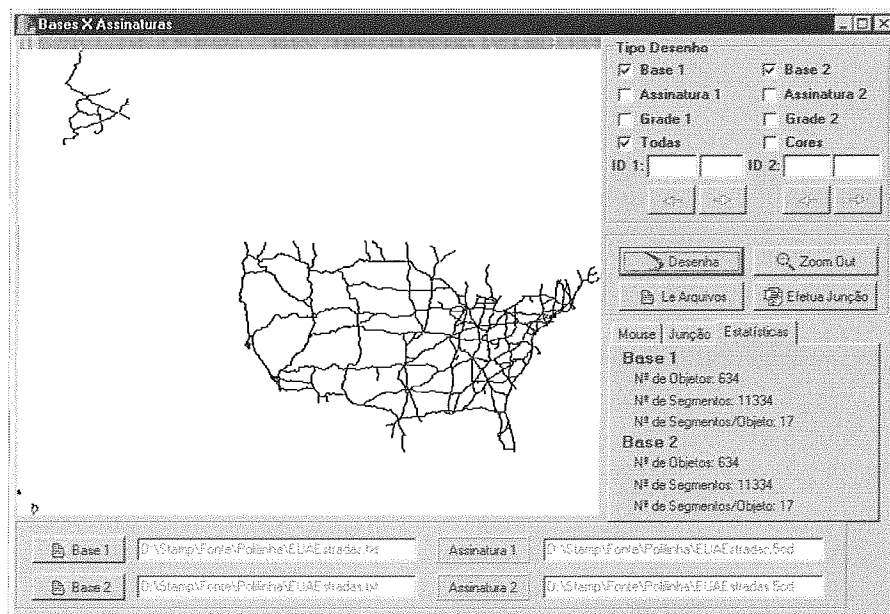


Figura A.1 - Estradas dos Estados Unidos

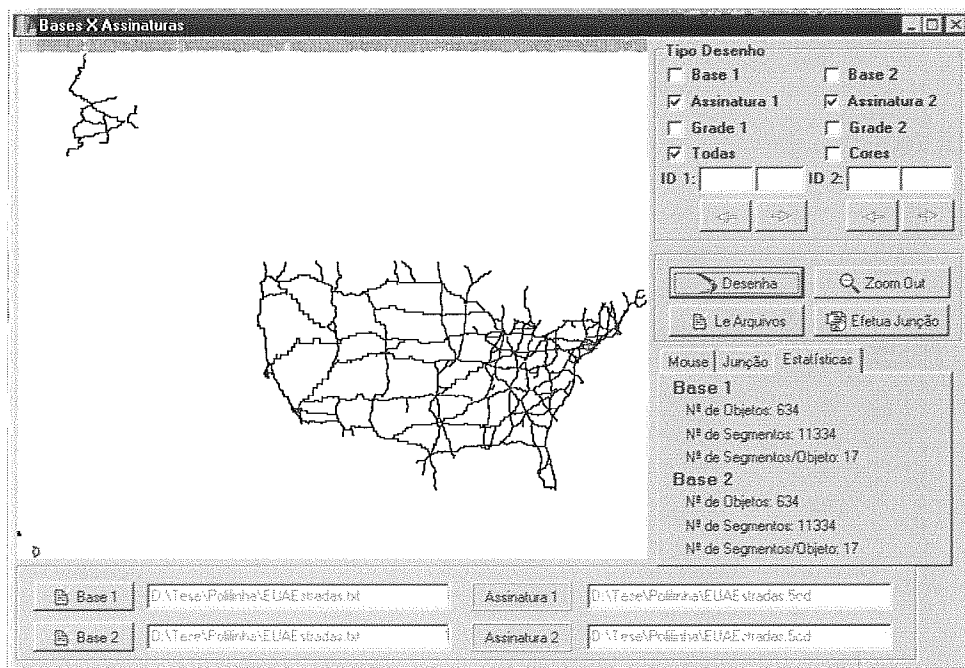


Figura A.2 - Aproximações 5CDRS das estradas dos Estados Unidos

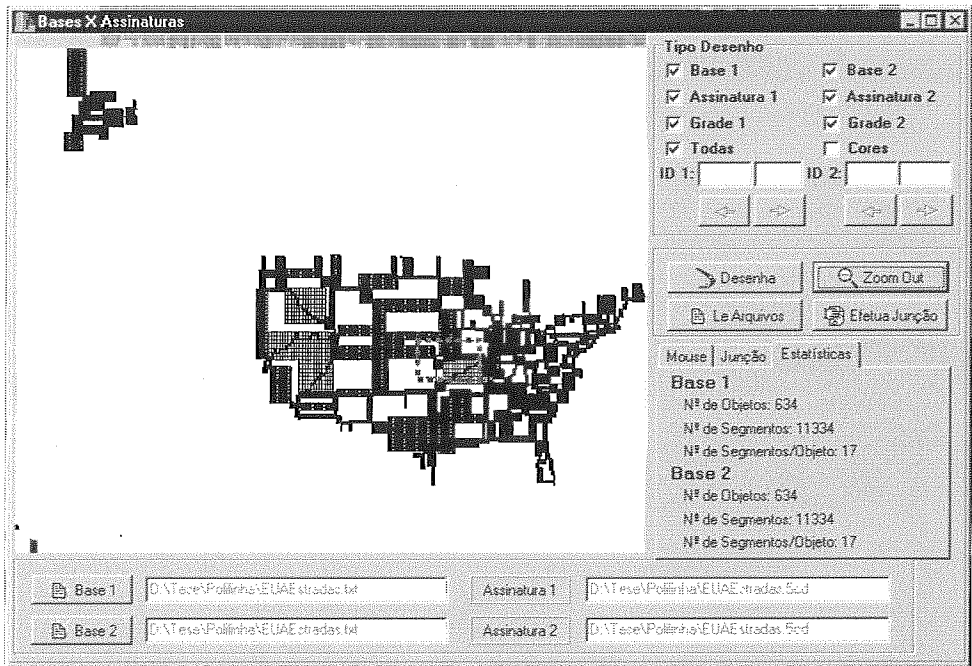


Figura A.3 - Estados dos Estados Unidos com suas aproximações 5CDRS e suas grades Raster

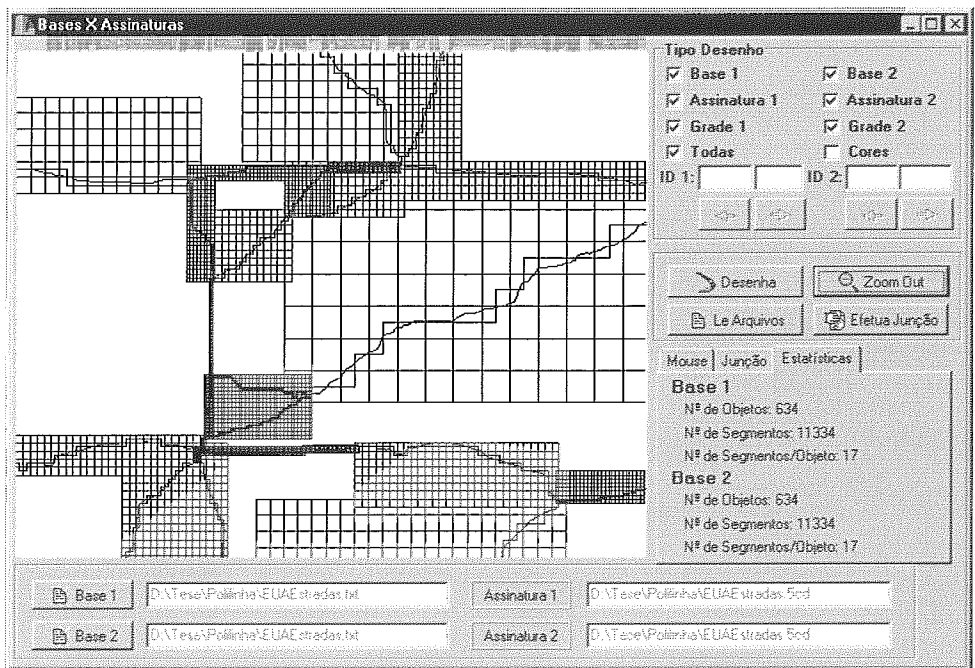


Figura A.4 - Ampliação de parte das estados dos Estados Unidos com suas aproximações 5CDRS e suas grades Raster.

## ANEXO B

Neste anexo são apresentadas as telas dos rios dos Estados Unidos (Figura B.1); suas aproximações 5CDRS (Figura B.2); os rios com suas aproximações e suas grades *raster* (Figura B.3) e uma ampliação de parte desta tela (Figura B.4).

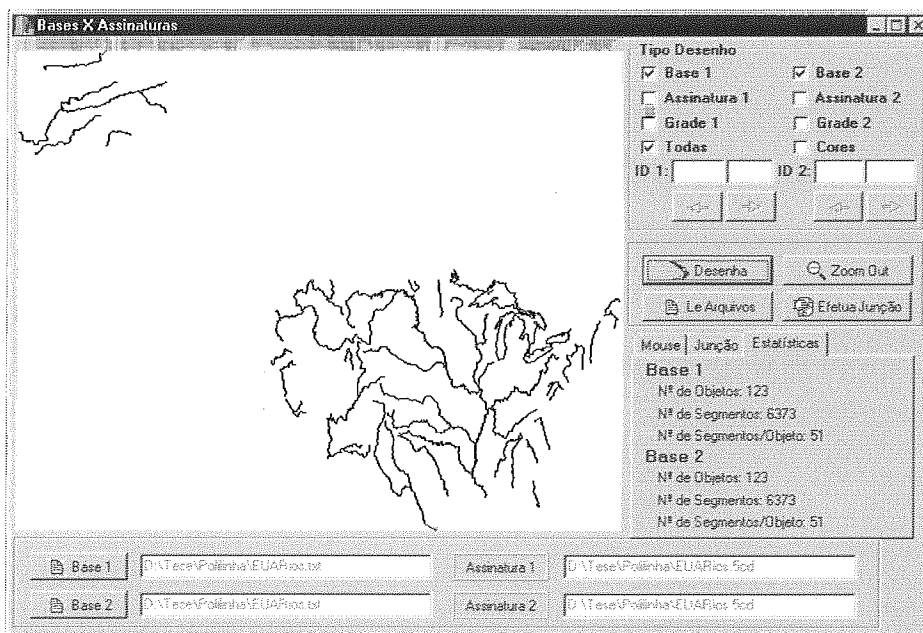


Figura B.1 - Rios dos Estados Unidos.

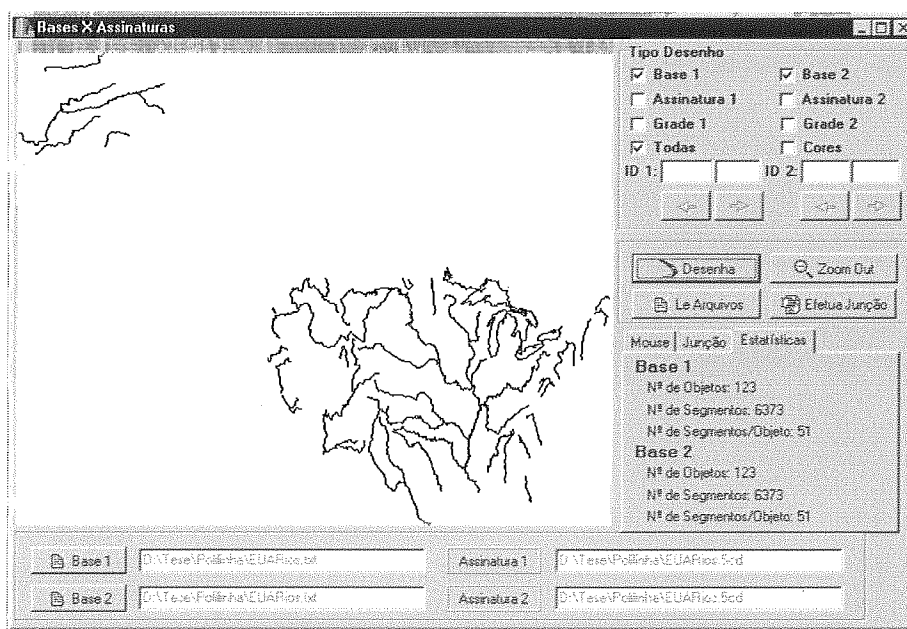


Figura B.2 - Aproximações 5CDRS dos rios dos Estados Unidos.

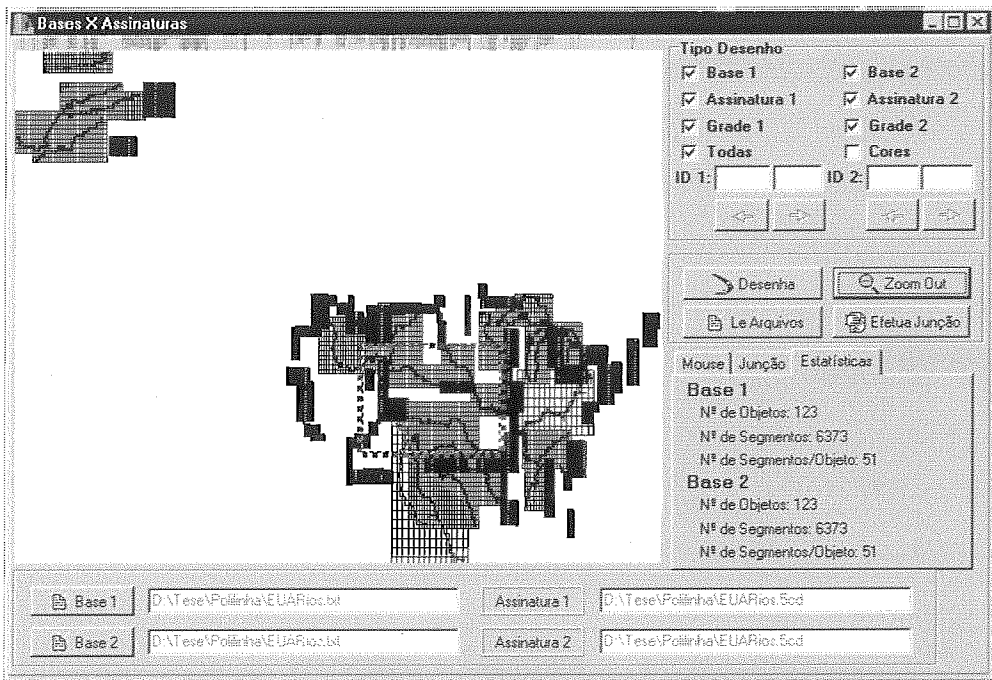


Figura B.3 - Rios dos Estados Unidos com suas aproximações 5CDRS e suas grades *Raster*.

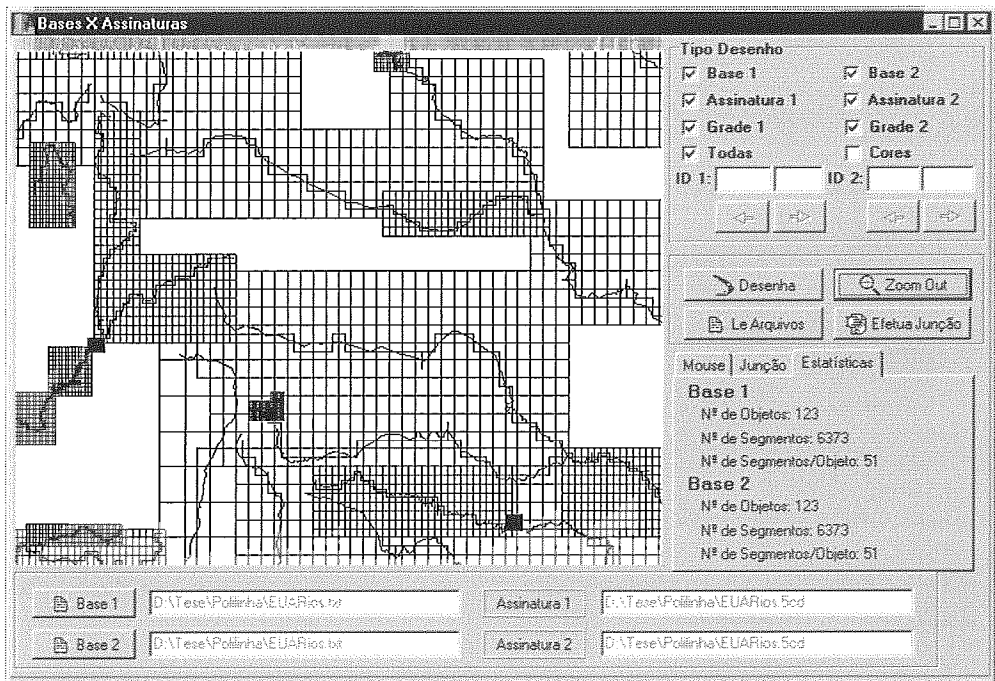


Figura B.4 - Ampliação de parte dos rios dos Estados Unidos com suas aproximações 5CDRS e suas grades *Raster*.



## ANEXO C

Neste anexo a tela das estradas de Nova Jersey e apresentada (Figura C.1). Optamos por não apresentar as telas com as assinaturas e as grades devido a grande densidade dos dados. Isto inviabilizou a confecção de figuras capazes de ilustrar de forma clara estes dados.

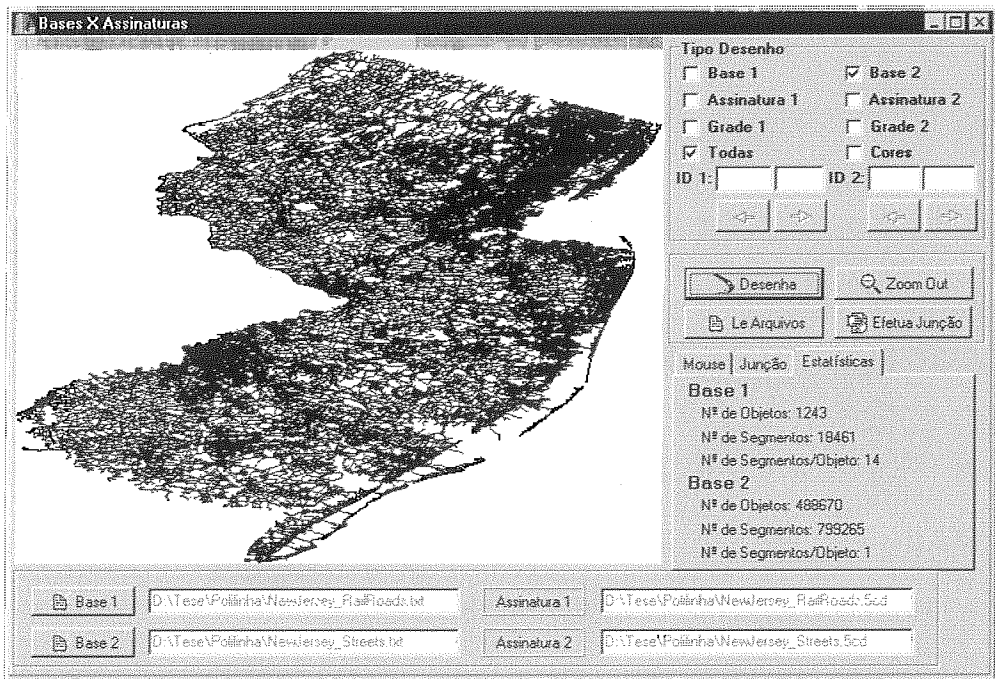


Figura C.1 – Estradas de Nova Jersey.

## ANEXO D

Neste anexo são apresentadas as telas das ferrovias de Nova Jersey (Figura D.1); suas aproximações 5CDRS (Figura D.2); as ferrovias com suas aproximações e suas grades *raster* (Figura D.3) e uma ampliação de parte desta tela (Figura D.4).

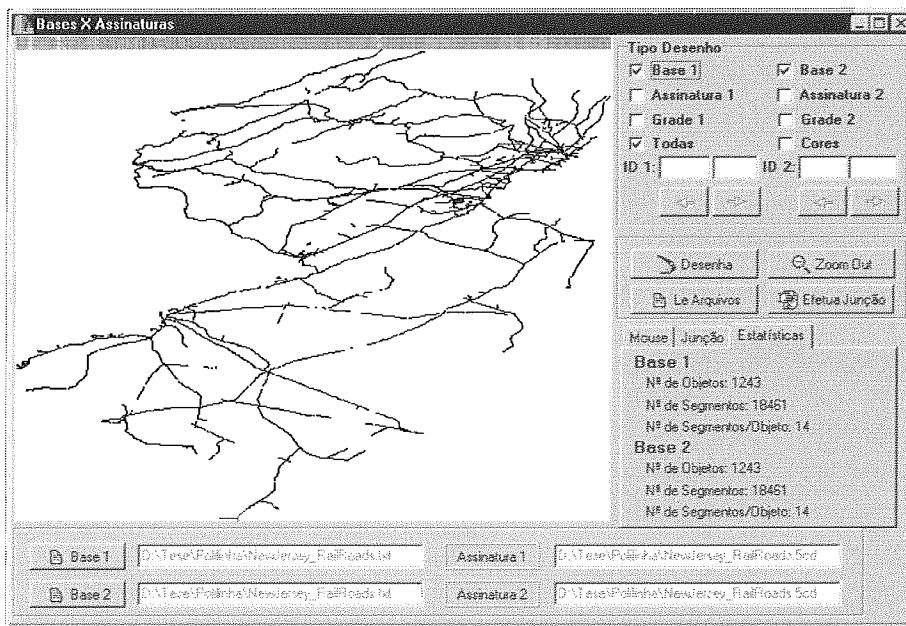


Figura D.1 - Ferrovias de Nova Jersey.

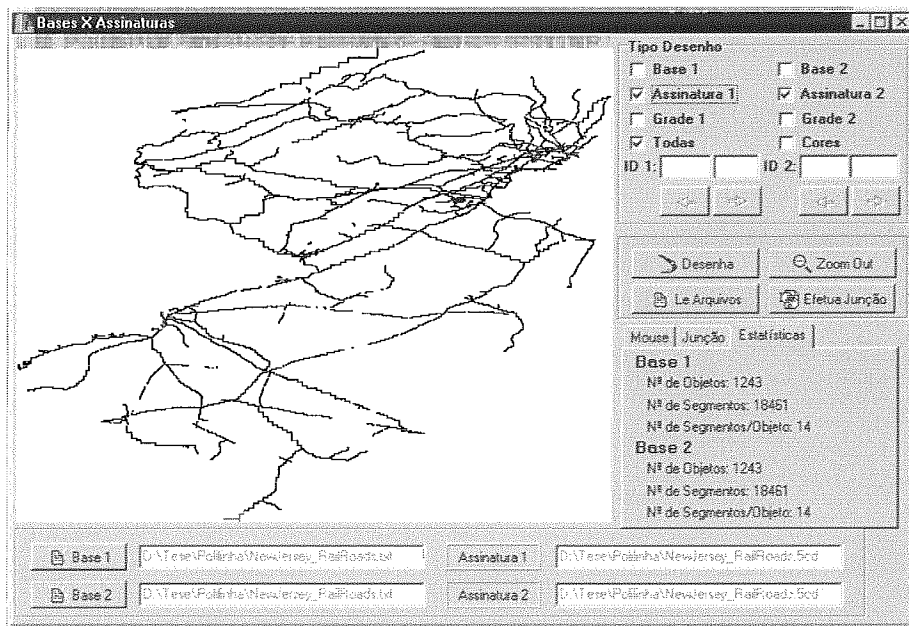


Figura D.2 - Aproximações 5CDRS das ferrovias de Nova Jersey.

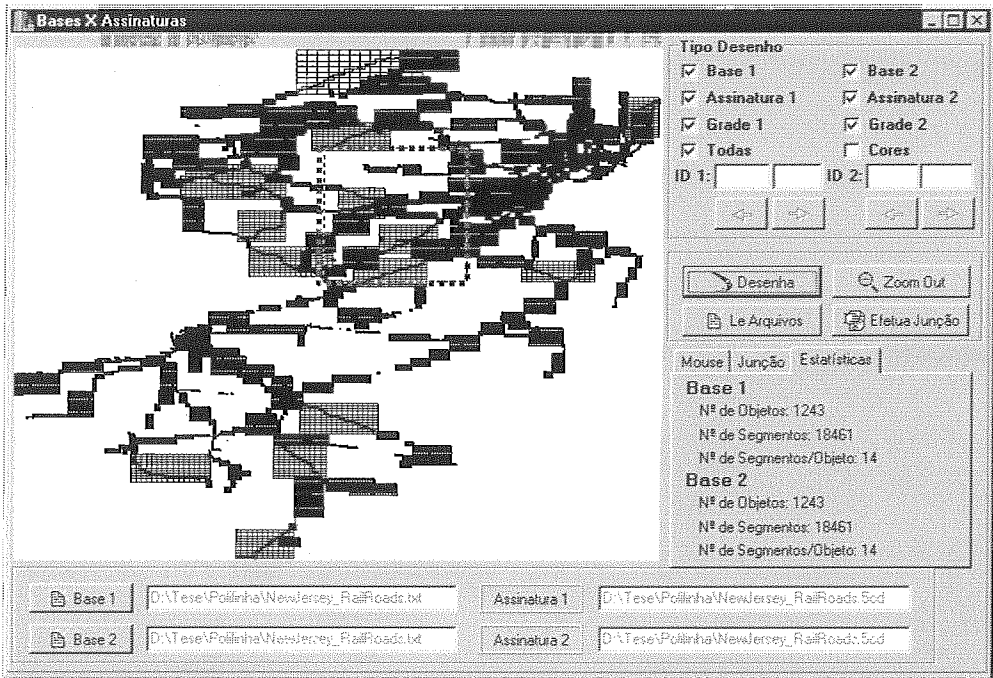


Figura D.3 - Ferrovias de Nova Jersey com suas aproximações 5CDRS e suas grades Raster

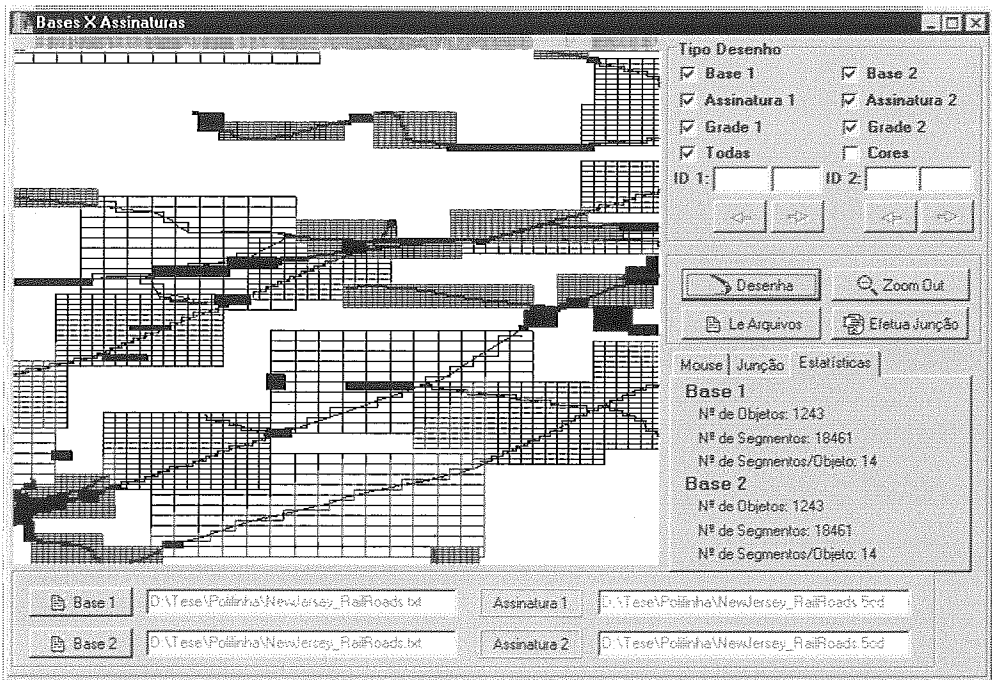


Figura D.4 - Ampliação de parte das ferrovias de Nova Jersey com suas aproximações 5CDRS e suas grades Raster.

## ANEXO E

Neste anexo são apresentadas as telas dos rios de Nova Jersey (Figura E.1); suas aproximações 5CDRS (Figura E.2); os rios com suas aproximações e suas grades *raster* (Figura E.3) e uma ampliação de parte desta tela (Figura E.4).

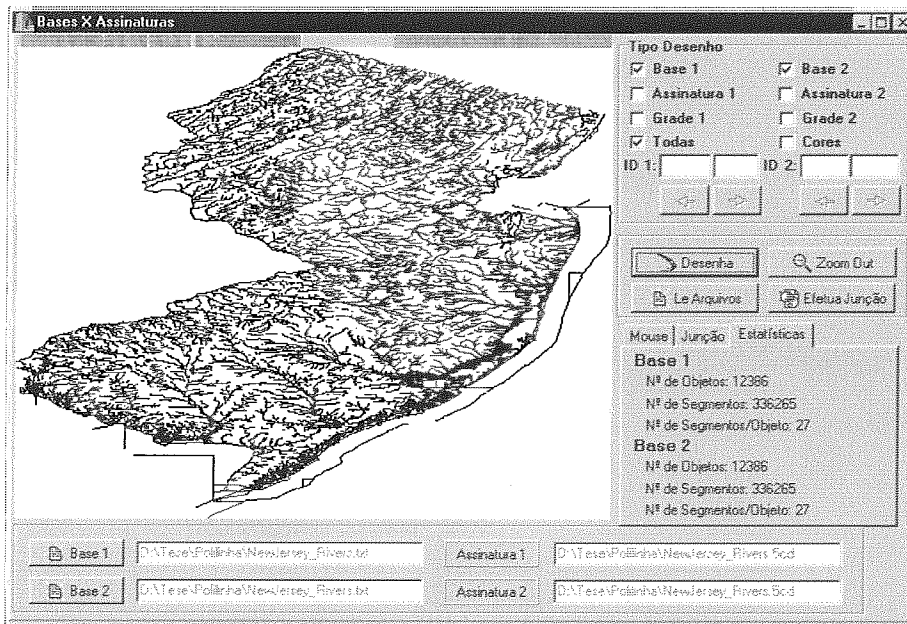


Figura E.1 - Rios de Nova Jersey.

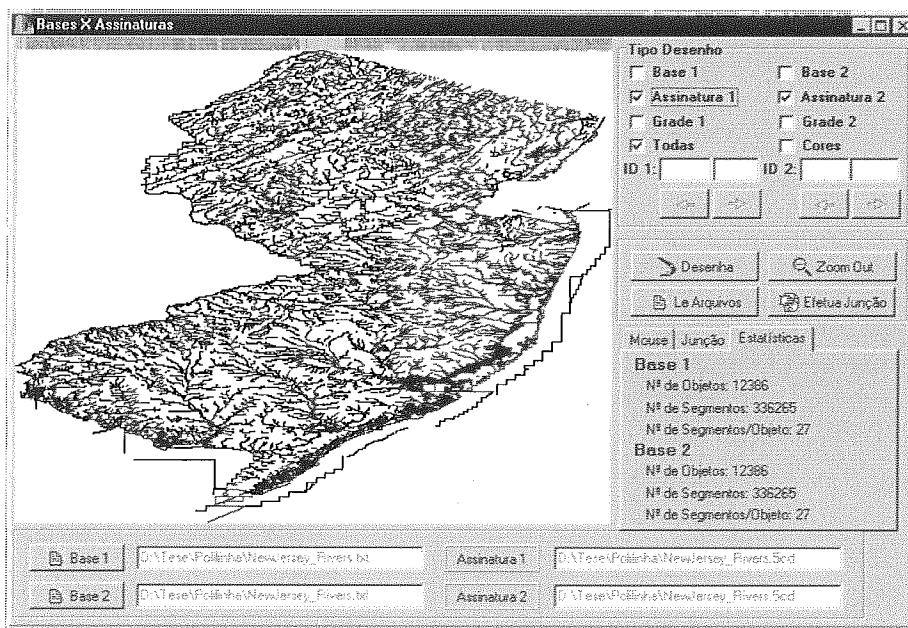


Figura E.2 - Aproximações 5CDRS dos Rios de Nova Jersey.

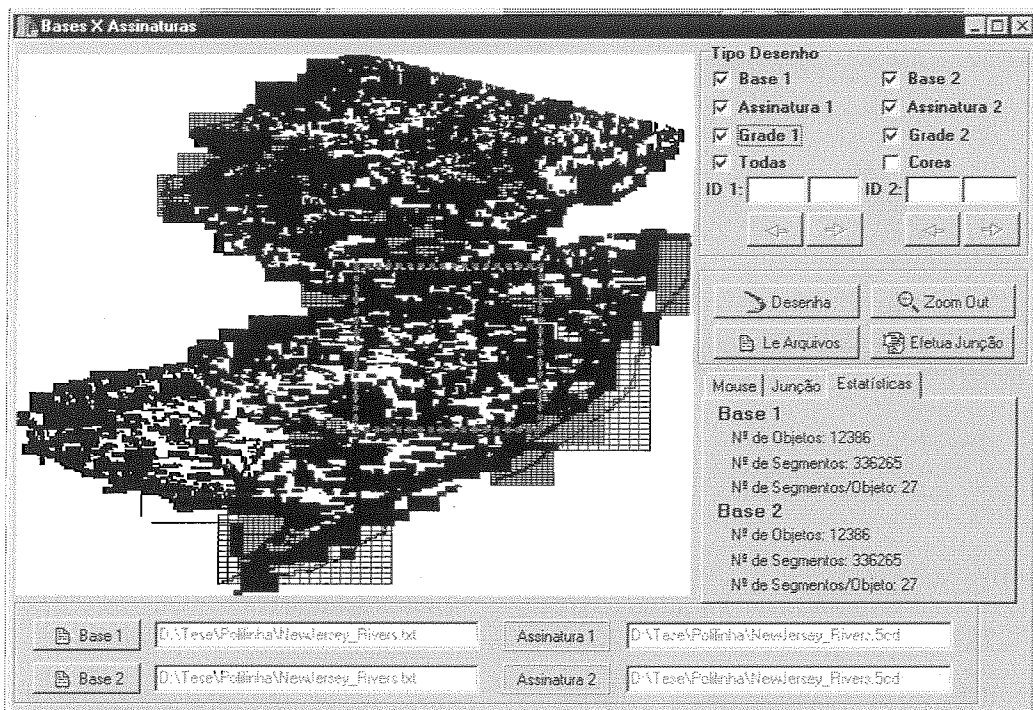


Figura E.3 - Rios de Nova Jersey com suas aproximações 5CDRS e suas grades *Raster*.

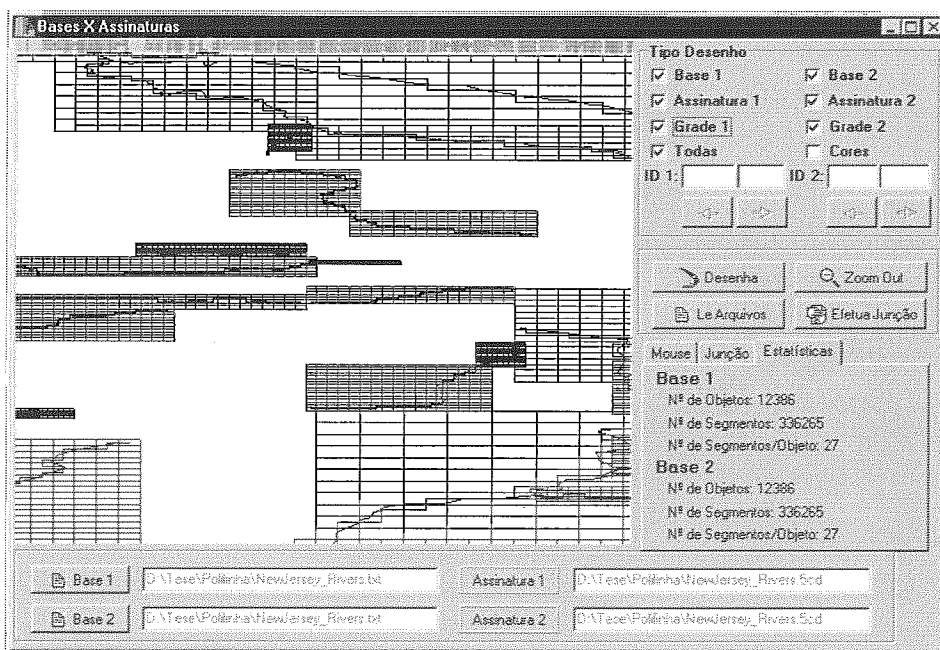


Figura E.4 - Ampliação de parte dos Rios de Nova Jersey com suas aproximações 5CDRS e suas grades *Raster*.

## ANEXO F

Neste anexo são apresentadas as telas das estradas do Canadá (Figura F.1); suas aproximações 5CDRS (Figura F.2); as estradas com suas aproximações e suas grades *raster* (Figura F.3) e uma ampliação de parte desta tela (Figura F.4).

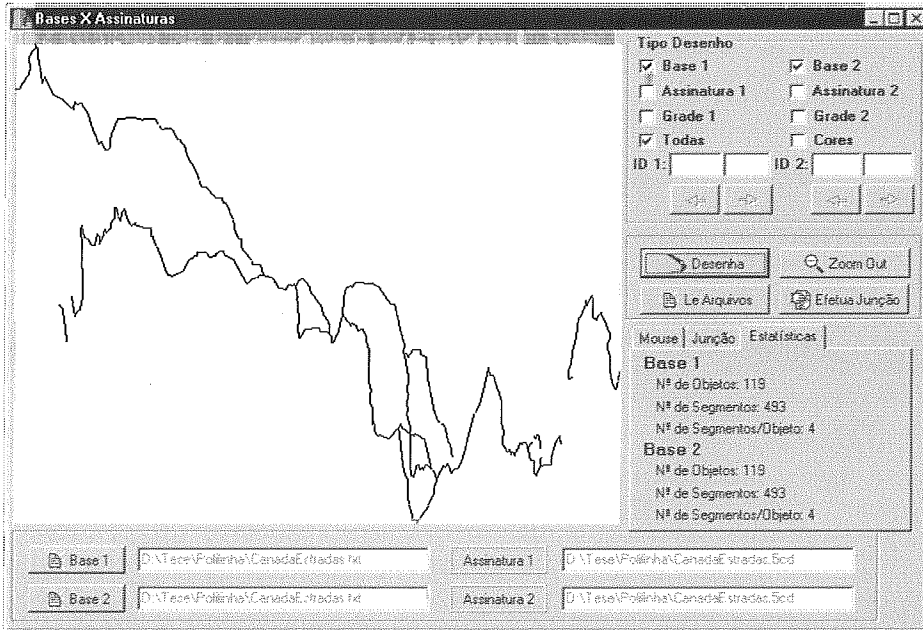


Figura F.1 - Estradas do Canadá

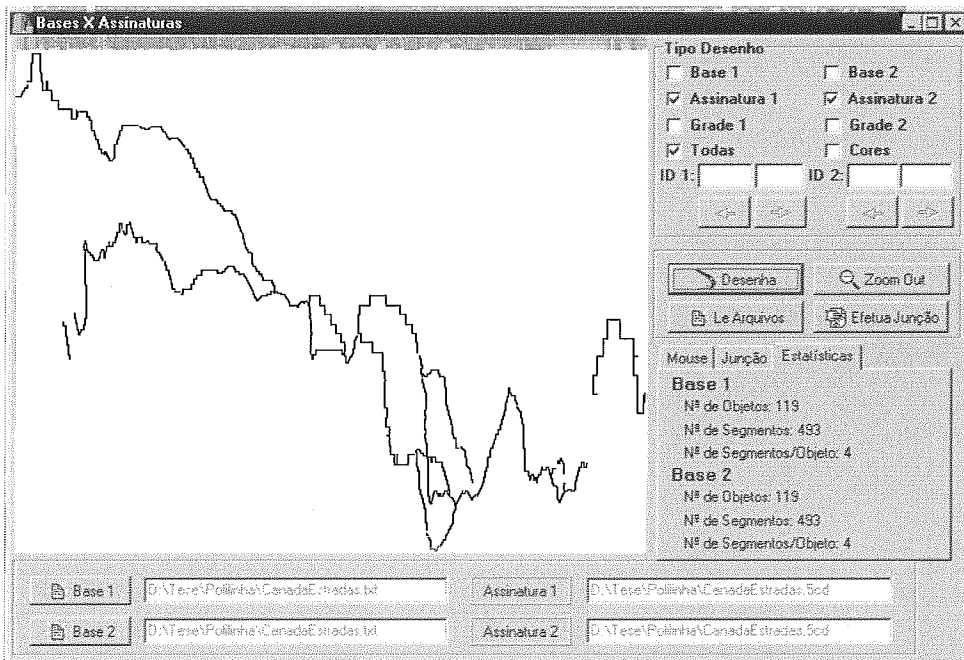


Figura F.2 - Aproximações 5CDRS das estradas do Canadá

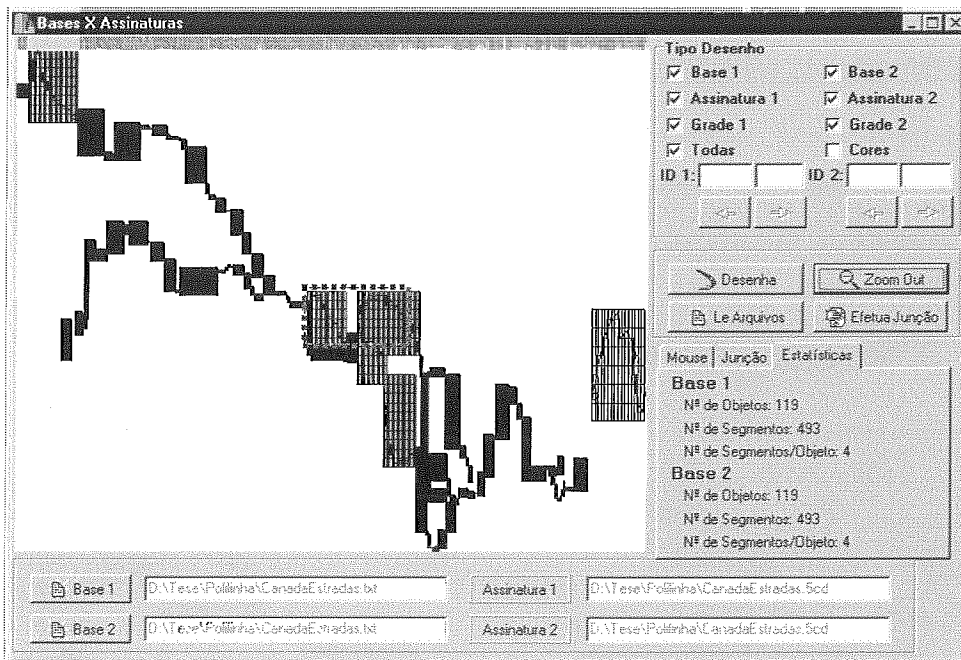


Figura F.3 - Estradas do Canadá com suas aproximações 5CDRS e suas grades Raster.

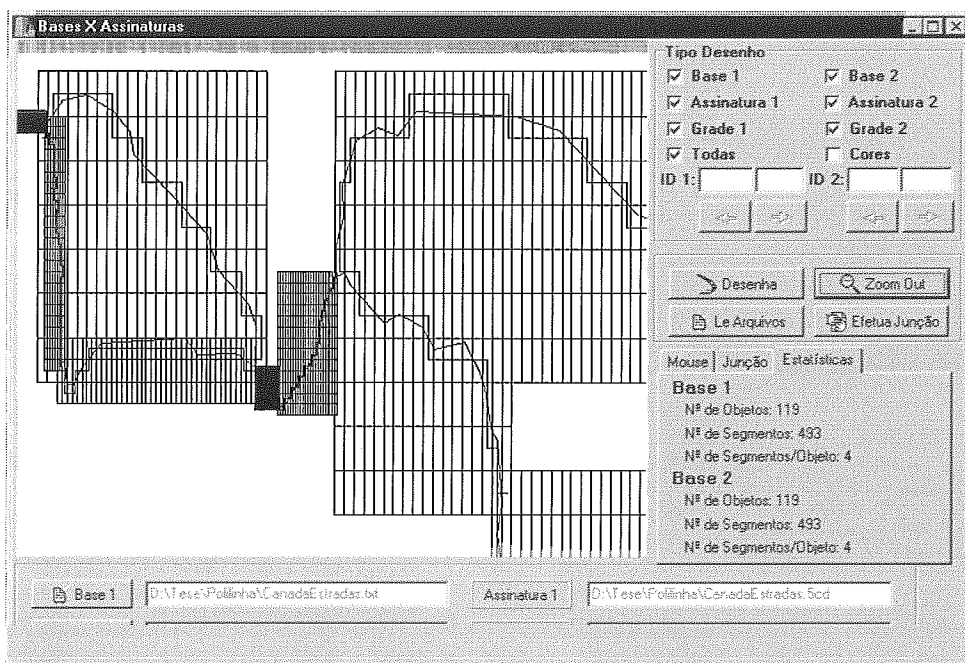


Figura F.4 - Ampliação de parte das estradas do Canadá com suas aproximações 5CDRS e suas grades Raster.

## ANEXO G

Neste anexo são apresentadas as telas dos rios do Canadá (Figura G.1); suas aproximações 5CDRS (Figura G.2); os rios com suas aproximações e suas grades *raster* (Figura G.3) e uma ampliação de parte desta tela (Figura G.4).

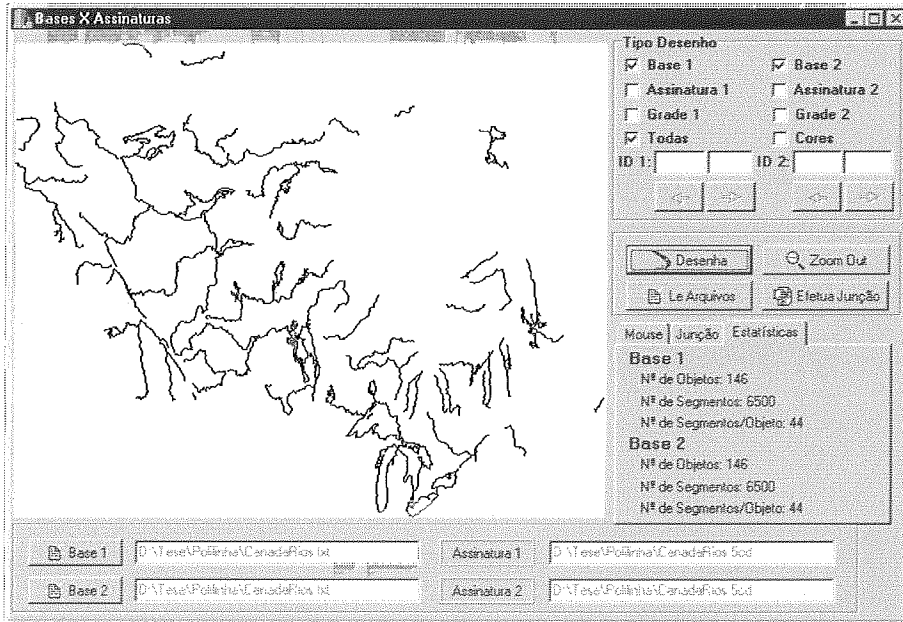


Figura G.1 - Rios do Canadá.

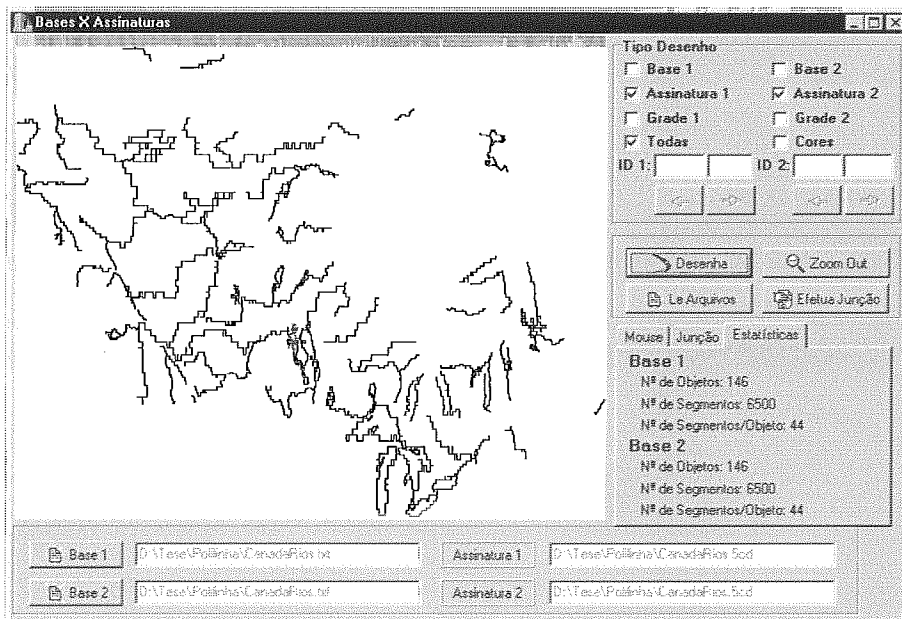


Figura G.2 - Aproximações 5CDRS dos Rios do Canadá.



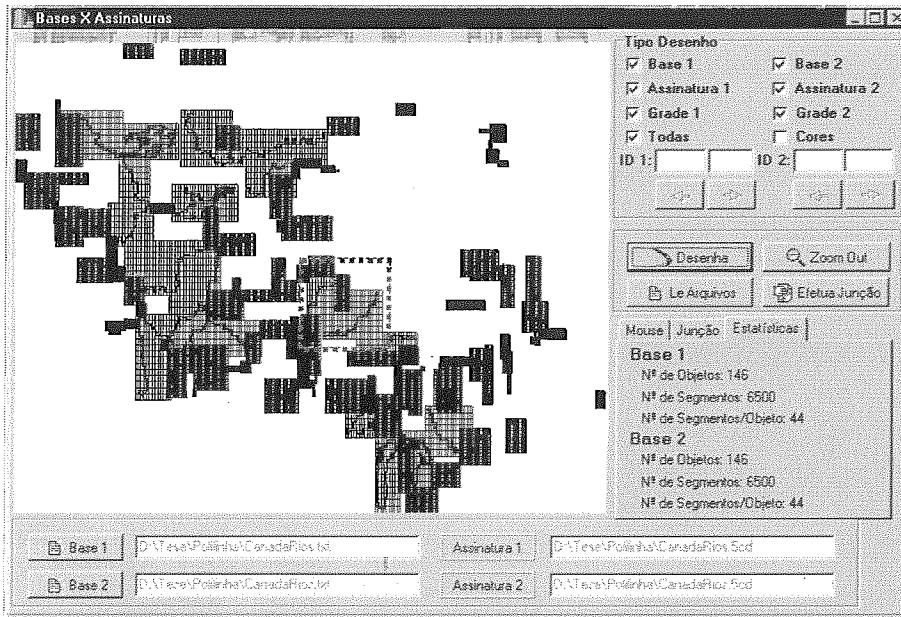


Figura G.3 - Rios do Canadá com suas aproximações 5CDRS e suas grades Raster.

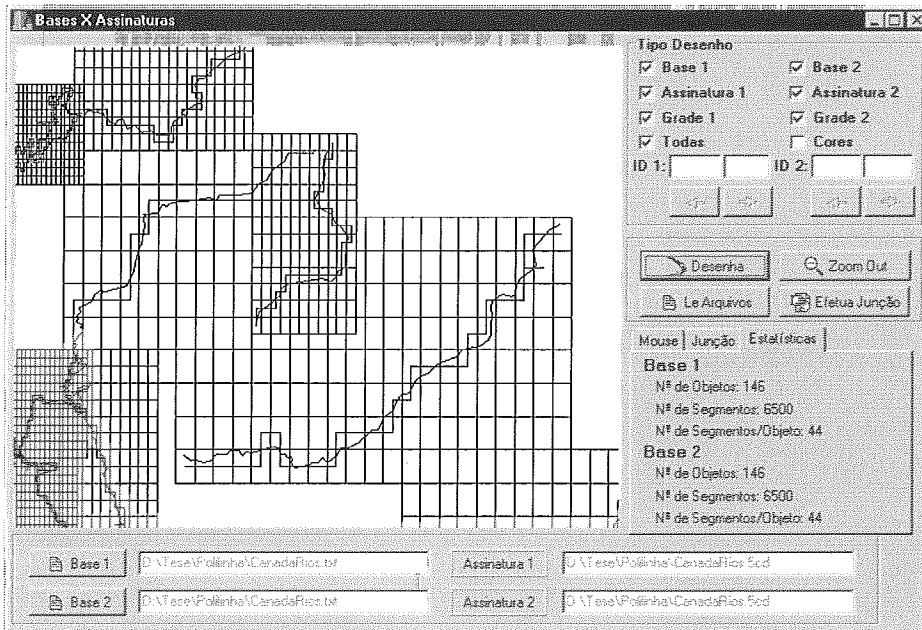


Figura G.4 - Ampliação de parte dos Rios do Canadá com suas aproximações 5CDRS e suas grades Raster.

## ANEXO H

Neste anexo são apresentadas as telas das estradas da Amazônia (Figura H.1); suas aproximações 5CDRS (Figura H.2); as estradas com suas aproximações e suas grades *raster* (Figura H.3) e uma ampliação de parte desta tela (Figura H.4).

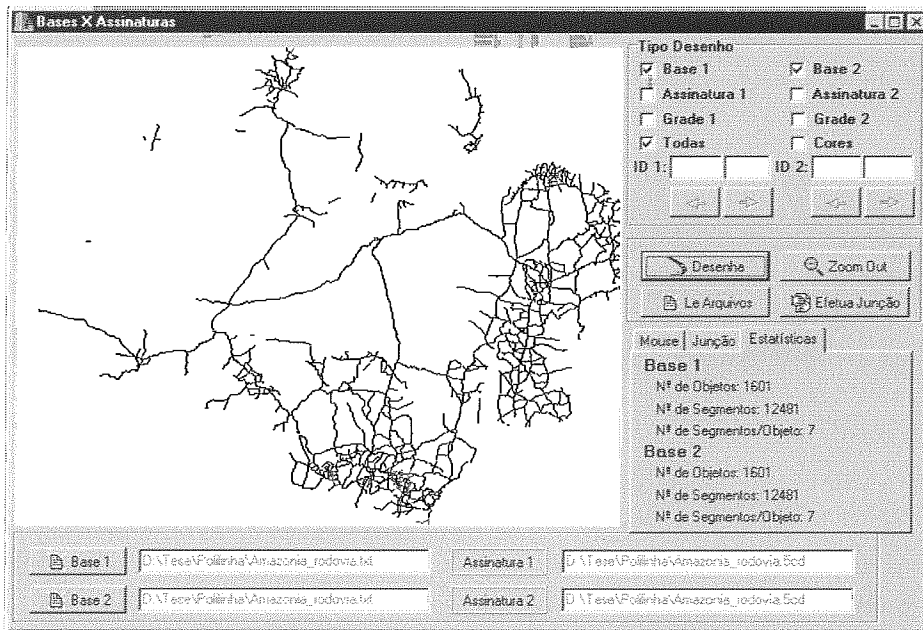


Figura H.1 - Estradas da Amazônia.

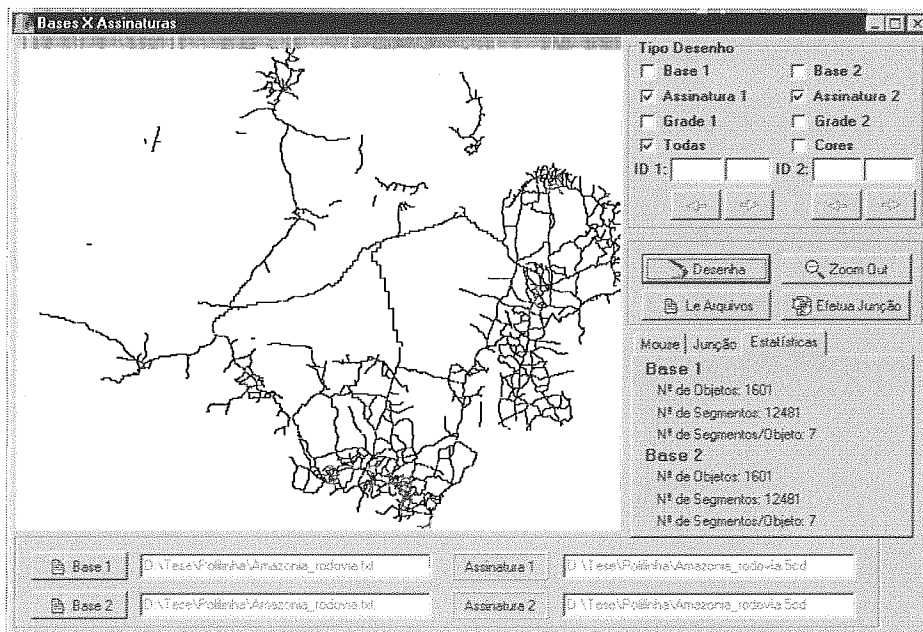


Figura H.2 - Aproximações 5CDRS das estradas da Amazônia.

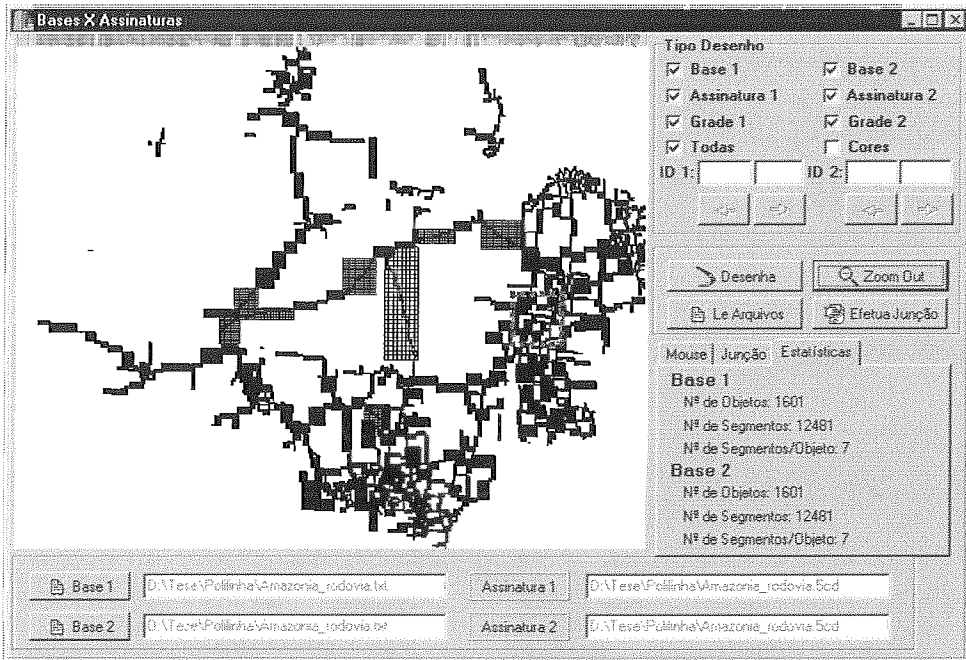


Figura H.3 - Estadas da Amazônia com suas aproximações 5CDRS e suas grades Raster.

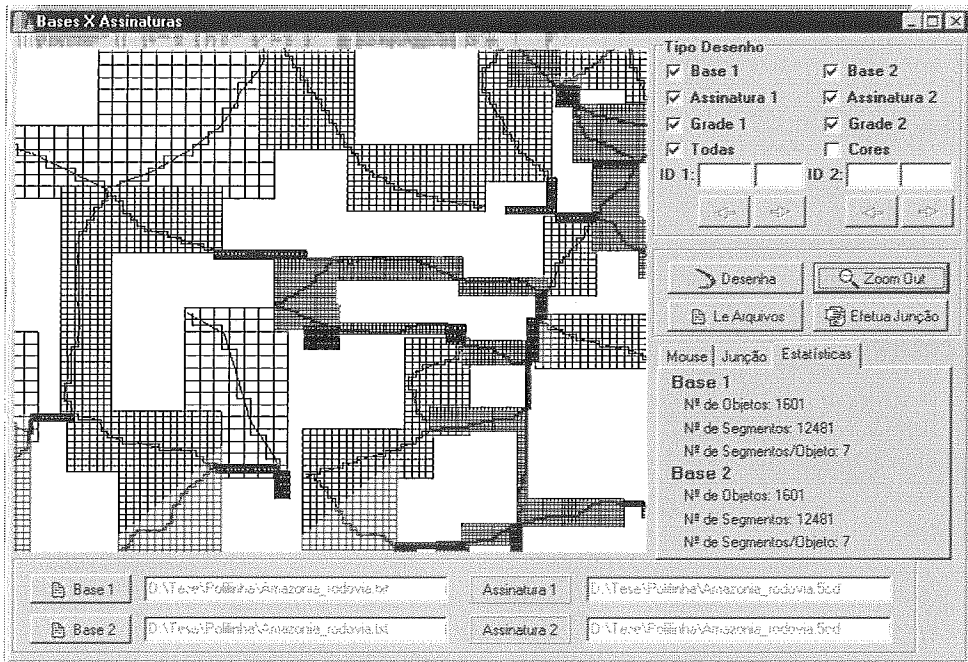


Figura H.4 - Ampliação de parte das estadas da Amazônia com suas aproximações 5CDRS e suas grades Raster.

## ANEXO I

Neste anexo são apresentadas as telas dos rios do Amazônia (Figura I.1); suas aproximações 5CDRS (Figura I.2); os rios com suas aproximações e suas grades *raster* (Figura I.3) e uma ampliação de parte desta tela (Figura I.4).

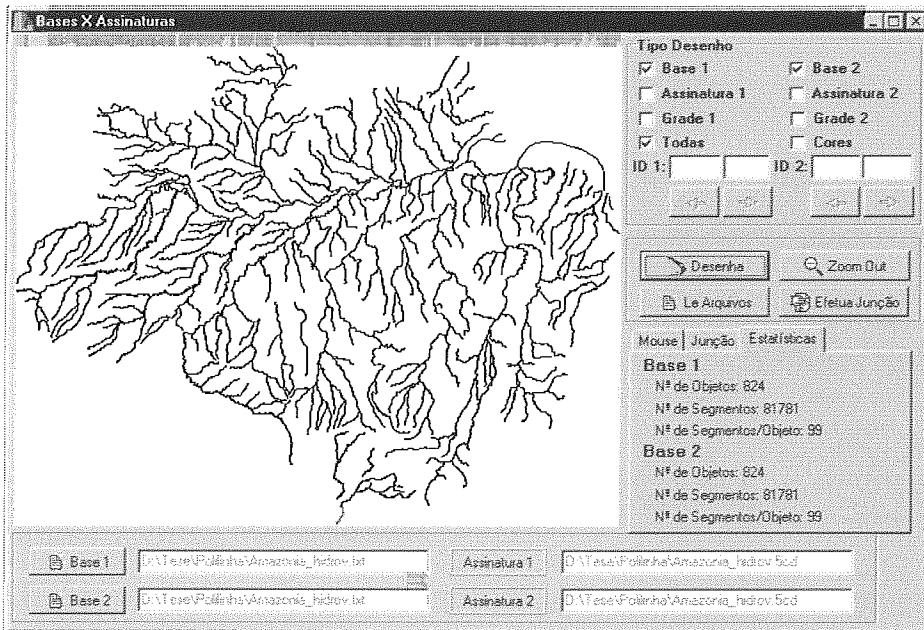


Figura I.1 - Rios da Amazônia.

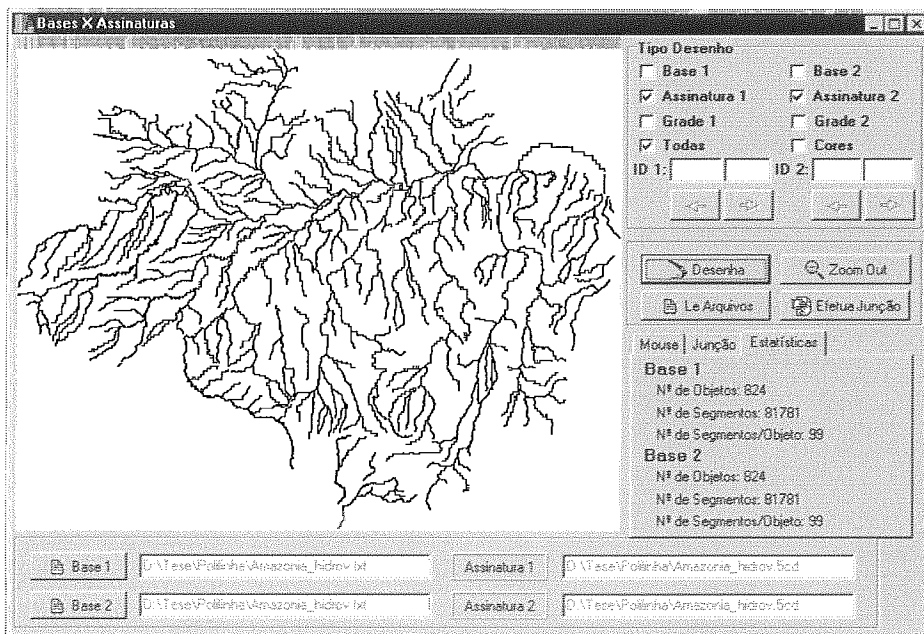


Figura I.2 - Aproximações 5CDRS dos rios da Amazônia.

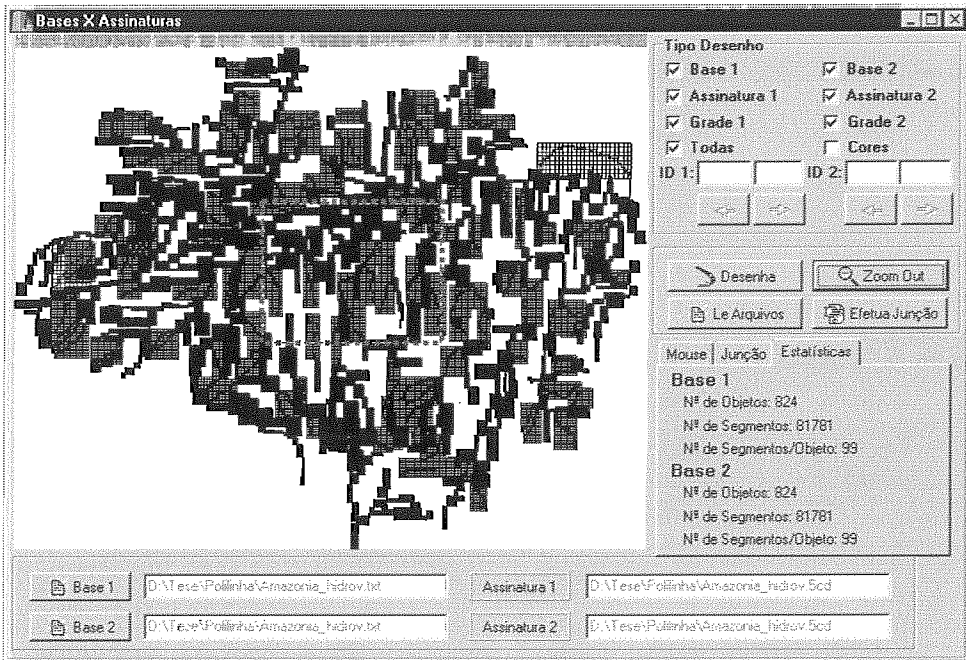


Figura I.3 - Rios da Amazônia com suas aproximações 5CDRS e suas grades *Raster*.

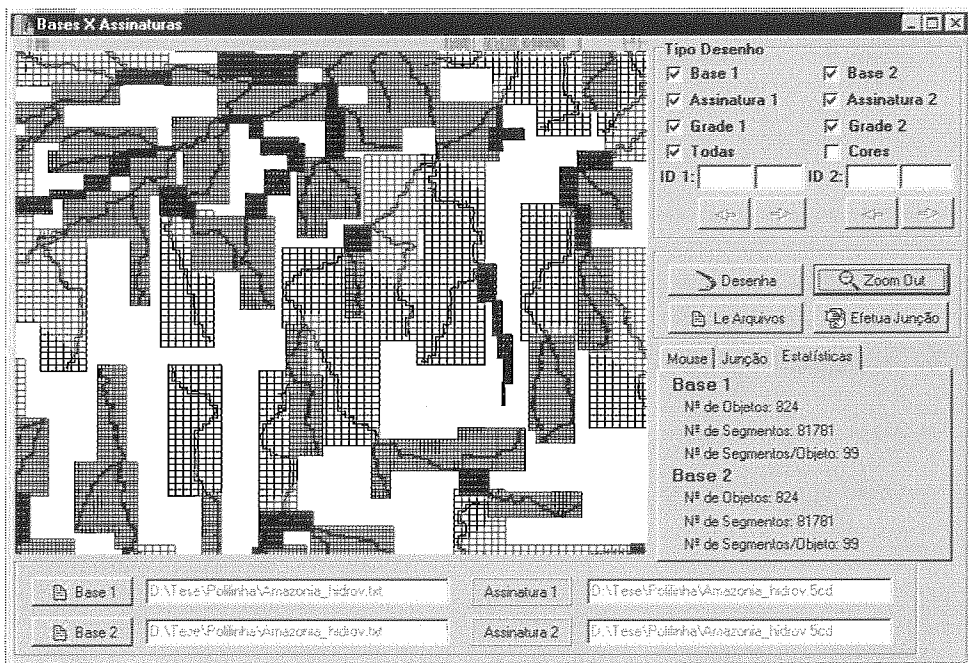


Figura I.4 - Ampliação de parte dos rios da Amazônia com suas aproximações 5CDRS e suas grades *Raster*.