FILTRO RASTER DE TRÊS CORES PARA JUNÇÃO DE OBJETOS ESPACIAIS

Rafael Brand Rodrigues

DISSERTAÇÃO 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. Geraldo Zimbrão da Silva, D.Sc.

Prof. Leonardo Guerreiro Azevedo, D.Sc.

Prof. Jano Moreira de Souza, Ph.D.

Profa. Julia Celia Mercedes Strauch, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2007

RODRIGUES, RAFAEL BRAND

Filtro Raster de Três Cores para Junção de Objetos Espaciais [Rio de Janeiro] 2007

XIII, 88 p., 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2007)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Banco de Dados Espaciais

I. COPPE/UFRJ II. Título (série)

Agradecimentos

Ao meu orientador, professor Geraldo Zimbrão da Silva, por me apoiar no desenvolvimento do trabalho, ajudando nas dúvidas mais difíceis.

Ao meu co-orientador, professor Leonardo Guerreiro Azevedo, por além de ser um grande amigo, oferecer toda ajuda e suporte, mais do que necessária e sem a qual este trabalho nunca poderia ter sido concluído. Obrigado pelo empenho e disposição para fazer com que este trabalho ficasse sempre cada vez melhor.

Ao Prof. Dr. Ralf Hartmut Güting que apesar da distância, participou da concepção da proposta desse trabalho e no artigo para o ACM-GIS.

Aos professores Jano Moreira de Souza e Julia Strauch, por aceitarem fazer parte da banca, encontrando tempo em meio a tantos compromissos importantes.

Ao professor Blaschek, Gustavo e Márcio, por me ajudarem e confiarem em mim, fazendo com que eu crescesse na vida profissional sem em nenhum momento afetar meu desempenho acadêmico.

Aos meus pais e irmãos, por confiarem em mim e acreditarem que eu poderia chegar mais longe do que eu mesmo jamais poderia imaginar.

À Ana, por todo apoio e carinho, sempre me fazendo ir para frente, me incentivando, seja por palavras, pelo exemplo pessoal, ou mesmo pela simples e fundamental presença ao meu lado.

Aos amigos e colegas, sempre dispostos a responder todas as perguntas, trocando informações e apoio, e aliviando com descontração os momentos mais cansativos.

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

FILTRO RASTER DE TRÊS CORES PARA JUNÇÃO DE OBJETOS ESPACIAIS

Rafael Brand Rodrigues

Junho / 2007

Orientadores: Geraldo Zimbrão da Silva
Leonardo Guerreiro Azevedo

Programa: Engenharia de Sistemas e Computação

A análise eficiente de consultas espaciais é um aspecto importante em sistemas de bancos de dados espaciais. Dentre as operações espaciais, a junção espacial é muito utilizada, sendo a interseção o predicado mais comum. Entretanto, o teste exato de interseção de dois objetos espaciais é o passo que mais consome tempo e Entrada / Saída.no processamento de junções espaciais. Por outro lado, o uso de aproximações pode reduzir a necessidade de examinar a geometria exata de objetos espaciais a fim de determinar os pares que se intersectam. Este trabalho propõe uma nova aproximação *raster*, chamada Assinatura Raster de 3 Cores (Three-Color Raster Signature - 3CRS), para representar diferentes tipos de dados espaciais (polígonos, polilinhas e pontos) e para ser utilizado como um filtro no segundo passo na arquitetura *Multi-Step Query Processor* (MSQP – Processamento de Consultas em Múltiplos Passos). Além disso, nós implementamos esta assinatura em um banco de dados extensível, chamado SECONDO e executamos testes experimentais em dados reais, onde os resultados demonstraram a eficiência da nossa proposta.

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

THREE-COLOR RASTER SIGNATURE FOR SPATIAL JOIN

Rafael Brand Rodrigues

June / 2007

Advisors: Geraldo Zimbrão da Silva
Leonardo Guerreiro Azevedo

Department: Computer and Systems Engineering

Efficient evaluation of spatial queries is an important issue in spatial database. Among spatial operations, spatial join is very useful, intersection being the most common predicate. However, the exact intersection test of two spatial objects is the most time-consuming and I/O-consuming step in processing spatial joins. On the other hand, the use of approximations can reduce the need for examining the exact geometry of spatial objects in order to find the intersecting ones. This work proposes a new raster approximation (Three-Color Raster Signature - 3CRS) for representing different data types (polygons, polylines and points), and to be used as filter in the second step of the Multi-Step Query Processor. We have also implemented the signature in an extensible Database System, namely SECONDO and executed experimental tests over real datasets, the results having demonstrated the effectiveness of our approach.

Índice:

1. Introdução 1
1.1. Objetivo 3
1.2. Organização do trabalho 4
2. Trabalhos relacionados 5
2.1. Arquiteturas para processamento de junção espacial 5
2.2. Filtros geométricos 8
2.2.1. Aproximações conservadoras 8
2.2.2. Aproximações progressivas 11
2.2.3. Aproximações generalizadoras 13
2.3. Assinatura Raster de Quatro Cores 13
2.4. Assinatura Raster Direcional de Cinco Cores 16
3. Secondo 19
3.1. Arquitetura 20
3.2. Assinatura de segunda ordem 21
3.3. Representação de dados espaciais 22
3.3.1. Propriedade *InsideAbove* 23
3.4. Armazenamento de dados 24
4. Assinatura Raster de 3 Cores 26
4.1. Geração da assinatura 28
4.1.1. Divisão do espaço em células 28
4.1.2. Células *inconclusivas* 31
4.1.3. Células *cheio* 36
4.2. Interseção de assinaturas 38
4.2.1. Mudança de resolução 38
4.2.2. Algoritmo de interseção 40
5. Avaliação experimental 46
5.1. Conjuntos de dados 46
5.2. Métricas utilizadas 47
5.2.1. Espaço de armazenamento 47

5.2.2. Número de acessos ao disco 48
5.2.3. Tempo de geração da assinatura 48
5.2.4. Porcentagem de pares analisados no teste exato 48
5.2.5. Tempo total de execução 49
5.3. Ambientes de teste 49
5.4. Número máximo de células 49
5.5. Resultados experimentais 49
5.5.1. Espaço de armazenamento 50
5.5.1.1. Polígono 50
5.5.1.2. Polilinha 51
5.5.1.3. Pontos 52
5.5.2. Polígono × Polígono 53
5.5.2.1. Junções utilizadas 54
5.5.2.2. Tempo total de execução 54
5.5.2.3. Acessos ao disco 55
5.5.2.4. Porcentagem de pares analisados no teste exato 56
5.5.2.5. Tempo de geração da assinatura 57
5.5.3. Polígono × Polilinha 58
5.5.3.1. Junções utilizadas 59
5.5.3.2. Tempo total de execução 59
5.5.3.3. Acessos ao disco 60
5.5.3.4. Porcentagem de pares analisados no teste exato 61
5.5.3.5. Tempo de geração da assinatura 62
5.5.4. Polígono × Pontos 64
5.5.4.1. Junções utilizadas 64
5.5.4.2. Tempo total de execução 64
5.5.4.3. Acessos ao disco 65
5.5.4.4. Porcentagem de pares analisados no teste exato 66
5.5.4.5. Tempo de geração da assinatura 67
5.5.5. Linha × Linha 68
5.5.5.1. Junções utilizadas 68

5.5.5.2. Tempo total de execução 69
5.5.5.3. Acessos ao disco 70
5.5.5.4. Porcentagem de pares analisados no teste exato 71
5.5.5.5. Tempo de geração da assinatura 72
5.5.6. Linha × Pontos 73
5.5.6.1. Junções utilizadas 73
5.5.6.2. Tempo total de execução 73
5.5.6.3. Acessos ao disco 74
5.5.6.4. Porcentagem de pares analisados no teste exato 75
5.5.6.5. Tempo de geração da assinatura 76
5.5.7. Pontos × Pontos 77
5.5.7.1. Junções utilizadas 78
5.5.7.2. Tempo total de execução 78
5.5.7.3. Acessos ao disco 79
5.5.7.4. Porcentagem de pares analisados no teste exato 80
5.5.7.5. Tempo de geração da assinatura 81
6. Conclusão 83
Referências 86

Índice de Figuras

Figura 1. Arquitetura de processamento de junção espacial em dois passos..... 6

Figura 2. Arquitetura de processamento de junção espacial em três passos..... 7

Figura 3. Exemplo de área falsa na aproximação 8

Figura 4. Exemplos de aproximações conservadoras 10

Figura 5. Exemplos de interseção de objetos avaliando a aproximação conservadora 11

Figura 6. Exemplos de aproximações progressivas 12

Figura 7. Exemplos de teste de interseção utilizando aproximações progressistas 12

Figura 8. Exemplos de aproximações generalizadoras 13

Figura 9. Exemplos de polígono representado com a aproximação 4CRS..... 15

Figura 10. Algoritmo para teste de interseção de assinaturas..... 16

Figura 11. Tipos de célula para 5CDRS: (a) horizontal e vertical; (b) horizontal; (c) vertical; (d) inconclusiva; e (e) vazia. (ZIMBRÃO *et al.*, 2000)..... 17

Figura 12. Integração dos componentes do sistema SECONDO 21

Figura 13. Exemplos de dados espaciais que podem ser armazenados no SECONDO. (a) Ponto; (b) Pontos; (c) Linha; e (d) Região 23

Figura 14. Exemplos de valores para a propriedade InsideAbove 24

Figura 15. Exemplo de polígono representado com a assinatura 3CRS..... 27

Figura 16. Exemplo de polilinha representada com a assinatura 3CRS 28

Figura 17. Alinhamento dos cantos das células (ZIMBRAO *et al.*, 1998)..... 30

Figura 18. Resultado do passo 1, dividindo o espaço em células..... 31

Figura 19. Algoritmo inicial de marcação de células Inconclusivas 32

Figura 20. Resultado do passo 2, marcando células Inconclusivas 33

Figura 21. Exemplo de matriz de troca gerada no segundo passo..... 34

Figura 22. Algoritmo de preenchimento de células *Inconclusivo* 35

Figura 23. Algoritmo de preenchimento de células cheio 36

Figura 24. Exemplo de execução do algoritmo de marcação de células cheio 37

Figura 25. Resultado do passo 3, marcando células *Cheias*..... 38

Figura 26. Todos os passos da geração da assinatura 3CRS. a) objeto original; b) objeto sobreposto à grade (etapa 1); c) células do tipo *Inconclusivo* (etapa 2); d) células *Cheio* (etapa 3); e) representação do objeto sem células *Vazio*. 38

Figura 27. Abordagem ineficiente para igualar resoluções, dividindo células..... 39

Figura 28. Abordagem utilizada na mudança de resolução, agrupando células.... 40

Figura 29. Abordagem utilizada na mudança de resolução, agrupando células.... 41

Figura 30. Algoritmo de teste de interseção de duas assinaturas 3CRS..... 42

Figura 31. Exemplo de teste de interseção de assinaturas, sem interseção 44

Figura 32. Exemplo de teste de interseção de assinaturas, com interseção..... 45

Figura 33. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação aos polígonos originais, para um máximo de 250, 500, 1000 e 1500 células..... 51

Figura 34. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação às polilinhas originais, para um máximo de 250, 500, 1000 e 1500 células..... 52

Figura 35. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação aos pontos originais, para um máximo de 250, 500, 1000 e 1500 células..... 53

Figura 36. Tempo gasto (ms) para realizar a junção entre polígonos..... 55

Figura 37. Acessos ao disco para realizar as junções polígono x polígono (Relação - % - entre arquitetura em três passos e arquitetura em dois passos) 56

Figura 38. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polígono (Relação - % - entre arquitetura em três passos e arquitetura em dois passos) 57

Figura 39. Tempo (ms) de geração da assinatura dos objetos envolvidos na junção polígono – comparação entre 3CRS e 4CRS..... 58

Figura 40. Tempo (ms) gasto para realizar as junções polígono x polilinha..... 60

Figura 41. Acessos ao disco para realizar as junções polígono x polilinha..... 61

Figura 42. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polilinha62

Figura 43. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x polilinha (ms) 63

Figura 44. Tempo total da junção (ms), considerando a geração da assinatura *on-the-fly* 64

Figura 45. Tempo gasto para realizar as junções polígono x pontos (ms) 65
Figura 46. Acessos ao disco para realizar as junções polígono x pontos 66
Figura 47. Porcentagem de pares analisados no teste exato para realizar as junções polígono x pontos 67
Figura 48. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x pontos (ms) 68
Figura 49. Tempo gasto para realizar as junções polilinha x polilinha (ms) 70
Figura 50. Acessos ao disco para realizar as junções polilinha x polilinha 71
Figura 51. Porcentagem de pares analisados no teste exato para realizar as junções polilinha X polilinha 72
Figura 52. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X polilinha (ms) 73
Figura 53. Tempo gasto para realizar as junções polilinha x pontos (ms) 74
Figura 54. Acessos ao disco para realizar as junções polilinha x pontos 75
Figura 55. Porcentagem de pares analisados no teste exato para realizar as junções polilinha X pontos 76
Figura 56. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X pontos (ms) 77
Figura 57. Tempo gasto para realizar as junções pontos x pontos (ms) 79
Figura 58. Acessos ao disco para realizar as junções pontos x pontos 80
Figura 59. Porcentagem de pares analisados no teste exato para realizar as junções pontos X pontos 81
Figura 60. Tempo de geração da assinatura dos objetos envolvidos na junção pontos X pontos (ms) 82

Índice de Tabelas

Tabela 1. Valores possíveis para as células na aproximação 4CRS 14
Tabela 2. Combinações possíveis de pares de células na assinatura 4CRS 15
Tabela 3. Valores possíveis para as células da assinatura 5CDRS 17
Tabela 4. Combinações possíveis de pares de células na assinatura 5CDRS 18
Tabela 5. Valores possíveis para as células na aproximação 3CRS 26
Tabela 6. Comparação entre objetos representados pelas assinaturas 4CRS, 5CDRS e 3CRS 27
Tabela 7. Combinações possíveis de pares de células na assinatura 3CRS 40
Tabela 8. Conjunto de dados utilizado para os testes experimentais 47
Tabela 9. Comparação do espaço de armazenamento do objeto e da assinatura para polígonos 51
Tabela 10. Comparação do espaço de armazenamento do objeto e da assinatura para polilinhas 52
Tabela 11. Comparação do espaço de armazenamento do objeto e da assinatura para pontos 53
Tabela 12. Tempo gasto para realizar a junção entre polígonos 54
Tabela 13. Acessos ao disco para realizar as junções polígono x polígono (Relação entre arquitetura em três passos e arquitetura em dois passos) 56
Tabela 14. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polígono (Relação entre arquitetura em 3 passos e arquitetura em dois passos) 57
Tabela 15. Tempo de geração da assinatura dos objetos envolvidos na junção polígono – comparação entre 3CRS e 4CRS 58
Tabela 16. Tempo gasto para realizar as junções polígono x polilinha 59
Tabela 17. Acessos ao disco para realizar as junções polígono x polilinha 60
Tabela 18. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polilinha 61

Tabela 19. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x polilinha 62

Tabela 20. Tempo gasto para realizar as junções polígono x pontos 65

Tabela 21. Acessos ao disco para realizar as junções polígono x pontos..... 66

Tabela 22. Porcentagem de pares analisados no teste exato para realizar as junções polígono x pontos 67

Tabela 23. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x pontos 68

Tabela 24. Tempo gasto para realizar as junções polilinha x polilinha..... 69

Tabela 25. Acessos ao disco para realizar as junções polilinha x polilinha 70

Tabela 26. Porcentagem de pares analisados no teste exato para realizar as junções polilinha x polilinha 71

Tabela 27. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X polilinha 72

Tabela 28. Tempo gasto para realizar as junções polilinha x pontos 74

Tabela 29. Acessos ao disco para realizar as junções polilinha x pontos..... 75

Tabela 30. Porcentagem de pares analisados no teste exato para realizar as junções polilinha x pontos 76

Tabela 31. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X pontos 77

Tabela 32. Tempo gasto para realizar as junções pontos x pontos..... 78

Tabela 33. Acessos ao disco para realizar as junções pontos x pontos 79

Tabela 34. Porcentagem de pares analisados no teste exato para realizar as junções pontos x pontos 80

Tabela 35. Tempo de geração da assinatura dos objetos envolvidos na junção pontos X pontos 81

Tabela 36. Melhor abordagem para os diferentes tipos de junção espacial 84

1. Introdução

O aumento da capacidade de armazenamento, a redução dos custos de hardware e o aumento da complexidade das aplicações permitiram que as aplicações lidem com grandes volumes de dados, envolvendo Gigabytes, Terabytes e até mesmo Petabytes de informações. Esta característica é comum em Bancos de Dados Espaciais, onde os dados geralmente têm alta complexidade e estão disponíveis em grandes volumes. 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 utilizam bancos de dados espaciais.

Dados espaciais representam objetos referentes ao espaço, constituídos de geometria, tais como pontos, linhas, regiões, volumes e até dados em dimensões maiores que incluem a dimensão tempo (SAMET, 1990). Dados espaciais representam cidades, rios, rodovias, estados, países, zonas de plantio, modelos tridimensionais de cadeias de moléculas, etc. Normalmente, dados espaciais apresentam uma geometria associada a atributos convencionais. Por exemplo, nomes de ruas, endereços, temperatura, etc. Um exemplo onde dado espacial está relacionado é a descrição de um município, que é representado por um polígono descrevendo seu contorno (geometria), nome, população e temperatura média (atributos).

Segundo GUTING (1994), sistemas de bancos de dados espaciais apresentam as seguintes características:

(1) são sistemas bancos de dados;

(2) oferecem tipos de dados espaciais; e

(3) suportam dados espaciais na sua implementação, fornecendo índices espaciais e algoritmos eficientes para junções espaciais.

O primeiro requisito enfatiza o fato de que informações espaciais estão, na prática, sempre relacionadas a informações convencionais. Portanto, sistemas de banco de dados espaciais são sistemas de bancos de dados com a capacidade adicional de trabalhar com dados espaciais. O segundo requisito prevê uma abstração fundamental para modelar a estrutura de entidades geométricas no espaço, assim como modelar relações entre entidades (por exemplo, *l* intersecta *r*), suas propriedades (por exemplo, propriedade, área de *r* - *área(r) > 1000*) e operações (por exemplo, *interseção(l, r)* – a parte de *l* que está contida

em *r*). Sem um tipo de dado espacial, um sistema não oferece suporte adequado para a modelagem de dados espaciais. Finalmente, o terceiro requisito indica que um sistema deste tipo deve ser capaz de recuperar objetos de uma grande coleção, os quais estão contidos em uma determinada região do espaço, sem percorrer o conjunto inteiro. Assim, a indexação espacial é fundamental. Além disso, o sistema deve suportar relacionar objetos de diferentes classes a partir de alguma relação espacial de um modo mais eficiente do que filtrando o produto cartesiano.

Existem várias aplicações na área de sistema de bancos de dados espaciais, como supervisão de tráfego, controle de vôos, previsão do tempo, planejamento urbano, otimização de rotas, cartografia, agricultura, administração de recursos naturais, monitoramento costeiro, controle de fogo e de epidemias, agricultura de precisão e rodovias inteligentes (ARONOFF, 1989; GORDON *et al.*, 1994; TAO *et al.*, 2003; DUCZMAL *et al.*, 2006; MEDINA *et al.*, 2006). Cada tipo de aplicação trabalha com diferentes características, escalas e propriedades espaços-temporais.

Análises eficientes de consultas espaciais representam um assunto importante em sistemas de bancos de dados espaciais. Entre as operações espaciais, junções espaciais são as mais utilizadas. Além disso, interseção é o predicado de junção mais utilizado. Muitos trabalhos indicam que o teste de geometria exata é o passo mais custoso no processamento de consultas espaciais, no que se refere tanto à Entrada e Saída (I / O) como também em processamento (CPU). BRINKHOFF *et al.* (1994b) apresentam resultados experimentais, confirmando que o teste de geometria exata, normalmente *plane-sweep* (BOISSONNAT *et al.*, 1997; FREISEISEN, 1998) é responsável pela maior parte do custo de processamento. O custo de I / O associado com o teste de geometria exata ocorre devido ao acesso à representação real dos objetos espaciais, o que pode ocupar grande espaço de armazenamento. O custo de processamento, por outro lado, é elevado devido à utilização de algoritmos complexos sobre uma quantidade grande de pontos.

Junções espaciais são amplamente estudadas na literatura e existem vários métodos para se processar operações deste tipo. Considerando pontos, polilinhas e polígonos como os três tipos de dados mais comuns em bancos de dados espaciais, existem nove classes diferentes de junções espaciais, referente à combinação de cada um destes tipos. Devido à sua utilidade e complexidade, as junções envolvendo polígonos são as mais estudadas,

enquanto a junção de pontos é menos estudada devido a sua similaridade com junções relacionais (SAMET, 1990), enquanto que existem algumas propostas para o processamento de polilinhas e junções envolvendo polígonos e polilinhas. Existem diferentes abordagens na literatura para processamento de junções espaciais. ORENSTEIN (1986) ressaltou que a maior parte destas abordagens realiza as junções em uma arquitetura de dois passos, onde o primeiro representa um método de acesso espacial para reduzir o espaço de busca e o segundo, o teste de geometria exata. Esta arquitetura é descrita em mais detalhes na seção 2.1. Para aumentar a eficiência, a organização e o estudo de índices e filtros de dados espaciais, BRINKHOFF *et al.* (1994b) propuseram uma arquitetura de processamento de junções espaciais em três passos, chamada de *Multi-Step Query Processor* (MSQP – Processamento de Consultas em Múltiplos Passos). O principal objetivo desta arquitetura é reduzir o tempo gasto no passo mais custoso. Tal redução é feita utilizando-se um passo intermediário, após o método de acesso espacial, que filtra os objetos espaciais, reduzindo o número de comparações no terceiro passo (teste de geometria exata). Esta arquitetura também é estudada de forma mais detalhada na seção 2.1.

1.1. Objetivo

Este trabalho propõe uma nova aproximação raster para ser utilizada como um filtro no segundo passo da arquitetura MSQP, envolvendo os três tipos comuns de dados espaciais (polígono, polilinha e pontos) e as diferentes combinações de junções entre eles. A assinatura proposta é chamada de Assinatura Raster de Três Cores (3CRS - Three Color Raster Signature), a qual é baseada na Assinatura Raster de Quatro Cores (4CRS - Four Color Raster Signature), proposta por ZIMBRAO *et al.* (1998).

A assinatura 3CRS tem a vantagem de ser gerada em um tempo menor, além de poder ser utilizada para representar pontos, polilinhas e polígonos, utilizando o mesmo algoritmo para processar a junção envolvendo quaisquer destes três tipos de dados. Em contrapartida, a assinatura 4CRS pode ser utilizada apenas para polígonos, o único tipo de dados previsto em seu algoritmo. Outra vantagem da proposta é que o baixo tempo de geração da assinatura permite que ela seja gerada quando for necessária (*"on the fly"*). Por exemplo, ao invés de calcular previamente a assinatura de cada objeto e armazená-la, ela

pode ser gerada apenas quando necessário, economizando espaço de armazenamento. Os testes experimentais realizados neste trabalho comprovam este fato.

Com o intuito de avaliar a eficiência da nossa proposta, realizamos dois conjuntos de testes experimentais:

(1) comparação entre o processamento de junções espaciais utilizando-se a assinatura 3CRS (processamento em três passos) contra o processamento de junções espaciais sem o uso de assinaturas (processamento em dois passos); e

(2) comparação entre o processamento de junções espaciais utilizando-se a assinatura 3CRS contra o processamento utilizando-se a assinatura 4CRS (ambos processamentos em três passos).

Em nossos testes experimentais empregamos conjuntos de dados reais e utilizamos o SECONDO (GÜTING *et al.*, 2000; GÜTING *et al.*, 2005) como ambiente de execução dos testes. O SECONDO é um sistema de banco de dados extensível apropriado tanto para a implementação de protótipos experimentais como para lecionar conceitos de banco de dados. Os testes experimentais que executamos demonstraram a eficiência da nossa proposta.

1.2. Organização do trabalho

Este trabalho está organizado da seguinte forma: o capítulo 1 é a presente introdução. O capítulo 2 descreve o conceito de junção espacial, apresentando trabalhos propostos na literatura para solucionar o problema. O capítulo 3 apresenta o sistema SECONDO, utilizado como ambiente para implementar e avaliar a assinatura 3CRS. No capítulo 4 a proposta é descrita em maiores detalhes, no qual apresentamos a geração e utilização da assinatura. Os resultados dos testes experimentais utilizados para avaliar a proposta são apresentados no capítulo 5. As conclusões e considerações finais são apresentadas no capítulo 6.

2. Trabalhos relacionados

Uma das operações mais comuns em bancos de dados espaciais é a junção espacial. Como definido em BRINKHOFF *et al.* (1994a), junção espacial consiste do processamento de dois conjuntos de dados espaciais, através de uma operação. Exemplos de operações utilizadas em junções espaciais são a união, a diferença e, mais freqüentemente, a interseção. A junção espacial tem como entrada dois conjuntos de objetos espaciais e produz como saída um subconjunto que é composto pelos pares de objetos dos conjuntos de dados originais que, combinados, atendam a um predicado espacial.

2.1. Arquiteturas para processamento de junção espacial

Existem muitas abordagens para o processamento de operações de junção espacial. ZHU *et al.* (2000) enfatizam que os métodos tradicionais processam a junção espacial em dois passos. ORENSTEIN (1986) e KOTHURI *et al.* (2001) propuseram algoritmos eficientes para serem utilizados no segundo passo. No método de dois passos, apresentado na Figura 1, o primeiro passo emprega o Método de Acesso Espacial (Spatial Access Method – SAM) para reduzir o espaço de busca. O menor retângulo envolvente (Minimum Bounding Rectangle – MBR) normalmente é utilizado neste primeiro passo. Este passo não tem como saída o resultado da operação de junção. Ao invés disso, provê um conjunto de pares candidatos, que contém o conjunto solução. Este superconjunto é repassado para o segundo passo. O segundo passo é um passo de refinamento, onde os pares resultantes do primeiro passo são lidos do disco e têm sua geometria processada. Este é o passo mais custoso, exigindo muito tempo de Entrada e Saída (I / O) para buscar e ler os objetos espaciais do disco, além de exigir tempo de processamento (CPU) para avaliar o predicado espacial sobre os objetos reais retornando a resposta exata para a consulta.

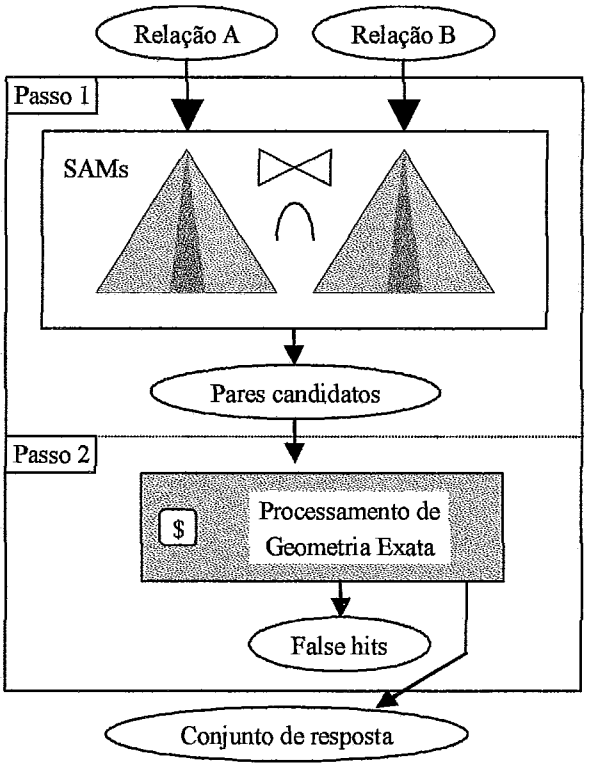


Figura 1. Arquitetura de processamento de junção espacial em dois passos

BRINKHOFF *et al.* (1994b) propuseram uma arquitetura em três passos para o processamento de junções espaciais, chamada de Multi-Step Query Processor (MSQP), apresentada na Figura 2. Nesta arquitetura, um passo adicional foi incluído entre o primeiro (SAM) e o segundo passo (processamento de geometria exata). O passo proposto consiste em comparar os pares candidatos resultantes do primeiro passo, utilizando-se um filtro geométrico. Um filtro geométrico utiliza uma representação compacta e aproximada do objeto mantendo suas características principais. Na seção 2.2 apresentamos em detalhes filtros geométricos. Como resultado deste passo, existem três possibilidades: pares que pertencem à solução (aceito ou *hit*), pares que não pertencem à solução (rejeitados ou *false hits*) e pares onde não se é possível obter uma resposta conclusiva (*inconclusivos*). Estes últimos pares são repassados para o terceiro passo (passo de processamento de geometria exata), onde os objetos são lidos do disco e têm suas geometrias exatas processadas, resultando em uma resposta definitiva.

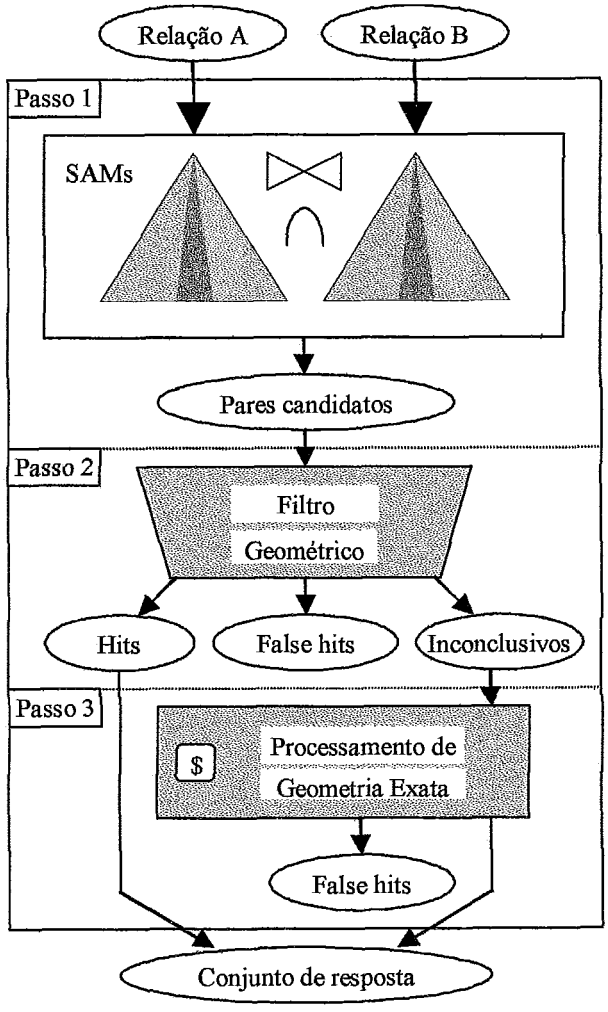


Figura 2. Arquitetura de processamento de junção espacial em três passos

Existem duas vantagens principais em se introduzir o passo de filtro. Primeiramente, o tamanho de uma representação aproximada é apenas uma fração do tamanho do objeto espacial; assim, ela pode até mesmo ser armazenada no índice, juntamente com o MBR do objeto. Dessa forma, se ao compararmos os filtros de dois objetos o resultado for *aceito,* a consulta requer um número menor de Entrada / Saída. A segunda vantagem é que o teste de duas aproximações requer um tempo de processamento muito menor do que o teste exato das representações reais dos objetos. Como os pares onde se obtém uma resposta conclusiva no segundo passo não são enviados para o terceiro passo, o tempo total da consulta é reduzido significativamente.

2.2. Filtros geométricos

Existem vários tipos de aproximações de objetos possíveis para o segundo passo da arquitetura MSQP. Segundo BRINKHOFF *et al.* (1993), as aproximações podem ser divididas em três categorias, de acordo com a sua relação com o objeto que representam: conservadoras, progressivas e generalizadoras. Nas próximas seções apresentaremos os tipos de aproximações em mais detalhes.

2.2.1. Aproximações conservadoras

As aproximações conservadoras são superconjuntos dos objetos originais, ou seja, todos os pontos do objeto original estão contidos na aproximação. A Figura 4 apresenta alguns exemplos de aproximações conservadoras. A Figura 4a mostra o objeto espacial que será utilizado nos exemplos. Na Figura 4b é apresentado o Retângulo Envolvente Mínimo (Minimum Bounding Rectangle – MBR). Esta é a aproximação conservadora mais comumente utilizada, sendo o objeto real representado pelo menor retângulo, alinhado pelos eixos, que o envolve completamente. O amplo uso desta aproximação se deve ao baixo custo de processamentos para a geração e para a comparação com outras aproximações do mesmo tipo.

A qualidade de uma aproximação conservadora é avaliada de acordo com sua área falsa, a qual corresponde à parte da aproximação que não é preenchida pelo objeto. Quanto menor a área falsa melhor é a aproximação. A Figura 3 apresenta um exemplo de aproximação, indicando a área falsa. A Figura 3a apresenta o objeto original; a Figura 3b mostra a aproximação por MBR, sobreposta ao objeto para facilitar a visualização; a Figura 3c mostra, em cinza, a área falsa neste exemplo.

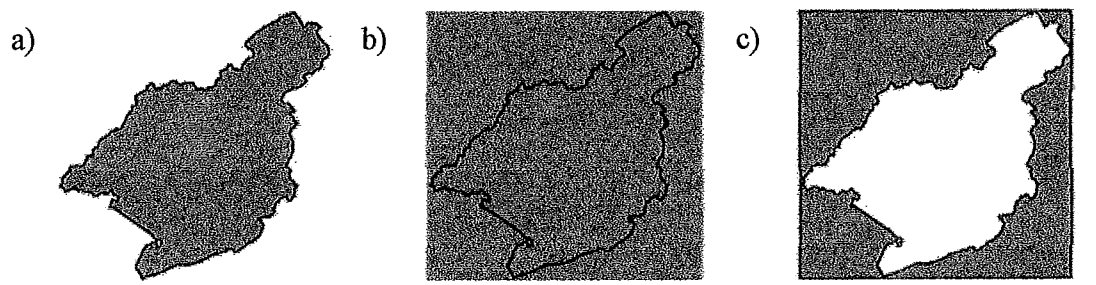


Figura 3. Exemplo de área falsa na aproximação

A Figura 4c é uma variação do MBR, onde o menor retângulo não necessariamente é alinhado com os eixos. Esta aproximação é conhecida como RMBR (Rotated Minimum Bounding Rectangle) e apresenta uma área falsa menor que a primeira alternativa, embora o custo de geração e comparação da mesma seja um pouco mais elevado. A Figura 4d e a Figura 4e apresentam, respectivamente, a aproximação por círculo envolvente mínimo (MBC – Minimum Bounding Circle) e por elipse envolvente mínima(MBE – Minimum Bounding Elipse). Uma apresentação que reduz bastante a área extra está representada na figura Figura 4f, o menor polígono convexo contendo o objeto (Convex Hull – CH). A desvantagem desta aproximação, entretanto é o alto custo de geração e a complexidade de comparação com outras aproximações. Muitas vezes este tipo de aproximação pode ser quase tão complexa quanto o objeto original. Outra forma de aproximação é através de polígonos de N lados. Este tipo de aproximação tem menor custo de geração do que o CH, porém apresenta uma área não utilizada maior que o CH. A Figura 4g e Figura 4h mostram exemplos deste tipo de aproximação, com 4 e 5 lados, respectivamente.

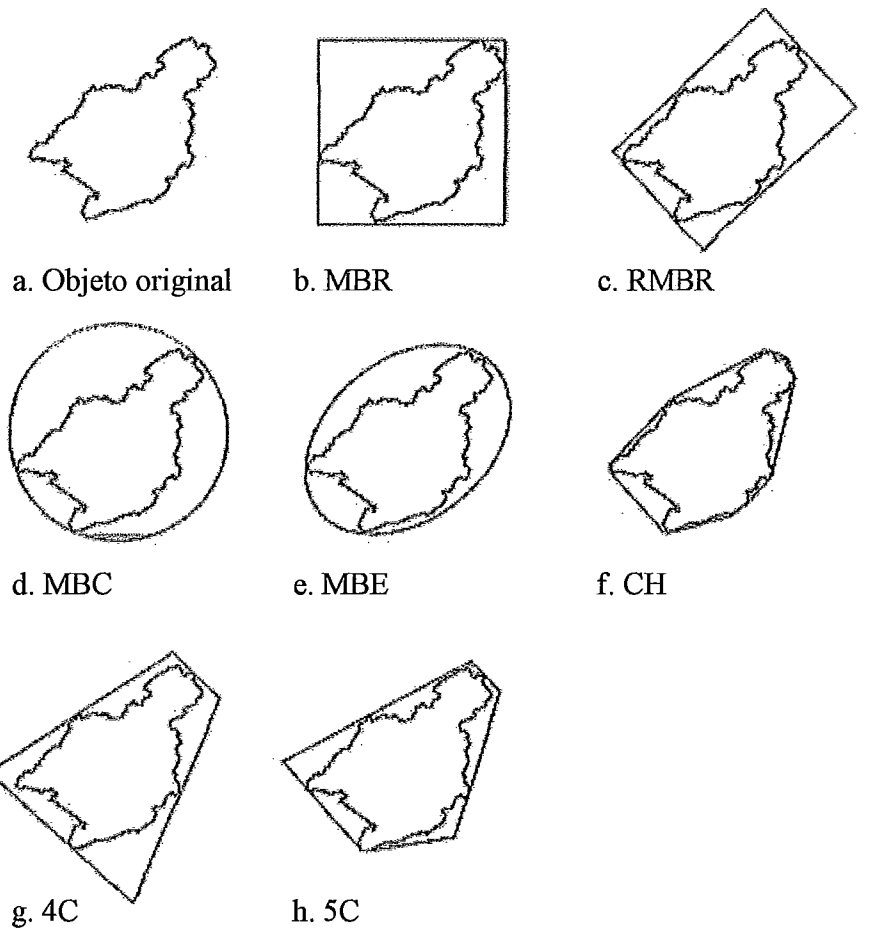


Figura 4. Exemplos de aproximações conservadoras

A vantagem das aproximações conservadoras para a operação de interseção é que se as aproximações não têm interseção, então os objetos por elas representados também não se interceptam. Um exemplo deste caso encontra-se na Figura 5a. O contrário, entretanto, não é sempre verdadeiro: se duas aproximações conservadoras têm interseção, então não necessariamente os objetos têm interseção, conforme demonstram os exemplos das Figura 5b e Figura 5c. Na Figura 5b, apesar das aproximações possuírem interseção, os objetos não interceptam. No exemplo da Figura 5c, tanto a assinatura quanto os objetos possuem interseção. Todavia, não podemos afirmar que os objetos interceptam, pois não podemos afirmar que há interseção comparando as aproximações. Isto ocorre porque estas aproximações possuem uma área que não contém nenhum ponto do objeto original. Se a interseção ocorrer apenas nesta área, então as aproximações têm interseção, mas os objetos

originais não. Assim, ao se utilizar uma aproximação conservadora podemos identificar pares que não possuem interseção (*false hits*). Se o resultado do teste das aproximações não for *não há interseção,* então deve ser aplicado o teste de geometria exata sobre o par de objetos (terceiro passo do MSQP).

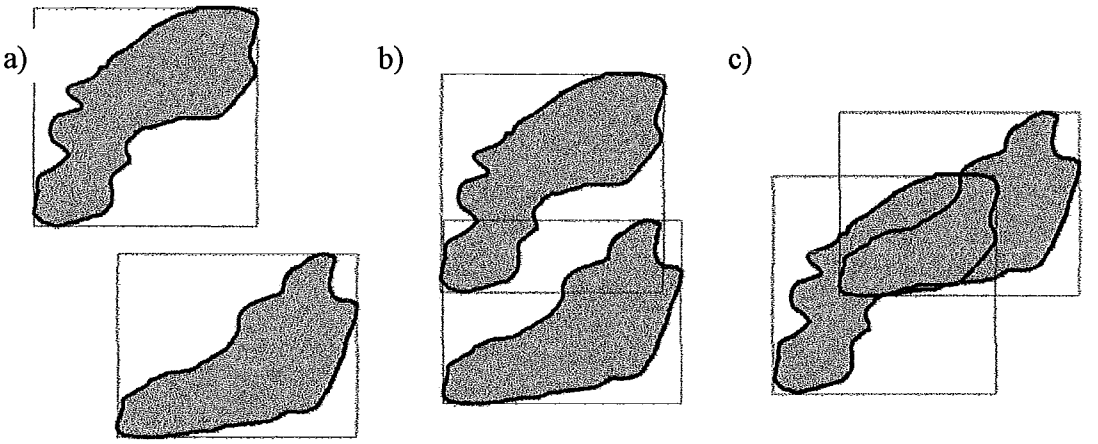


Figura 5. Exemplos de interseção de objetos avaliando a aproximação conservadora

2.2.2. Aproximações progressivas

A segunda categoria de aproximações descrita por BRINKHOFF *et al.* (1993) são as aproximações progressivas. Este tipo de aproximação representa o inverso das aproximações conservadoras: um polígono está progressivamente aproximado quando todos os pontos da aproximação estão contidos no objeto original. A Figura 6 apresenta dois exemplos de aproximações progressivas do mesmo objeto exemplificado na Figura 4a. Na Figura 6a foi utilizada a aproximação por maior retângulo contido; na Figura 6b foi utilizada a aproximação de maior círculo contido.

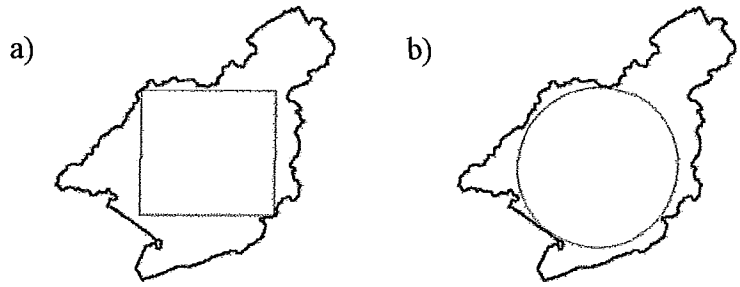


Figura 6. Exemplos de aproximações progressivas

Enquanto as áreas das aproximações conservadoras (Seção 2.2.1) são maiores ou iguais às áreas dos objetos representados, nas aproximações progressistas as áreas das aproximações são menores ou iguais às áreas dos objetos representados. Analogamente, quando um teste de interseção entre aproximações progressivas obtiver como resultado *possui interseção*, então podemos garantir que os objetos representados pelas mesmas têm interseção. Por outro lado, se o resultado não for *possui interseção*, então não se pode garantir que os objetos têm interseção. A Figura 7 apresenta três exemplos de comparação de objetos utilizando-se a aproximação progressista. Na Figura 7a, tanto os objetos quanto as aproximações não possuem interseção; na Figura 7b, os objetos possuem interseção, mas as aproximações possuem; e na Figura 7c, ambos (objetos e aproximações) possuem interseção.

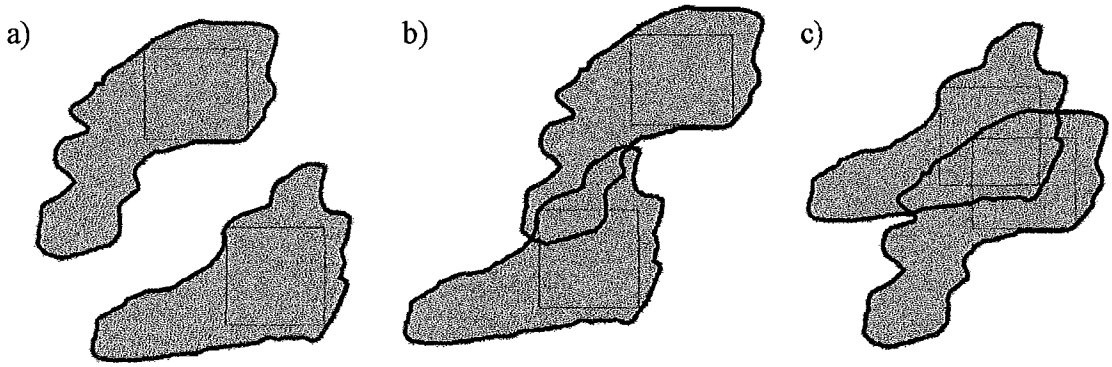


Figura 7. Exemplos de teste de interseção utilizando aproximações progressistas

2.2.3. Aproximações generalizadoras

A terceira categoria de aproximações é a generalizadora. Este tipo de aproximação tenta simplificar o objeto original, por exemplo, reduzindo o número de vértices. Geralmente não há uma relação topológica entre a aproximação e o objeto original, ou seja: nem o objeto original está completamente contido na aproximação, nem a aproximação está completamente contida no objeto. Três exemplos desta categoria estão representados na Figura 8. A Figura 8a apresenta um exemplo de aproximação generalizadora por redução de vértices; a Figura 8b mostra a representação do objeto por Assinatura Raster de Quatro Cores (a qual é apresentada em detalhes na seção 2.3); e na Figura 8c, a Assinatura Raster de Três Cores (que é abordada em maiores detalhes na seção 4).

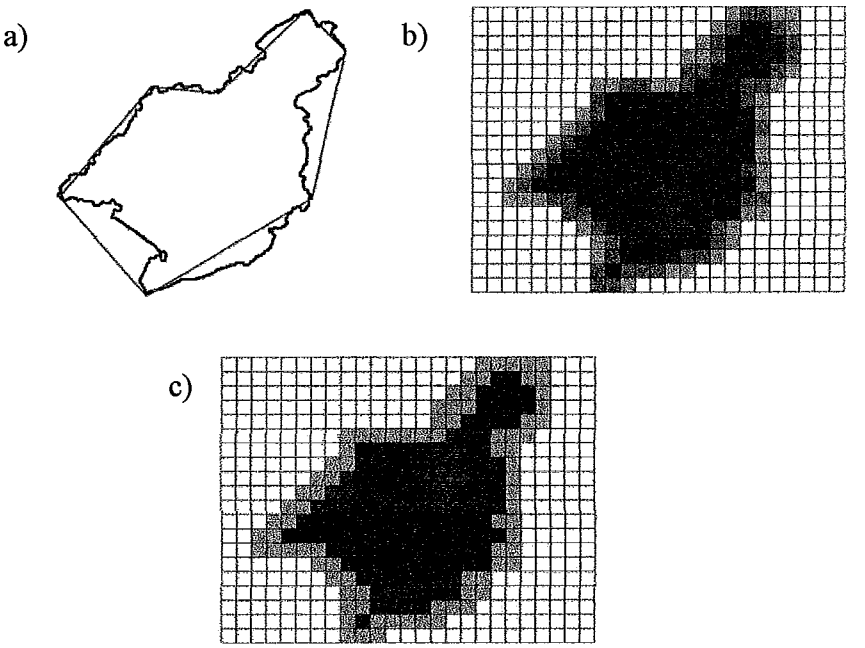


Figura 8. Exemplos de aproximações generalizadoras

2.3. Assinatura Raster de Quatro Cores

ZIMBRAO *et al.* (1998) propuseram a assinatura raster de quatro cores (Four Color Raster Signature – 4CRS), a qual pode ser utilizada para representar polígonos. Assinaturas 4CRS podem ser comparadas através da operação de interseção e podem ser utilizadas no segundo passo da arquitetura MSQP.

A assinatura 4CRS é composta de uma grade de células, onde cada célula representa um dentre quatro valores (ou cores) possíveis, conforme apresentado na Tabela 1. Cada valor representa um tipo de interseção entre o objeto e a célula. A quantidade e o tamanho das células pode variar, de acordo com a precisão exigida e o espaço utilizado para armazenamento e o tempo de processamento das assinaturas. Quanto mais células (menor tamanho de células), maior a precisão da assinatura e maior o espaço de armazenamento necessário, além de ser maior o tempo para geração e comparação das assinaturas. Um exemplo de um polígono representado por uma aproximação 4CRS, com dois tamanhos de células diferentes é apresentado na Figura 9.

Tabela 1. Valores possíveis para as células na aproximação 4CRS

Valor	Significado
Cheio	A célula está totalmente contida no interior do polígono
Muito	Mais de 50% da área da célula tem interseção com o polígono
Pouco	Até 50% da área da célula tem interseção com o polígono
Vazio	Nenhum ponto do polígono está dentro da célula

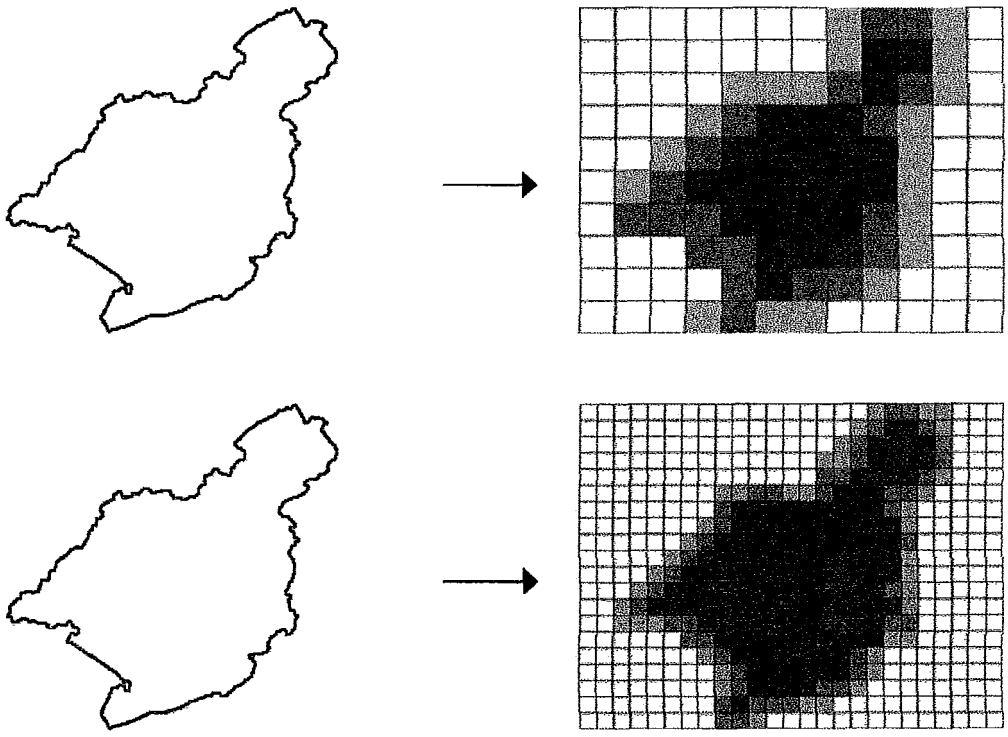


Figura 9. Exemplos de polígono representado com a aproximação 4CRS

A interseção de duas assinaturas é calculada comparando-se as células da assinatura de um objeto com as células correspondentes (que se sobrepõem) na assinatura do segundo objeto. Desta forma, existem quatro combinações de células possíveis. Cada combinação gera um resultado parcial de interseção, como indicado na Tabela 2.

Tabela 2. Combinações possíveis de pares de células na assinatura 4CRS

	Cheio	Muito	Pouco	Vazio
Cheio	*Hit*	*Hit*	*Hit*	Não há interseção
Muito	*Hit*	*Hit*	Inconclusivo	Não há interseção
Pouco	*Hit*	Inconclusivo	Inconclusivo	Não há interseção
Vazio	Não há interseção	Não há interseção	Não há interseção	Não há interseção

Apenas as células na área de interseção dos MBR são comparadas. Se a comparação de pelo menos um par de células gerar um resultado parcial de *Hit*, então o teste de interseção retorna como resultado final *Hit*. Se todas as células comparadas obtiverem resultado parcial *Não há interseção*, então não existe interseção entre as assinaturas e logo

não existe interseção entre os objetos. O último resultado possível é quando nenhum par gerou resultado parcial de *Hit* e pelo menos um dos pares de células retornou resultado parcial *Inconclusivo*. Neste caso, o resultado final é inconclusivo, ou seja, não é possível avaliar se os objetos têm interseção baseando-se em suas assinaturas. Os polígonos são então encaminhados para um passo de análise da geometria exata.

```
resultado = NAO_HA_INTERSECACO;
Para cada célula dentro da área de interseção dos MBR
      resultadoParcial = comparaCelulaCorrespondente
      se resultadoParcial = INTERSECAO
            retornar INTERSECAO;
      senão
            se resultadoParcial = INCONCLUSIVO
                  resultado = INCONCLUSIVO;
retornar resultado
```

Figura 10. Algoritmo para teste de interseção de assinaturas

2.4. Assinatura Raster Direcional de Cinco Cores

Outro filtro geométrico generalizador, proposto por ZIMBRÃO *et al.* (2000) é a Assinatura Raster Direcional de Cinco Cores (5 Color Directional Raster Signature – 5CDRS), utilizada para representar polilinhas. Cada célula da assinatura armazena os tipos de interseção da polilinha com a célula. Desta forma, cada célula pode assumir 1 dentre 5 valores (cores) possíveis, conforme a Tabela 3. A Figura 11 apresenta exemplos de cada uma destas células.

Tabela 3. Valores possíveis para as células da assinatura 5CDRS

Valor	Significado
Horizontal e Vertical	Polilinha intersecta a célula horizontal e verticalmente
Horizontal	Polilinha intersecta a célula apenas horizontalmente
Vertical	Polilinha intersecta a célula apenas verticalmente
Inconclusivo	Outros tipos de interseção
Vazio	Polilinha não intersecta a célula

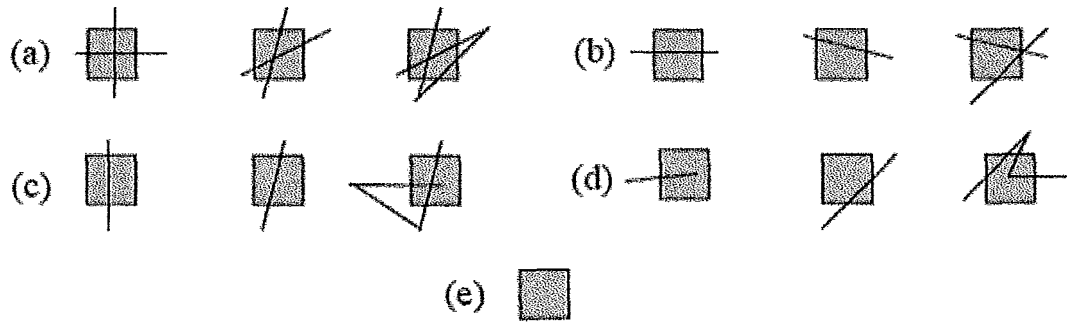


Figura 11. Tipos de célula para 5CDRS: (a) horizontal e vertical; (b) horizontal; (c) vertical; (d) inconclusiva; e (e) vazia. (ZIMBRÃO *et al.*, 2000)

Analogamente à assinatura 4CRS, duas assinaturas podem ser comparadas e cada par de células apresenta um resultado parcial, conforme indicado na Tabela 4. Como na comparação de assinaturas 4CRS, um resultado parcial de *hit* indica que as polilinhas se intersectam. Se tal resultado não for encontrado e todas os resultados parciais indicarem que não há interseção, então não há interseção entre as polilinhas. Se nenhum resultado parcial indicar *hit* e pelo menos um deles indicar inconclusivo, então não é possível identificar se as duas polilinhas se intersectam.

Tabela 4. Combinações possíveis de pares de células na assinatura 5CDRS

	Vazia	Inconclusiva	Vertical	Horizontal	Horizontal e Vertical
Vazia	Não há interseção	Não há interseção	Não há interseção	Não há interseção	Não há interseção
Inconclusiva	Não há interseção	Inconclusivo	Inconclusivo	Inconclusivo	Inconclusivo
Vertical	Não há interseção	Inconclusivo	Inconclusivo	*Hit*	*Hit*
Horizontal	Não há interseção	Inconclusivo	*Hit*	Inconclusivo	*Hit*
Horizontal e Vertical	Não há interseção	Inconclusivo	*Hit*	*Hit*	*Hit*

3. Secondo

Sistemas de bancos de dados relacionais convencionais não oferecem todos os requisitos necessários para aplicações não-convencionais, tais como sistemas de CAD (Computer Aided Design – desenho auxiliado por computador), GIS (Geographic Information System – Sistemas de Informação Geográfica), ou sistemas de processamento de multimídia. Com isso, novos sistemas de bancos de dados surgiram, com a capacidade de se estenderem para solucionar problemas específicos.

A primeira abordagem para estender bancos de dados relacionais através de tipos de dados abstratos foi proposta por ONG *et al.* (1983), que tinha como objetivo principal a otimização de consultas em tipos de dados abstratos e suporte a objetos complexos. Outro importante projeto que implementa um sistema relacional extensível foi Starbust (SCHWARZ *et al.*, 1986) que tem como principais objetivos o processamento de consultas e a arquitetura para armazenamento e indexação de objetos complexos. Mais recentemente, sistemas extensíveis comerciais surgiram, conhecidos como sistemas objeto-relacionais (CAREY *et al.*, 1996). Um exemplo deste tipo de sistema é o Informix Universal Server (PRESS, 1997). Outro exemplo mais conhecido deste tipo de sistema é o PostgreSQL (STONEBRAKER *et al.*, 1986). Este sistema foi inicialmente desenvolvido na universidade de Berkeley, em 1986, e a princípio chamado de Postgres, como sucessor de outro sistema, o Ingres, com o objetivo de implementar o conceito de sistema objeto-relacional, então uma novidade. Oito anos depois do início de seu desenvolvimento, o sistema se tornou comercial, chamado de Illustra e incorporado ao sistema Informix. Uma nova linha de trabalho continuou o desenvolvimento do sistema, sob a bandeira do código aberto, passando a adquirir o nome atual de PostgreSQL. O sistema está até hoje em desenvolvimento (atualmente encontra-se na versão 8.2, lançada em dezembro de 2006) e é amplamente utilizada na comunidade científica.

Apesar da flexibilidade de tipos de dados fornecida por estes sistemas, todos continuam restritos ao modelo de banco de dados relacional. Uma abordagem mais radical para apoiar o desenvolvimento de sistemas de bancos de dados não-convencionais é a abordagem de *toolkits* (CAREY *et al.*, 1996). Toolkits não prevêem nenhum modelo de dados específico, mas implementam funcionalidades que todos os sistemas de bancos de dados utilizam, independente do modelo, como gerenciamento de transações, controle de

concorrência, recuperação e otimização de consultas. Dois principais projetos desta abordagem são GENESIS (BATORY *et al.*, 1988) e EXODUS (CAREY *et al.*, 1986).

O sistema SECONDO (GÜTING *et al.*, 2000) oferece a facilidade de extensão dos sistemas objeto-relacionais combinado com a flexibilidade dos toolkits. Assim, o SECONDO fornece um ambiente para o desenvolvimento de novas *álgebras*, compostas de novos tipos de dados (ou até mesmo novos modelos de dados) e novas operações. Este sistema foi desenvolvido como um ambiente para o desenvolvimento de protótipos de pesquisa e para o estudo do funcionamento de um sistema de gerencia de banco de dados.

3.1. Arquitetura

A arquitetura do sistema SECONDO é divida principalmente em três partes: kernel, otimizador e interface. O kernel possui a implementação específica para os modelos de dados, extensíveis por álgebras e provê um processamento de consultas sobre as álgebras implementadas. Ele foi implementado sobre BerkeleyDB (OLSON *et al.*, 1999) e escrito em C++. O otimizador provê a capacidade de otimização das consultas. Além disso, implementa a parte essencial de linguagem de consulta nos moldes de SQL, em uma notação adaptada para PROLOG. O otimizador foi escrito em PROLOG. A interface gráfica com o usuário (GUI – Graphic User Interface) pode ser estendida para quaisquer novos tipos de dados implementados. Atualmente existe uma interface implementada para os tipos de dados espaciais e objetos em movimento. A interface foi implementada em Java.

A integração entre as três partes do SECONDO está descrita na Figura 12. A interface pode se comunicar diretamente com o kernel, enviando consultas e exibindo as respostas. A interface pode também se comunicar com o otimizador, que por sua vez se comunica com o kernel quando necessário para obter informações sobre esquemas de relações, cardinalidade de relações e seletividade de predicados. Neste caso, o otimizador age como um servidor para a interface e como um cliente para o kernel.

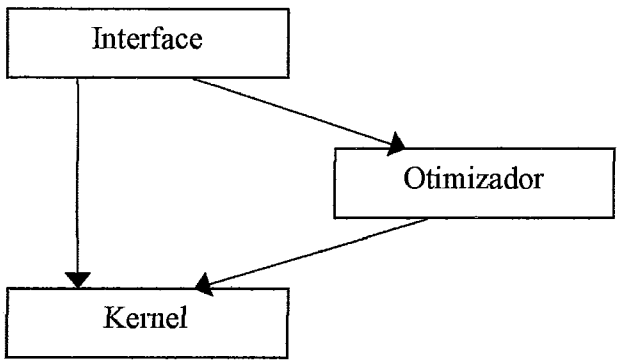


Figura 12. Integração dos componentes do sistema SECONDO

3.2. Assinatura de segunda ordem

Os tipos de dados e operações implementados no sistema SECONDO estão agrupados em álgebras. O conceito de álgebra é baseado no conceito de assinatura de segunda ordem (second-order signature) (GÜTING, 1993). A idéia é utilizar duas assinaturas conjuntas. Qualquer assinatura provê tipos e operações. A primeira assinatura provê tipos de dados. As operações nesta assinatura são construtores de tipos. Esta assinatura define como construtores de tipos podem ser aplicados para determinados tipos. A segunda assinatura define operações sobre tipos definidos na primeira assinatura. Um conjunto de assinaturas de segunda ordem é chamado então de uma *álgebra*.

Uma álgebra provê uma coleção de construtores de tipos, implementando uma estrutura de dados para cada um deles. Um pequeno grupo de funções de suporte é necessário para registrar o construtor de tipo dentro da álgebra. Similarmente, um módulo de álgebra provê operadores, implementando funções de apoio para eles, como mapeamento de tipos, avaliação, etc. (GÜTING *et al.*, 2005). Atualmente existem 43 álgebras implementadas no SECONDO, além das duas álgebras desenvolvidas neste trabalho (as álgebras Raster e RasterSpatial). Dentre estas álgebras, destacam-se as álgebras *Temporal*, *PlaneSweep*, *MP3*, *MIDI* e *Spatial*, destinada a trabalhar com vários tipos de dados e operações. A álgebra *Spatial* foi amplamente utilizada neste trabalho.

3.3. Representação de dados espaciais

O sistema SECONDO apresenta em sua álgebra espacial quatro tipos de dados espaciais:

- Ponto
- Pontos
- Linha
- Região

Como descrito em GUTING (1994), *ponto* representa um objeto para o qual apenas sua localização no espaço, e não o seu tamanho, é relevante (Figura 13a). Por exemplo, uma cidade pode ser modelada como um ponto em um modelo que descreve uma grande área geográfica (um mapa em escala pequena). *Pontos* é um objeto que representa um conjunto de objetos do tipo ponto (Figura 13b). É importante ressaltar que em um objeto deste tipo não existem segmentos ligando os pontos, apenas os pontos em si. *Linha* pode ser compreendida como uma curva no espaço, representada por uma polilinha, ou seja, uma seqüência de segmentos de linha (Figura 13c). *Linha* é a abstração básica para movimentações através do espaço ou conexões no espaço (rodovias, rios, cabos de telefone, eletricidade, etc.). *Região* é uma abstração para representar algo que possua uma extensão em um espaço bi-dimensional, por exemplo, um país, um lago ou uma zona florestal (Figura 13d). Uma *região* pode conter buracos e também pode consistir de várias partes disjuntas. O tipo de dado *região* foi utilizado na implementação da proposta como representação de *polígono*.

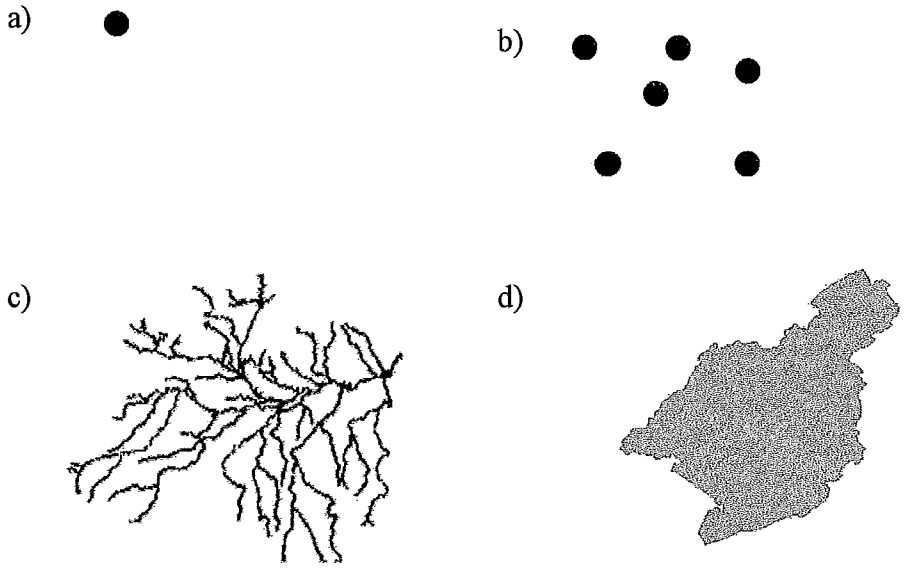
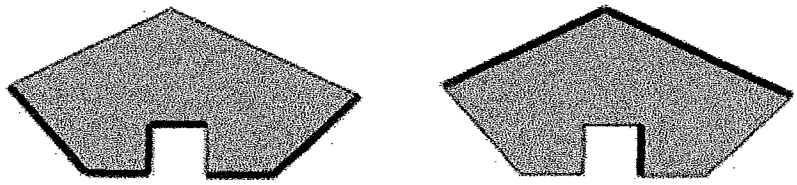


Figura 13. Exemplos de dados espaciais que podem ser armazenados no SECONDO. (a) Ponto; (b) Pontos; (c) Linha; e (d) Região

3.3.1. Propriedade *InsideAbove*

Uma das características armazenadas nos segmentos dos objetos espaciais (linha e região) que é importante para a nossa proposta é o indicador chamado de *InsideAbove*. Esta propriedade é utilizada na geração de assinaturas 3CRS, conforme descrito no capítulo 2.3. Esta propriedade assume o valor *true* quando a área interna do polígono está acima do segmento e *false* quando a área encontra-se abaixo do segmento. Quando o segmento é vertical, a propriedade assume o valor *true* quando a área interna encontra-se à esquerda do mesmo. A Figura 14 apresenta exemplos de segmentos, indicando o valor da propriedade InsideAbove para cada um deles. Na Figura 14a, os segmentos em negrito apresentam a propriedade InsideAbove com valor true. Na Figura 14b, os segmentos em negrito apresentam a propriedade com valor *false*.



a) InsideAbove = true b) InsideAbove = false

Figura 14. Exemplos de valores para a propriedade InsideAbove

3.4. Armazenamento de dados

Durante o processamento de consultas, expressões e valores são passados entre os diferentes módulos e funções. Para tal, é utilizado o conceito de *nested lists* (listas aninhadas), utilizado em linguagens funcionais (HUDAK, 1989) como um método genérico para representar expressões, consultas e valores constantes em consultas, resultado de consultas e arquivos externos. Uma *nested list* é ou um valor de um *tipo atômico* (*inteiro, real, booleano, string, texto ou símbolo*), ou uma lista de tamanho arbitrário. Cada elemento na lista por sua vez é ou um valor atômico ou uma *nested list*. A representação textual de uma *nested list* consiste de um parêntese esquerdo, seguido por um número arbitrário de elementos, separados por um espaço em branco, seguido de um parêntese direito. Por exemplo, a expressão (1 2.01 ((int) ("XX" TRUE))) representa uma *nested list* de três elementos: o átomo inteiro 1 é o primeiro elemento, o átomo real 2.01 é o segundo e a lista ((int) ("XX" TRUE)) é o terceiro (GÜTING *et al.*, 2000).

Além da representação dos dados por *nested list*, outra forma que pode ser utilizada para o armazenamento de dados é o *FLOB* (*Faked Large OBject*). Este é um grande objeto que pode ser usado como parte de uma representação de um tipo de dados com a particularidade de alterar automaticamente sua representação de acordo com seu tamanho. Um FLOB pode ser armazenado juntamente com o restante da representação do tipo de dados ou ser armazenado como uma entidade independente. Para alguns tipos de dados, o tamanho da parte variável pode oscilar entre muito pequeno e muito grande. Como um exemplo, pode ser considerado um polígono, que pode conter três vértices ou trezentos mil vértices. Estes vértices então podem ser armazenados em um FLOB. Considerando que sistemas de armazenamento normalmente trabalham com base em páginas (o menor bloco

de espaço de armazenamento transferido), quando o tamanho do valor é pequeno, pode ser mais eficiente armazenar a parte variável juntamente com o resto da representação do valor, para evitar um acesso ao disco caso a parte variável seja utilizada. Em contrapartida, se esta parte é muito grande, pode ser mais eficiente armazená-la como um objeto independente, que é acessado (e, portanto, carregado para a memória) apenas quando necessário. (RODRÍGUEZ-LUACES, 2000)

4. Assinatura Raster de 3 Cores

A Assinatura Raster de 3 Cores (*Three-Color Raster* Signature – 3CRS), que publicamos na conferência ACM-GIS (AZEVEDO *et. al*, 2006), é uma forma simplificada de representar objetos espaciais através de um *grid* de células que usa poucas cores. A assinatura 3CRS é baseada na assinatura 4CRS (Seção 2.3), e pode ser utilizada para diminuir os custos de processamento e acesso a disco em operações como a junção espacial, como demonstram os experimentos no capítulo 5. Esta representação consiste em um grupo de células, onde cada uma pode assumir um dentre três valores possíveis, os quais representam tipos de interseção entre o objeto sendo aproximado e a célula. A Tabela 5 apresenta os valores possíveis para células de aproximações 3CRS.

Tabela 5. Valores possíveis para as células na aproximação 3CRS

Valor	Significado
Vazio	A célula não possui interseção com o objeto.
Inconclusivo	Existe uma parte do objeto no interior da célula, e ela não está totalmente preenchida pelo mesmo.
Cheio	A célula está totalmente ocupada pelo objeto. Este tipo de célula ocorre apenas na representação de polígonos.

A assinatura 3CRS se assemelha à assinatura 4CRS (apresentada no capítulo 2.3), porém, na primeira, a célula *Inconclusiva* representa tanto as células *Pouco* quanto *Muito* da segunda, sem precisar calcular a porcentagem da célula contida no objeto. Esta diferença faz com que a assinatura 3CRS seja gerada mais rapidamente do que a assinatura 4CRS, apesar representar de forma menos precisa o objeto original, pois as células *Pouco* e *Muito* são reduzidas no tipo de célula *Inconclusivo*. Outra vantagem da assinatura 3CRS sobre a 4CRS é que a 4CRS é restrita a polígonos, enquanto a 3CRS pode ser utilizada para representar polígonos, polilinhas e pontos. A única diferença da representação de polígonos para a representação de polilinhas e pontos é que neste caso a célula do tipo *Cheio* não é utilizada, já que seriam necessários infinitos pontos ou infinitos segmentos de linha para preencher completamente uma célula. O algoritmo de interseção de assinaturas, entretanto é exatamente o mesmo para qualquer combinação de tipos de dados (polígono x polígono,

polígono x polilinha, polígono x pontos, polilinha x polilinha, polilinha x pontos ou pontos x pontos). A Tabela 6 apresenta um comparativo entre as assinaturas 4CRS, 5CDRS e 3CRS em relação aos tipos objetos que podem ser representados: polígonos, polilinhas e pontos. Apenas a assinatura 3CRS permite representar os três tipos de objeto.

Tabela 6. Comparação entre objetos representados pelas assinaturas 4CRS, 5CDRS e 3CRS

Característica	4CRS	5CDRS	3CRS
Representa polígonos	Sim	Não	Sim
Representa polilinhas	Não	Sim	Sim
Representa pontos	Não	Não	Sim

Um exemplo de um polígono aproximado com a assinatura 3CRS é ilustrado na Figura 15. A Figura 16 apresenta uma polilinha utilizando a mesma representação. O algoritmo utilizado na geração da assinatura é apresentado na seção 4.1; o algoritmo para o uso da assinatura na junção de objetos espaciais é apresentado na seção 4.2.

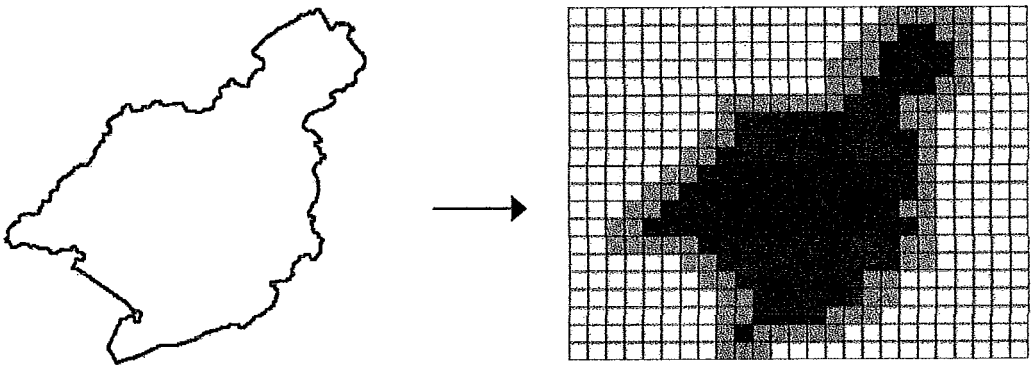


Figura 15. Exemplo de polígono representado com a assinatura 3CRS

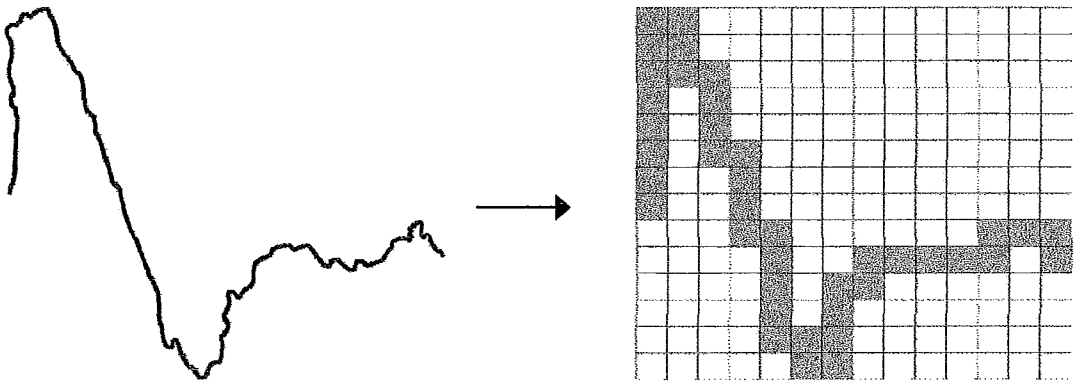


Figura 16. Exemplo de polilinha representada com a assinatura 3CRS

4.1. Geração da assinatura

A geração da Assinatura Raster de 3 Cores é dividida em três etapas:

- Etapa 1: Consiste em atribuir uma grade regular para representar o objeto. Nesta primeira etapa, todas as células são inicializadas como com o tipo *Vazio*.
- Etapa 2: A segunda etapa tem como objetivo marcar as células *Pouco* da assinatura. No caso de pontos, estas serão as células que onde existem pontos. No caso de polilinha, as células *Inconclusivo* são as células interceptadas por polilinhas, Finalmente, no caso de polígono, as células *Pouco* correspondem às células interceptadas pela borda do polígono. Nos casos de pontos e polilinhas, o algoritmo de geração da assinatura termina nesta etapa.
- Etapa 3: Esta etapa existe apenas para polígono, e é responsável por marcar as células *Cheias*, ou seja, as células completamente cobertas pelo polígono.

Cada etapa será descrita com mais detalhes nas seções a seguir.

4.1.1. Divisão do espaço em células

O primeiro passo para construir a assinatura de um objeto é atribuir uma grade regular que divide o espaço em células. Estas serão inicialmente preenchidas com o valor *vazio*. Nos próximos passos, os valores de algumas células serão alterados para

inconclusivo e, se o objeto tratado for um polígono, algumas poderão ser alteradas para *cheio*. O objetivo deste primeiro passo é definir quantidade e a posição das células utilizadas na assinatura. A divisão do espaço em células para a assinatura raster de três cores utiliza o mesmo algoritmo de divisão do espaço das assinaturas 4CRS (seção 2.3) e 5CDRS (seção 2.4) (ZIMBRAO *et al.*, 1998).

A quantidade de células utilizadas na assinatura deve obedecer a um limite máximo *k*. A quantidade de células é definida de acordo com o tamanho das mesmas. É importante ressaltar que dentro de uma assinatura, todas as células possuem o mesmo tamanho. Quanto menor o tamanho das células, maior a quantidade utilizada na assinatura.

Quando duas assinaturas são comparadas, é necessário que ambas possuam a mesma resolução (mesmo tamanho de células). Caso as assinaturas não atendam a esta exigência, a assinatura com maior resolução terá a sua resolução diminuída até que se equipare com a outra assinatura. Para facilitar esta mudança de resolução, o tamanho das células deve ser uma potência de 2 (2^n, *n* inteiro). Assim, para diminuir a resolução de uma assinatura, basta agrupar células adjacentes. O tamanho da nova célula resultante deste agrupamento também será uma potência de 2. A mudança de resolução será detalhada na seção 4.2.1.

A definição da posição deve, ao comparar aproximações de objetos distintos, garantir que suas relações de posicionamento sejam mantidas. Este objetivo é alcançado dividindo-se o espaço globalmente enquadrando o objeto dentro das células, e não empregando uma divisão de acordo com cada um. Ao comparar duas assinaturas, deve-se ainda garantir que duas células que se intersectam possuam a mesma coordenada. Para isso, as coordenadas das células são múltiplos de potências de 2 ($2^k a$, onde *k* e *a* são inteiros).

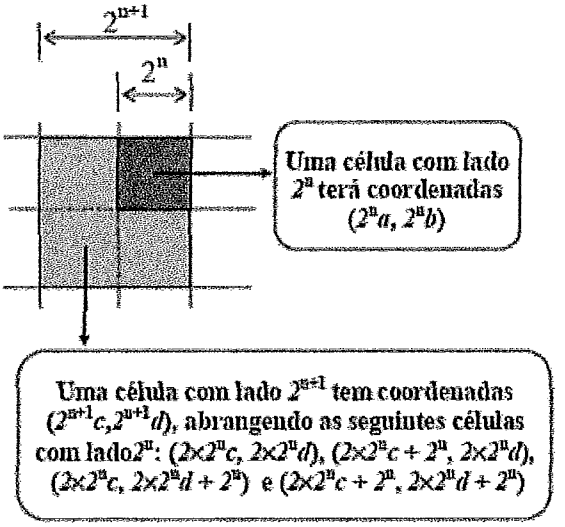


Figura 17. Alinhamento dos cantos das células (ZIMBRAO *et al.*, 1998)

O MBR da assinatura não necessariamente é o mesmo MBR do objeto. Para a assinatura, é criado um novo MBR, chamado MBR-2^k. O MBR-2^k do objeto corresponde a um MBR cujos vértices são da forma $(2^k a_0, 2^k b_0)$ e $(2^k a_m, 2^k b_n)$, onde a_0, am, 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 x 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 $2^k a_0$, $2^k a_1$,..., $2^k a_m$ definem um conjunto de linhas paralelas ao eixo vertical que chamaremos de X_0, X_1,...,X_m. De modo semelhante, definimos as linhas horizontais Y_0, Y_1,...,Y_n. (AZEVEDO, 2001)

A Figura 18 apresenta um exemplo de um primeiro passo da geração da assinatura. À esquerda, o objeto original. À direita, o objeto, sobreposto à grade de células *vazias* geradas no primeiro passo.

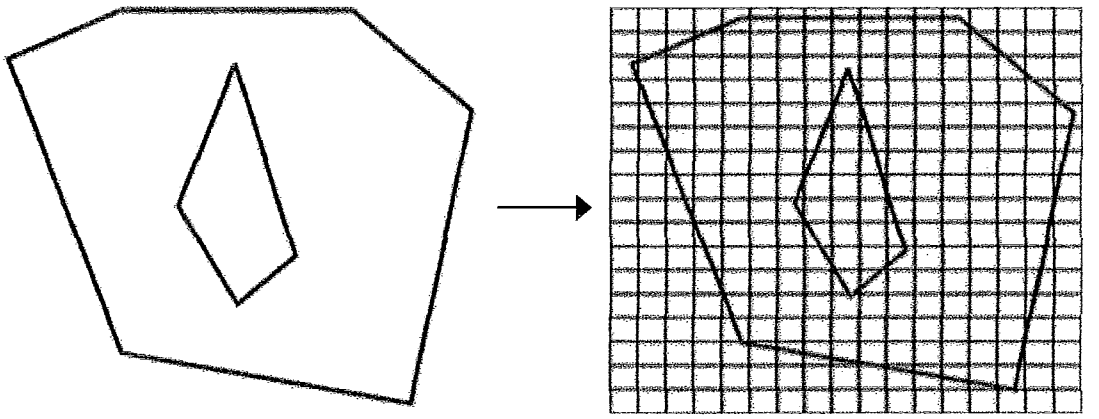


Figura 18. Resultado do passo 1, dividindo o espaço em células

4.1.2. Células *inconclusivas*

Com o espaço dividido em células de tamanho e posições definidos, e todas marcadas como *vazias*, passamos para o segundo passo, que tem como objetivo marcar as células *Inconclusivas*. De acordo com o que foi definido na Tabela 5, as células do tipo *Pouco* possuem pelo menos um ponto do objeto em seu interior, embora a célula não esteja totalmente contida no interior do mesmo. Assim, podemos afirmar que a borda do objeto intersecta esta célula, seja uma borda externa ou a borda de um eventual "buraco" no mesmo. A estratégia desta etapa então é percorrer os segmentos do objeto, marcando as células intersectadas pelos mesmos como *Inconclusivas*.

O algoritmo se inicia selecionando um segmento do objeto. No caso de o objeto ser um ponto ou um conjunto de pontos, é selecionado um dos pontos. O algoritmo pode ser iniciado a partir de qualquer segmento, sem necessidade de se escolher algum em especial. A célula que contém o ponto mais à esquerda do segmento é marcada como *Inconclusivas*. O algoritmo segue todas as células intersectadas pelo segmento, marcando cada uma delas também como *Inconclusivas*. Para descobrir tais células, o algoritmo descobre qual borda da célula atual é cortada pelo segmento. Para tal, é utilizado o algoritmo Sutherland-Cohen (NEWMAN *et al.*, 1979). Quando a borda intersectada é à direita, significa que a célula à direita também tem interseção com o segmento. A célula atual passa a ser a célula à direita, que também é marcada como *Inconclusiva*. Analogamente, a célula atual passa a ser a célula abaixo quando a borda intersectada é a inferior ou passa a ser a célula acima quando

a borda intersectada é a superior. Como o segmento é percorrido a partir do ponto à esquerda e segue para o ponto à direita, uma interseção com a borda esquerda da célula não desloca a célula atual para esta direção, já que esta célula já foi analisada anteriormente. O algoritmo segue até o segmento ser inteiramente analisado, ou seja, até a célula atual ser a célula que contém o ponto mais à direita do segmento. A exceção é quando o segmento é vertical. Neste caso, nenhum dos dois pontos fica situado mais à esquerda. Neste caso, o ponto mais acima é utilizado e a célula atual só é alterada para a célula inferior, no caso de o segmento intersectar a borda inferior da célula. No caso de pontos, a primeira célula é também a célula final. Neste caso, apenas uma célula é marcada como *Inconclusiva*, por ponto. O algoritmo inicial para este passo está descrito em pseudo-linguagem na Figura 19. A Figura 20 apresenta um exemplo do resultado deste passo, com as células *Inconclusivas* marcadas em cinza. O desenho do objeto original foi sobreposto às células, para facilitar a visualização da assinatura.

```
para cada segmento
    célulaAtual = célula que contém o ponto mais à esquerda do
segmento;
    célulaFinal = célula que contém o ponto mais à direita do
segmento;
    se célulaAtual = vazio
        marcarInconclusivo(célulaAtual);
    enquanto célulaAtual != célulaFinal
        realiza interseção do segmento com célula atual;
        se segmento intersecta borda esquerda da célula
            célulaAtual.x = célulaAtual.x + 1;
        se segmento intersecta borda superior da célula
            célulaAtual.y = célulaAtual.y + 1;
        se segmento intersecta borda inferior da célula
            célulaAtual.y = célulaAtual.y - 1;
        se célulaAtual = vazio
            marcarInconclusivo(célulaAtual);
```

Figura 19. Algoritmo inicial de marcação de células Inconclusivas

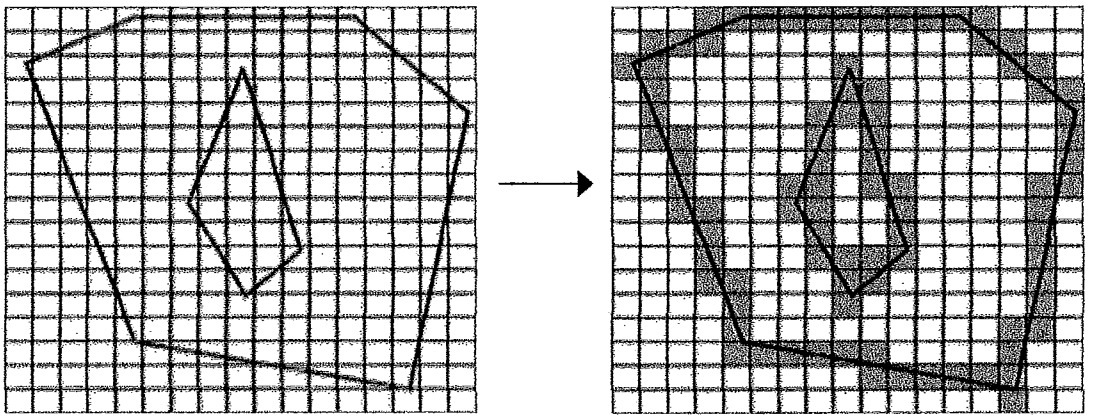


Figura 20. Resultado do passo 2, marcando células Inconclusivas

O algoritmo descrito até aqui, entretanto, ainda não está completo. Outro objetivo desta etapa da geração da assinatura é construir uma matriz que será utilizada para a próxima etapa, no preenchimento das células *cheio*. Como células *cheio* são utilizadas apenas para polígonos, a construção da matriz é necessária apenas para este tipo de objeto. Esta matriz é uma matriz de números inteiros, onde cada número corresponde a uma célula na assinatura. Todos os números são inicializados com o valor zero. Sempre que um segmento intersecta uma borda lateral de uma célula, o número correspondente na matriz é alterado, de acordo com o valor da propriedade InsideAbove do segmento. Esta propriedade foi descrita na seção 3 e indica se o interior do polígono está situado acima do segmento. Quando o valor da propriedade InsideAbove do segmento for *true*, então o número correspondente à célula é somado em 1. Quando o valor for *false*, então o número é subtraído em 1. Estes valores serão utilizados para identificar as células que estão contidas no interior do polígono, no próximo passo. As interseções com as bordas superiores e inferiores não alteram a matriz de troca. É importante ressaltar que, quando a borda esquerda e a borda direita são intersectadas, o valor da matriz é alterado duas vezes. A Figura 21 apresenta uma matriz de exemplo, gerada no mesmo exemplo da Figura 20. Aqui também o objeto original foi representado sobre a matriz para facilitar a compreensão da geração dos valores. O algoritmo completo utilizado neste passo está descrito em pseudo-linguagem na Figura 22.

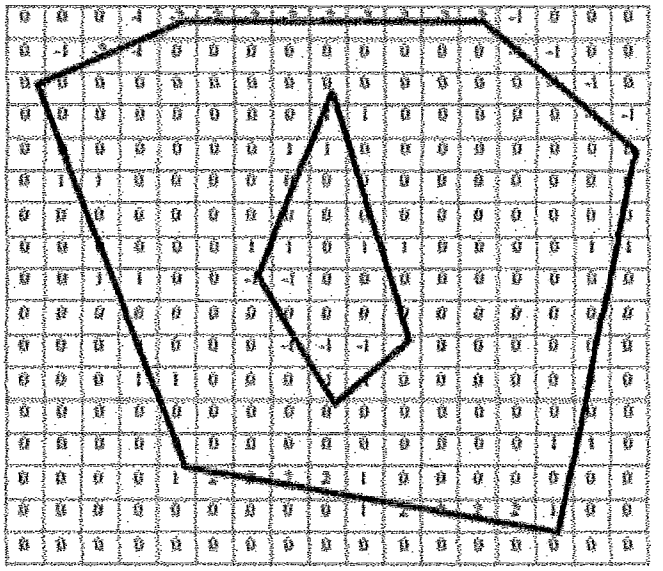


Figura 21. Exemplo de matriz de troca gerada no segundo passo

É importante enfatizar alguns aspectos deste passo:

- Não há uma ordem específica para percorrer os segmentos, já que por este algoritmo um segmento não tem influência sobre outro;
- É possível recuperar uma célula indexada na assinatura em um tempo constante. Assim, é possível identificar a célula que contém um ponto em tempo constante;
- Dada a equação da linha que compõe o segmento, é possível seguir o segmento e marcar as células intersectadas em um tempo constante por célula. Assim, para um segmento, o tempo necessário será linear de acordo com o número de células marcadas;
- Normalmente o tamanho do segmento é pequeno em relação ao tamanho da célula e intersecta apenas um número pequeno delas, assim sendo necessário tempo O(1) por segmento;
- Finalmente, este passo computa todas as células *Inconclusivas* e utiliza um tempo O(n) se todos os segmentos forem suficientemente curtos em relação ao tamanho da célula.

```
para cada segmento
      célulaAtual = célula que contém o ponto mais à esquerda do
segmento;
      célulaFinal = célula que contém o ponto mais à direita do
segmento;
      se célulaAtual = vazio
            marcarInconclusivo(célulaAtual);
      enquanto célulaAtual != célulaFinal
            realiza interseção do segmento com célula atual;
            se segmento intersecta borda esquerda da célula
                  se insideAbove = true
                        matrizTroca[célulaAtual] =
                                    matrizTroca[célulaAtual] + 1;
                  senão
                        matrizTroca[célulaAtual] =
                                    matrizTroca[célulaAtual] - 1;
            se segmento intersecta borda direta da célula
                  se insideAbove = true
                        matrizTroca[célulaAtual] =
                                    matrizTroca[célulaAtual] + 1;
                  senão
                        matrizTroca[célulaAtual] =
                                    matrizTroca[célulaAtual] - 1;
                  célulaAtual.x = célulaAtual.x + 1;
            se segmento intersecta borda superior da célula
                  célulaAtual.y = célulaAtual.y + 1;
            se segmento intersecta borda inferior da célula
                  célulaAtual.y = célulaAtual.y - 1;
            se célulaAtual = vazio
                  marcarInconclusivo(célulaAtual);
```

Figura 22. Algoritmo de preenchimento de células *Inconclusivo*

4.1.3. Células *cheio*

No final do segundo passo, o espaço já está dividido em células e as células do tipo *inconclusivo* estão marcadas. Neste ponto, todas as células que não são *Inconclusivas* estão marcadas como *vazias*. Quando o objeto representado é do tipo ponto ou polilinha, a geração da assinatura está concluída. Entretanto, quando o objeto representado é um polígono, ainda é necessário marcar as células no interior do mesmo como *Cheias*. Este é o objetivo deste terceiro passo. O algoritmo utiliza a matriz de troca calculada no passo anterior, conforme exemplificado na Figura 21. O algoritmo percorre todas as células, de baixo para cima, uma coluna de cada vez. Uma variável, *contador*, é utilizada para somar os valores da matriz, nas posições correspondentes às células percorridas. Esta variável é zerada a cada nova coluna. Em cada célula percorrida não marcada como *Inconclusiva*, o valor de *contador* é analisado. Se o valor de *contador* for igual a 2, então a célula é marcada como *Cheia*; caso contrário ela permanece inalterada (como no início deste passo as células estavam marcadas como *vazias* ou *Inconclusivas* e a célula não está marcada como *Inconclusiva*, então a célula permanece como *vazia*). O algoritmo termina quando percorre todas as células da assinatura, marcando as necessárias como *Cheias*. O pseudo-código deste passo está representado na Figura 23. Como o número máximo de células na assinatura é um valor constante k, então este passo requer um tempo $O(k)$ para ser executado.

```
para cada coluna de células na assinatura
      contador = 0;
      para cada linha na coluna (de baixo para cima)
            contador = contador + matrizTroca[coluna, linha];
            celulaAtual = celula(coluna, linha);
            se contador = 2 e valor(celulaAtual) != inconclusivo
                  marcarCheio(celulaAtual);
```

Figura 23. Algoritmo de preenchimento de células cheio

A Figura 24 apresenta um exemplo de parte da execução deste passo do algoritmo. Neste exemplo, o conjunto de células resultado do exemplo da Figura 20 é examinado para a marcação das células *cheio*. Para este exemplo, examinaremos apenas a coluna indicada (a décima coluna). À esquerda das células, estão indicados os valores correspondentes de *contador*, conforme o algoritmo examina cada linha desta coluna. A variável *contador* é inicializada com o valor zero. Na primeira linha, o valor de *contador* continua zero, então a célula continua marcada como *vazia*. Na segunda linha, o valor correspondente da matriz de troca tem valor 1 (como calculado no passo anterior do algoritmo, já que um segmento do objeto intersecta o lado direito da célula e possui o atributo *InsideAbove* como *true*), então o valor de *contador* é incrementado em 1. A célula já estava marcada como *Inconclusiva*, então seu valor não é alterado. O mesmo ocorre na terceira linha. Na quarta linha o valor de *contador* é 2 e seu tipo não é inconclusivo. Assim, a célula é marcada como *Cheia*. O algoritmo segue até o final da coluna (e das linhas) até que todas as células *Cheias* estejam marcadas. O resultado final do terceiro passo para este exemplo está representado na Figura 25.

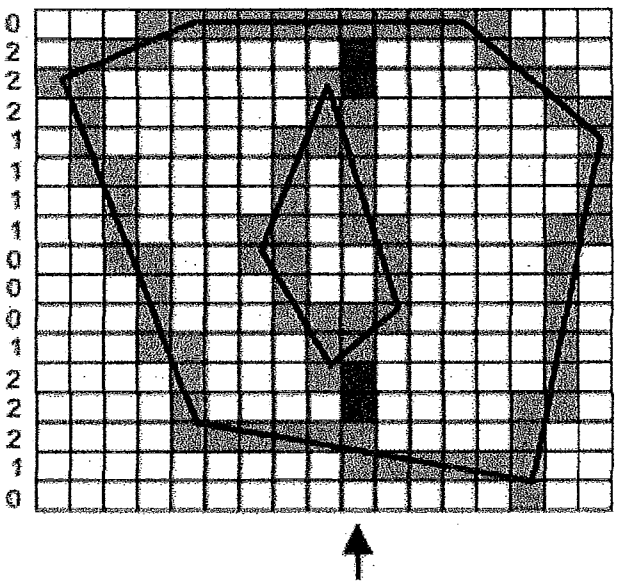


Figura 24. Exemplo de execução do algoritmo de marcação de células cheio

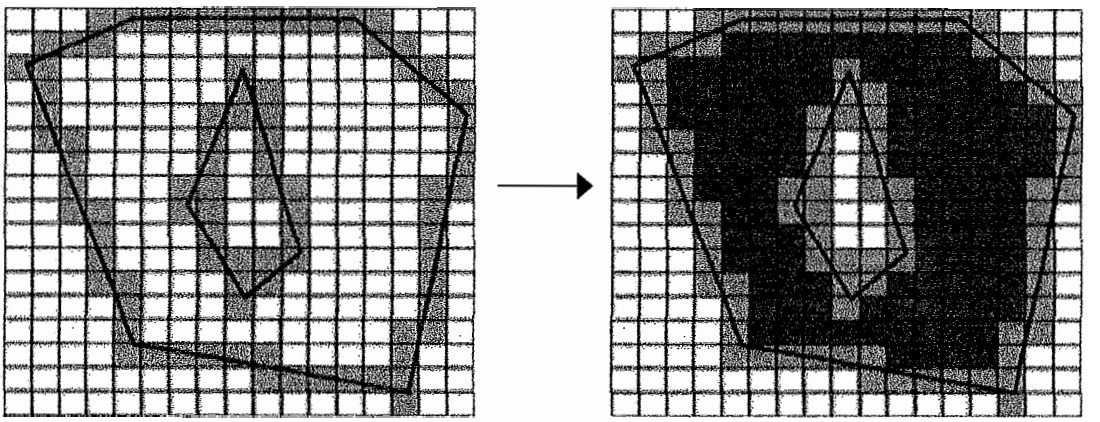


Figura 25. Resultado do passo 3, marcando células *Cheias*

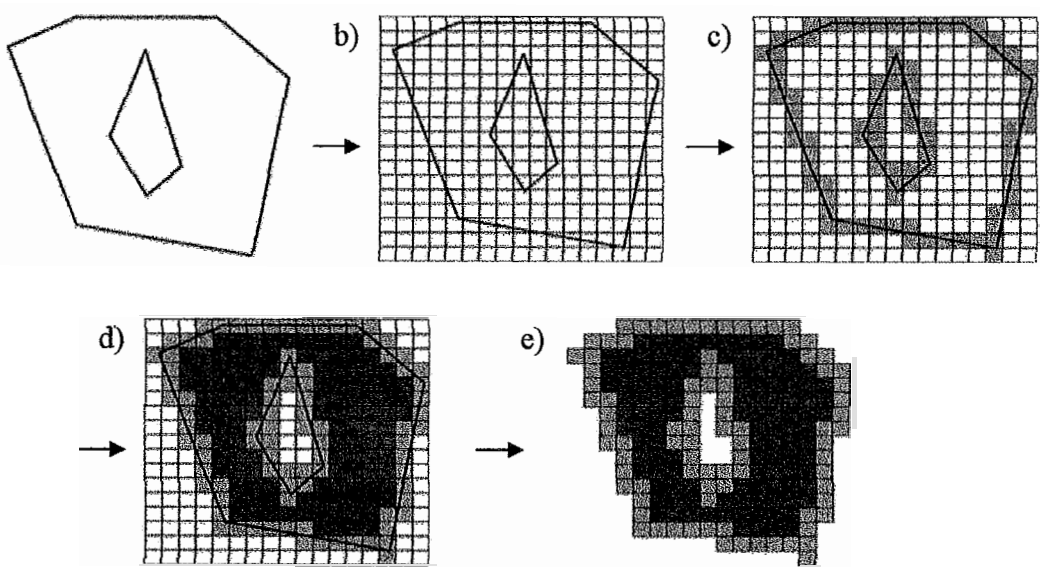


Figura 26. Todos os passos da geração da assinatura 3CRS. a) objeto original; b) objeto sobreposto à grade (etapa 1); c) células do tipo *Inconclusivo* (etapa 2); d) células *Cheio* (etapa 3); e) representação do objeto sem células *Vazio*.

4.2. Interseção de assinaturas

4.2.1. Mudança de resolução

A partir de duas assinaturas 3CRS geradas, é possível realizar um teste de interseção para ser utilizado como o segundo passo na arquitetura de processamento de consultas em múltiplos passos. Entretanto, antes de realizar tal teste de interseção, é necessário que as

duas assinaturas apresentem o mesmo tamanho de célula, ou seja, a mesma resolução. Quando isto não ocorre, é necessário realizar uma mudança de resolução em uma das assinaturas. Surgem então duas alternativas: aumentar a resolução da assinatura com menor resolução ou diminuir a resolução da assinatura com maior resolução.

Para aumentar a resolução de uma assinatura, ou seja, diminuir o tamanho de suas células, cada célula deve ser dividida em um grupo de células de tamanho menor. Esta abordagem não é muito eficiente, já que para marcar corretamente as novas células criadas como *Cheia*, *Inconclusiva* ou *Vazia*, seria necessário analisar o objeto original. A Figura 27 mostra um exemplo desta alternativa. A célula original é *Inconclusiva*. Dentre as quatro células derivadas da mesma, três são *vazias* e uma *Inconclusiva*.

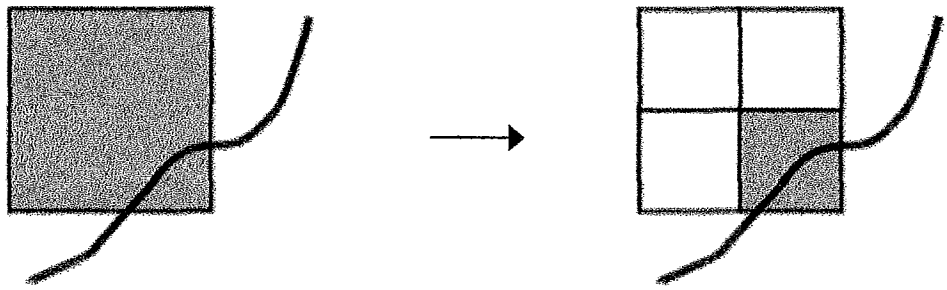


Figura 27. Abordagem ineficiente para igualar resoluções, dividindo células.

A segunda alternativa, mais eficiente, é a que foi adotada neste trabalho: diminuir a resolução (aumentar o tamanho das células) da assinatura com maior resolução. Para isso, as células são agrupadas, formando novas células. O tipo desta nova célula é determinado pelas células que foram agrupadas. Se todas as células originais eram do tipo *Cheia*, então a nova célula também será do tipo *Cheia*. Analogamente, se todas as células originais eram do tipo *vazia*, então a nova célula também será *vazia*. Qualquer outra combinação de células (todas *Inconclusivas*, ou um conjunto heterogêneo de células) resultará em uma célula *Inconclusiva*. A Figura 28 mostra um exemplo deste tipo de mudança de resolução. Como uma das células originais era *Inconclusiva*, então a nova célula também é do tipo *Inconclusiva*.

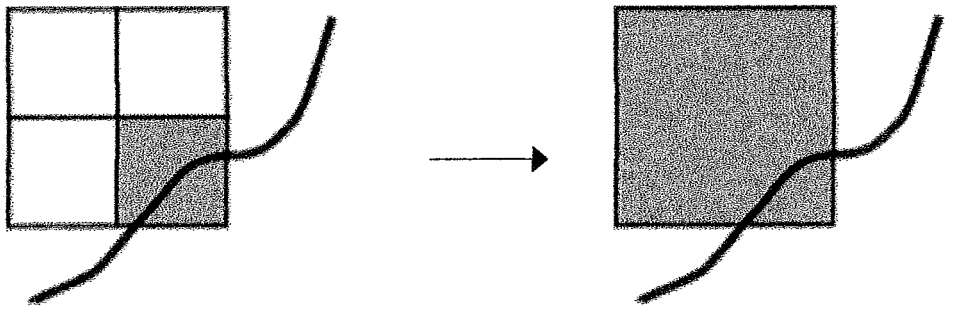


Figura 28. Abordagem utilizada na mudança de resolução, agrupando células

4.2.2. Algoritmo de interseção

Quando as duas assinaturas a serem comparadas apresentam a mesma resolução, é possível realizar o teste de interseção entre as mesmas. Para tal, as assinaturas são sobrepostas e as células correspondentes são comparadas. Apenas as células contidas na interseção dos MBR-2^k (seção 4.1.1) são comparadas, já que as células fora desta área não têm interseção. A avaliação de cada par de células correspondentes apresenta um resultado parcial. Os possíveis resultados parciais obtidos através das diferentes combinações de células estão discriminados na Tabela 7 e exemplificados na Figura 29.

Tabela 7. Combinações possíveis de pares de células na assinatura 3CRS

	Cheio	Inconclusivo	Vazio
Cheio	*Hit*	*Hit*	Não há interseção
Inconclusivo	*Hit*	Inconclusivo	Não há interseção
Vazio	Não há interseção	Não há interseção	Não há interseção

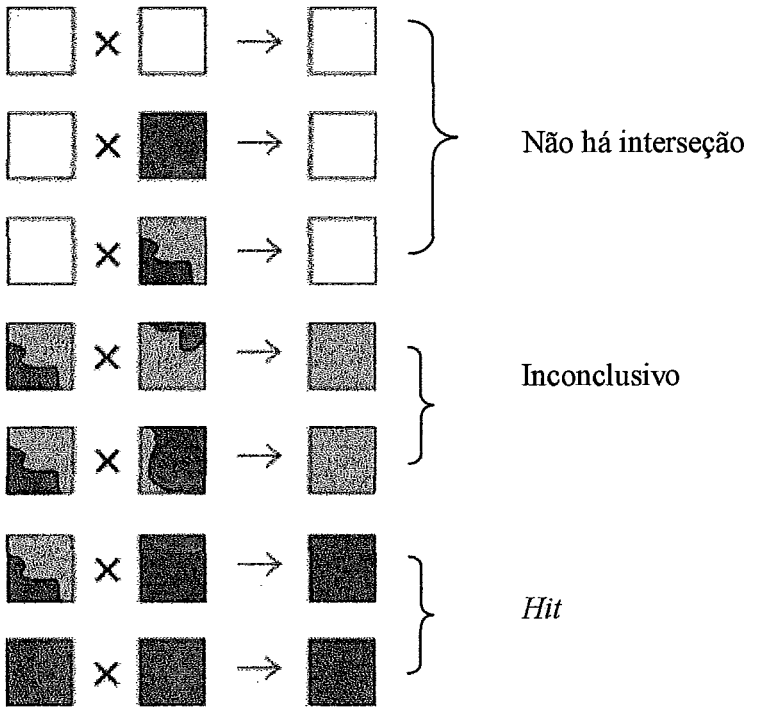


Figura 29. Abordagem utilizada na mudança de resolução, agrupando células

Quando uma das células é do tipo *vazia*, independente do tipo da outra célula comparada, o resultado parcial é *Não há interseção*. Quando uma das células é do tipo *Cheia* e a outra é do tipo *Inconclusiva* ou *Cheia*, então certamente no espaço definido por estas células existe interseção, então o resultado parcial é *hit*. Quando ambas as células são do tipo *inconclusivo*, então não é possível determinar se neste espaço existe interseção sem analisar o objeto original. Neste caso então, o resultado parcial é *inconclusivo*.

Para obter o resultado final do teste de interseção, cada par de células correspondente dentro do espaço com interseção dos MBR-2^k é analisado. Se pelo menos um resultado parcial for *hit*, então dentro desta célula ocorreu uma interseção. Logo, os objetos possuem interseção e o resultado final é *possui interseção*. Se, por outro lado, todos os resultados parciais forem *Não há interseção*, então em nenhuma célula ocorreu uma interseção. Como todas as partes dos objetos original estão contidas em alguma célula, neste caso podemos afirmar que não há interseção entre os objetos originais. Se nenhum destes dois casos ocorrer, ou seja, nenhum par obteve como resultado parcial *hit* e pelo menos um par obteve resultado parcial *inconclusivo*, isto significa que o teste baseado nas assinaturas não foi capaz de identificar se existe ou não interseção entre os objetos. Neste

caso, a resposta final do teste é *inconclusivo* e o teste de geometria exata é chamado para se obter uma resposta. O algoritmo supra-citado está descrito em pseudo-linguagem na Figura 30.

```
algoritmo possuiInterseção(3CRS1, 3CRS2)
   interMBR = MBRinterseção(3CRS1, 3CRS2);
   se 3CRS1.tamanhoDaCélula < 3CRS2.tamanhoDaCélula
      p3CRS = alteraResolução(3CRS1, 3CRS2.tamanhoDaCélula);
      g3CRS = 3CRS2;
   senão
      se 3CRS1.tamanhoDaCélula > 3CRS2.tamanhoDaCélula
            g3CRS = 3CRS1;
            p3CRS = alteraResolução(3CRS2, 3CRS1.tamanhoDaCélula);
      senão
            p3CRS = 3CRS1;
            g3CRS = 3CRS2;
   resultado = SEM_INTERSEÇÃO;
   para cada célula g de g3CRS contida em interMBR
      para cada célula p de p3CRS que intersecta g
            se g.tipo = VAZIO ou p.tipo = VAZIO
                  continua;
            se g.tipo = INCONCLUSIVO ou p.tipo = INCONCLUSIVO
                  resultado = INCONCLUSIVO
            se ( (g.tipo = CHEIO) e
                 (p.tipo = CHEIO ou
                  p.tipo = INCONCLUSIVO) ) ou
            se ( (p.tipo = CHEIO) e
                 (g.tipo = CHEIO ou
                  g.tipo = INCONCLUSIVO) )
                  retornar EXISTE_INTERSEÇÃO;
   retornar resultado;
```

Figura 30. Algoritmo de teste de interseção de duas assinaturas 3CRS

A Figura 31 e a Figura 32 apresentam dois exemplos de teste de interseção utilizando a assinatura 3CRS. A primeira apresenta um exemplo onde o resultado do teste é *falso*, ou seja, não há interseção entre os objetos. Na segunda, o exemplo apresenta como resultado *verdadeiro*, ou seja, há interseção entre os objetos.

A Figura 31(a) mostra os dois polígonos sendo testados neste exemplo. Nota-se que não existe interseção entre os objetos. A Figura 31(b) mostra as assinaturas 3CRS dos dois objetos, sobrepostas aos mesmos para facilitar a visualização. Como descrito no algoritmo de interseção, apenas as células dentro do MBR-2^k são analisadas. O MBR-2^k está demarcado em negrito nas duas assinaturas. Os pares de células correspondentes são analisados. Apenas dois casos ocorrem neste exemplo: célula *vazia* x célula *vazia*; e célula *vazia* x célula *Inconclusiva*. Em ambos os casos o resultado parcial é *Não há interseção*. Como todos os resultados parciais encontrados foram *Não há interseção*, então podemos garantir que o resultado final do teste é *Não há interseção*.

No segundo exemplo, os objetos a serem testados são exibidos na Figura 32(a). Neste caso, existe interseção entre eles. A Figura 31(b), analogamente ao exemplo anterior, mostra as assinaturas 3CRS dos dois objetos, sobrepostas aos mesmos para facilitar a visualização. Neste caso, ao analisar o par de células no canto esquerdo superior do MBR-2^k, representados pelas células *Cheia* no primeiro polígono e *inconclusivo* no segundo, o algoritmo garante que existe interseção entre os objetos neste espaço.

É importante ressaltar que, em ambos os exemplos, o resultado do teste foi obtido sem analisar a representação exata dos objetos, o que diminui o número de acessos ao disco no teste.

a)

b)

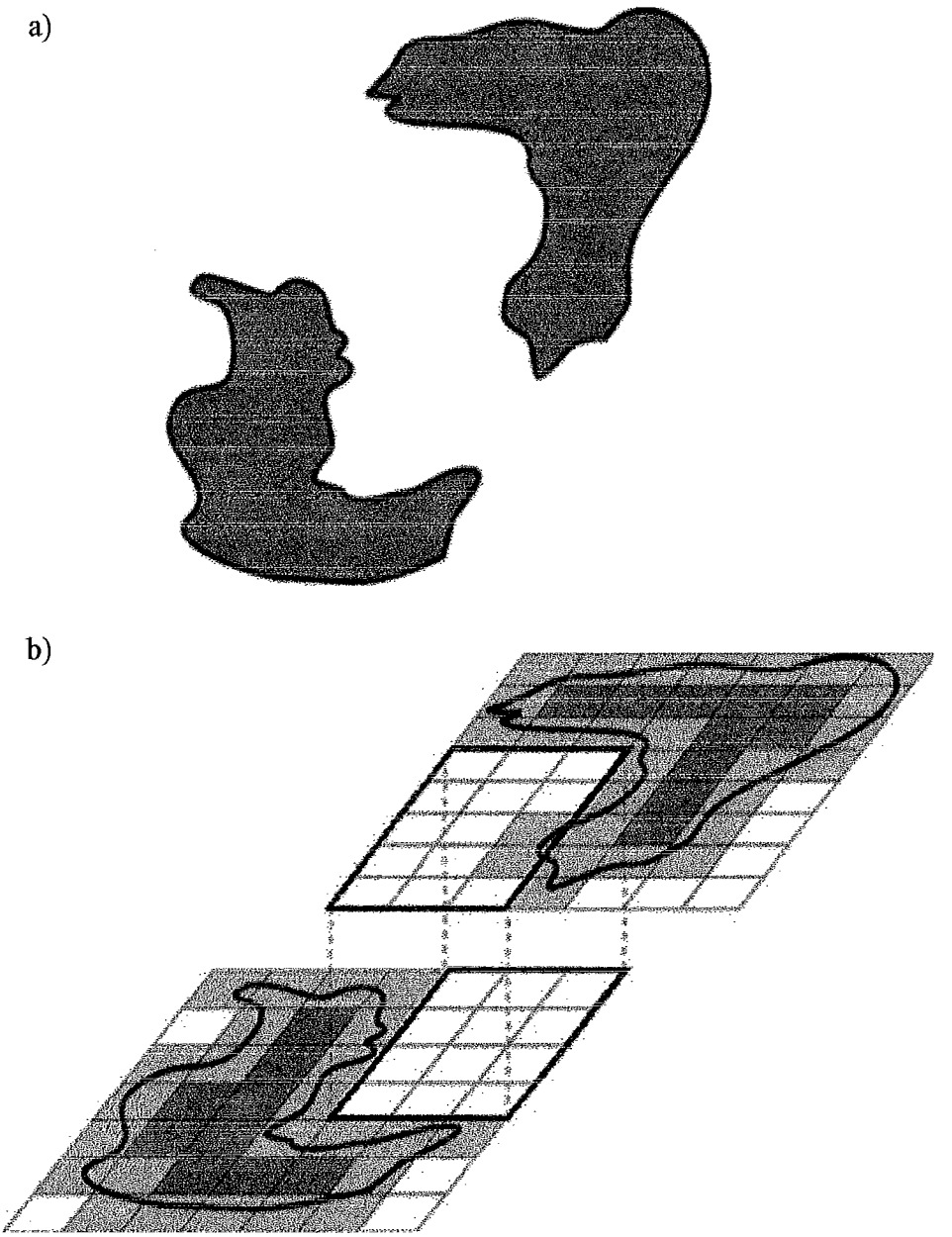


Figura 31. Exemplo de teste de interseção de assinaturas, sem interseção

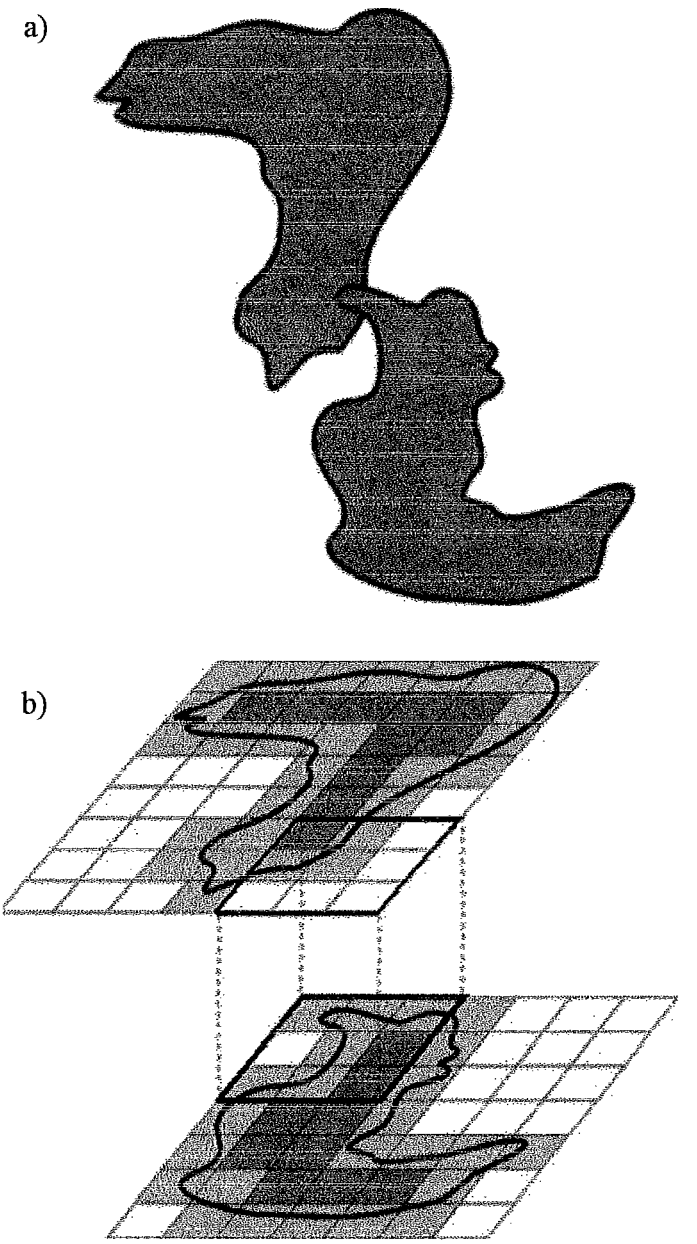


Figura 32. Exemplo de teste de interseção de assinaturas, com interseção

5. Avaliação experimental

Este seção é dedicada a apresentar os resultados experimentais referentes à avaliação do uso da Assinatura Raster de 3 Cores. A avaliação foi feita utilizando a assinatura no segundo passo da Arquitetura de Processamento de Consultas em Múltiplos Passos (MSQP), realizando as seguintes comparações:

- Comparação com a utilização da Assinatura Raster de 4 Cores também no segundo passo do MSQP, avaliando interseção de conjuntos de polígonos.
- Comparando com a Arquitetura em Dois Passos (correspondente ao MSQP sem o segundo passo). Neste caso, foram comparadas interseções de conjuntos de polígonos, polilinhas, pontos, e suas combinações.

5.1. Conjuntos de dados

O conjunto de dados dos polígonos utilizado nestes testes consistem em limites dos municípios brasileiros (IBGE, 2000). Com o intuito de simular um conjunto mais extenso de dados, os dados foram replicados, conforme sugerido em BRINKHOFF *et al.* (1994b). Os polígonos originais foram deslocados em uma distância aleatória sobre as coordenadas x e y (o conjunto gerado foi chamado de *municípios brasileiros'*). As características detalhadas dos dados utilizados estão descritas na Tabela 8. Foram analisados dados com tamanhos variáveis, entre 10Mb e 69Mb e média de 48Mb (aproximadamente). O número de objetos foi em torno de 4.700 objetos. O número médio de segmentos foi em torno de 399.000 segmentos (aproximadamente). O tamanho de cada objeto foi em média 83 segmentos / pontos. O conjunto de dados é composto de objetos de complexidade média – menos de 1000 segmentos por objeto. Quanto maior for a complexidade do objeto, maior o tempo gasto no terceiro passo da arquitetura (teste exato). Portanto, um maior ganho de desempenho pode ser esperado quando o conjunto de dados for composto de objetos mais complexos.

Tabela 8. Conjunto de dados utilizado para os testes experimentais

Conjunto de dados		Tamanho (KB)	Número de objetos	Número de segmentos / pontos	Número médio de segmentos / pontos
Polígonos	Municípios	63.818	4.645	399.002	85
	Municípios'	63.818	4.645	399.002	85
Polilinhas	Municípios	69.379	4.645	399.002	85
	Municípios'	69.379	4.645	399.002	85
Pontos	Grupo 1	9.533	5.000	400.000	80
	Grupo 2	9.533	5.000	400.000	80
Média		47.577	4.763	399.335	83

O conjunto de polilinhas utilizado nos testes foi derivado dos polígonos referentes aos municípios de Brasil e dos municípios replicados. Cada polígono foi dividido em duas polilinhas, gerando assim um conjunto com o dobro de objetos. Entretanto, cada objeto possui metade do tamanho dos polígonos originais.

O conjunto de pontos utilizado nos testes foi gerado de forma aleatória, apenas para demonstrar a possibilidade de uso da assinatura. Observe que estamos utilizando a representação de Pontos do Secondo, ou seja, cada objeto é formado por um conjunto de pontos. Foram gerados dois conjuntos de pontos, cada um possuindo 5.000 grupos de pontos. Cada grupo possuía em média 80 pontos.

5.2. Métricas utilizadas

As métricas utilizadas para os testes experimentais foram número de acessos a disco, espaço de armazenamento, tempo de geração da assinatura, número de pares identificados no segundo passo e tempo total de execução. Cada métrica será explicada em maiores detalhes nas seções seguintes.

5.2.1. Espaço de armazenamento

Bancos de dados espaciais comumente armazenam objetos muito complexos. Para armazenar tais objetos, sem perder nenhuma característica do mesmo, pode ser necessário um grande espaço em disco (ZIMBRAO *et al.*, 1998). Grandes volumes de dados em disco

implicam não apenas no uso de dispositivos com maiores capacidades de armazenamento, mas também aumentam o número de acessos a disco (seção 5.2.2), aumentando o tempo total de execução. Assim, a assinatura visa utilizar um espaço de armazenamento pequeno em relação ao objeto original. Vale ressaltar que nesta implementação não foi utilizado nenhum algoritmo de compactação. Tal algoritmo poderia ser utilizado, tornando o tamanho da assinatura ainda menor em relação ao objeto original.

É importante ressaltar que as assinaturas 3CRS e 4CRS exigem o mesmo espaço de armazenamento, já que são necessários dois bits para armazenar a cor de cada célula.

5.2.2. Número de acessos ao disco

Acesso ao disco pode ser uma atividade custosa, uma vez que a mídia de armazenamento utilizada é mais lenta que a utilizada para a memória de trabalho, exigindo um tempo maior para a execução da junção. Objetos grandes podem necessitar de mais acessos ao disco.

5.2.3. Tempo de geração da assinatura

A utilização da assinatura pode fazer cair o tempo total de execução de junções espaciais. Entretanto, para utilizar a assinatura é necessário antes gerá-la. A geração da assinatura pode ser feita previamente e armazenada junto com o objeto original, ou pode ser calculada *on-the-fly*, ou seja, quando for necessária. Nos testes foram comparados os tempos de geração da assinatura 3CRS e da assinatura 4CRS. O tempo de geração foi também comparado com o tempo total de execução da junção.

5.2.4. Porcentagem de pares analisados no teste exato

O teste de interseção de duas assinaturas pode gerar três resultados: *existe interseção* (*hit*), *não existe interseção* (*false hit*), ou *inconclusivo* (não é possível afirmar que há interseção, ou que não há interseção). Os resultados do tipo *inconclusivo* são repassados para o terceiro passo (teste exato), que é o mais custoso, em termos de utilização de CPU e de acessos ao disco.

Esta métrica indica a relação, em porcentagem, entre o número de pares processados no terceiro passo da arquitetura MSQP e o número de pares processados no segundo passo

na arquitetura de dois passos. Quanto menor for a porcentagem, maior será a contribuição da assinatura e maior será a eficiência da consulta.

5.2.5. Tempo total de execução

O tempo total de execução foi analisado, considerando o teste de MBR e o teste exato, no caso da arquitetura em dois passos, contra o tempo do teste de MBR, do teste da assinatura e do teste exato, no caso da arquitetura em três passos (para as assinaturas 3CRS e 4CRS).

5.3. Ambientes de teste

Os testes foram executados em um computador PC com processador Athlon XP 1600+ 1.4 GHz com 256 Mb de memória RAM.

Os algoritmos referentes à assinatura 3CRS foram implementados no sistema SECONDO (capítulo 3). Os dados foram carregados neste sistema e as junções realizadas e avaliadas dentro deste sistema. A assinatura 4CRS também foi implementada e utilizada neste sistema, para efeito de comparação ao utilizar objetos do tipo polígono.

5.4. Número máximo de células

O número de células na assinatura 3CRS (assim como na assinatura 4CRS) é configurável. Um maior número máximo de células implica em uma assinatura mais detalhada, onde um maior número de pares pode ser identificado no segundo passo. Entretanto, um maior número de células implica em um tempo maior de geração da assinatura, um espaço de armazenamento maior e um maior número de acessos ao disco. Maiores detalhes sobre o número de células podem ser encontrados na seção 4.1.1. De forma a melhor avaliar o número máximo ideal de células, os testes foram realizados utilizando diferentes valores. Os testes foram realizados quatro vezes, em cada uma delas utilizando um número máximo de células diferente: 250, 500, 1000 e 1500. Os resultados foram comparados e são apresentados nas seções a seguir.

5.5. Resultados experimentais

Nesta seção são apresentados os resultados dos testes executados, sobre os diferentes tipos de dados. Na seção 5.5.1 são apresentados os resultados obtidos na

avaliação do espaço de armazenamento utilizado pela assinatura 3CRS. As seções 5.5.2 até 5.5.7 apresentam os resultados obtidos nas junções entre as diferentes combinações de objetos.

5.5.1. Espaço de armazenamento

Os resultados dos testes experimentais demonstram que o espaço ocupado pelas assinaturas é muito pequeno em relação ao espaço utilizado pelos objetos originais, principalmente para os casos de polígonos e polilinhas. Logicamente, quanto maior o número máximo de células permitido na assinatura, maior o tamanho da assinatura. Mesmo assim, para todos os limites de células testados (250, 500, 1000 e 1500), o tamanho da assinatura se mostrou insignificante, podendo ser armazenado praticamente sem impacto no espaço em disco total do banco de dados.

Vale ressaltar que nesta implementação não foi utilizado nenhum tipo de compactação da assinatura. Alguns algoritmos podem ser empregados, sem grande impacto na leitura da assinatura, mas diminuindo ainda mais o espaço utilizado. Para sistemas com limitações de armazenamento, tais algoritmos podem ser empregados facilmente.

Nas seções 5.5.1.1, 5.5.1.2 e 5.5.1.3 são apresentados os resultados obtidos nos testes experimentais para o espaço em disco da assinatura, em comparação com o espaço em disco dos objetos originais, para polígono, polilinha e pontos, respectivamente.

5.5.1.1. Polígono

Além da assinatura 3CRS, a assinatura 4CRS também pode ser utiliza para representar objetos do tipo polígono. Do modo como foram implementadas dentro do sistema SECONDO, ambas as assinaturas utilizam o mesmo espaço em disco (2 bits para armazenar a cor de cada célula). Desta forma, o resultado das duas assinaturas foi o mesmo e é apresentado em conjunto. A Tabela 9 apresenta os valores obtidos para os números máximos de células utilizados (250, 500, 1000 e 1500). A tabela apresenta o tamanho total dos objetos, assim como o tamanho total das assinaturas geradas (ambos em bytes) e a relação entre eles. O tamanho total dos objetos é aproximadamente 127Mb, enquanto as assinaturas ocuparam entre 700Kb e 2Mb (aproximadamente).

Tabela 9. Comparação do espaço de armazenamento do objeto e da assinatura para polígonos

Número máximo de células	Tamanho do objeto original (bytes)	Tamanho da assinatura (3CRS / 4CRS) (bytes)	Relação assinatura / dado original (%)
250	127.637.120	716.448	0,56
500	127.637.120	1.070.184	0,84
1000	127.637.120	1.737.468	1,36
1500	127.637.120	2.332.332	1,83

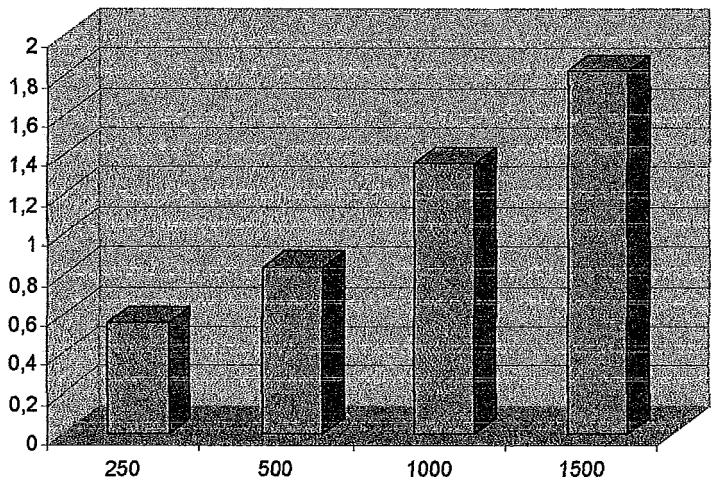


Figura 33. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação aos polígonos originais, para um máximo de 250, 500, 1000 e 1500 células

5.5.1.2. Polilinha

Uma limitação da assinatura 4CRS é capaz de representar apenas objetos do tipo polígono. A assinatura 3CRS, entretanto, é capaz de representar também objetos do tipo polilinha. Assim, os testes envolvendo este tipo de dados apresentam resultados apenas para a assinatura 3CRS.

Tabela 10. Comparação do espaço de armazenamento do objeto e da assinatura para polilinhas

Número máximo de células	Tamanho do objeto original (bytes)	Tamanho da assinatura (3CRS) (bytes)	Relação assinatura / dado original (%)
250	138.759.136	1.425.416	1,03
500	138.759.136	2.142.136	1,54
1000	138.759.136	3.468.272	2,50
1500	138.759.136	4.725.116	3,40

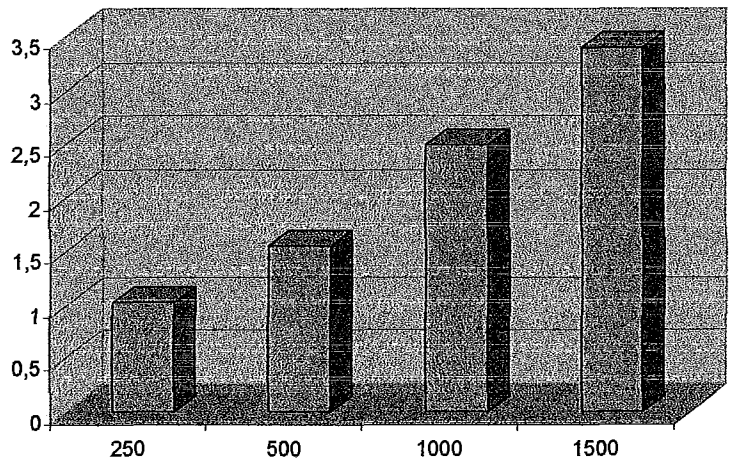


Figura 34. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação às polilinhas originais, para um máximo de 250, 500, 1000 e 1500 células

5.5.1.3. Pontos

Assim como objetos do tipo polilinha, a assinatura 4CRS não é capaz de representar objetos do tipo pontos. Deste modo, os testes para este tipo de objeto apresentam os resultados apenas para o uso da assinatura 3CRS.

Tabela 11. Comparação do espaço de armazenamento do objeto e da assinatura para pontos

Número máximo de células	Tamanho do objeto original (bytes)	Tamanho da assinatura (3CRS) (bytes)	Relação assinatura / dado orginal (%)
250	19.067.208	689.520	3,62
500	19.067.208	1.360.000	7,13
1000	19.067.208	1.587.392	8,33
1500	19.067.208	3.061.020	16,05

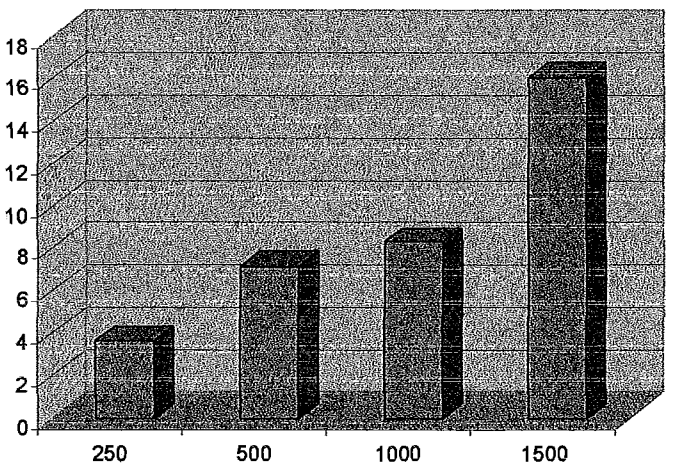


Figura 35. Espaço de armazenamento necessário (Mbytes) para assinaturas em relação aos pontos originais, para um máximo de 250, 500, 1000 e 1500 células

5.5.2. Polígono × Polígono

Esta seção apresenta os resultados obtidos através das junções entre polígonos contra polígonos. Do mesmo modo como foi apresentado na seção descritiva do espaço de armazenamento de polígonos (seção 5.5.1.1), para junções de polígonos também é possível utilizar assinaturas 4CRS. Assim, os testes foram realizados comparando o desempenho da arquitetura MSQP utilizando a assinatura 3CRS como segundo passo, contra o processamento utilizando a assinatura 4CRS como segundo passo, além de compararmos a o uso da 3CRS contra a Arquitetura de Dois Passos.

5.5.2.1. Junções utilizadas

Neste teste foram feitas as comparações entre os polígonos do objeto *municípios brasileiros* e os polígonos do objeto *municípios brasileiros'*. Os resultados, dentro de cada métrica analisada, são apresentados nas seções seguintes.

5.5.2.2. Tempo total de execução

A avaliação do tempo total de execução das junções – soma dos tempos gastos em cada passo de cada arquitetura – indicou que as assinaturas 3CRS e 4CRS apresentam um tempo semelhante, e ambos consideravelmente menores do que o tempo gasto utilizando a arquitetura de dois passos. A assinatura 3CRS apresentou um tempo total variando entre aproximadamente 27 e 38 porcento do tempo gasto na arquitetura de dois passos, dependendo do máximo de células utilizado. A assinatura 4CRS apresentou um tempo total variando entre aproximadamente 26 e 34 porcento do tempo da arquitetura de dois passos, também dependendo do número máximo de células utilizado. A Tabela 12 apresenta os tempos em cada arquitetura, em cada limite de células utilizado, assim como a relação entre o tempo das duas arquiteturas. Para a arquitetura de dois passos, o número máximo de células não é utilizado. Um gráfico comparativo é apresentado na Figura 36.

Tabela 12. Tempo gasto para realizar a junção entre polígonos

Método	Máximo de células	Tempo total (s)	Relação Assinatura / Dois passos (%)
3CRS	250	288,88	38,34
	500	221,01	29,33
	1000	203,28	26,98
	1500	234,28	31,10
4CRS	250	257,04	34,12
	500	204,60	27,16
	1000	198,94	26,41
	1500	229,78	30,50
Dois passos		753,43	

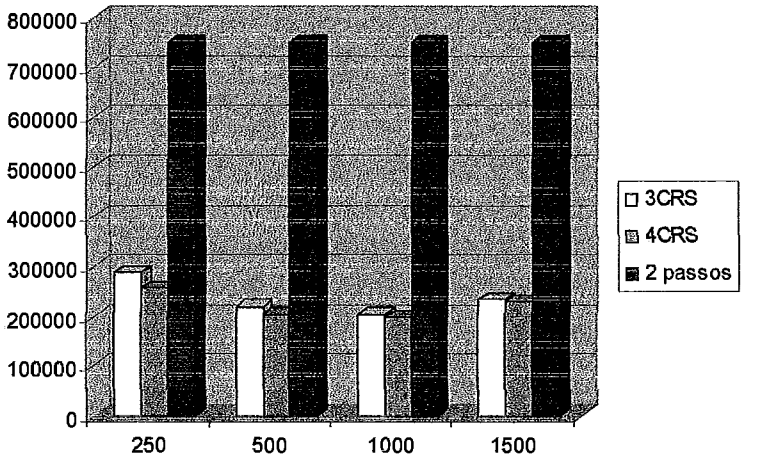


Figura 36. Tempo gasto (ms) para realizar a junção entre polígonos

5.5.2.3. Acessos ao disco

Ao utilizar as assinaturas 3CRS ou 4CRS, nota-se que há uma diminuição considerável no número de acessos ao disco. Essa diminuição se deve ao fato de que o maior responsável pelo grande número de acessos ao disco é o teste exato, onde é necessário ler o objeto original do disco. Utilizando assinaturas, são poucos os pares que chegam a executar este terceiro passo. Quanto maior o número máximo de células, menor o número de pares a serem testados no último passo (seção 5.5.2.4) e menor o número de acessos ao disco. Novamente podemos perceber que as assinaturas de 3 e 4 cores apresentam resultados muito semelhantes, com pequena vantagem para a assinatura de 4 cores, chegando a um menor número de resultados inconclusivos nos testes de assinaturas. Esta diferença entre as assinaturas se dá pelo fato que a assinatura de 4 cores é mais precisa, pois em alguns casos onde a assinatura de 3 cores tem como resultado inconclusivo, a assinatura de 4 cores pode indicar com certeza que houve uma interseção. É o caso da comparação entre duas células do tipo *Muito* na assinatura 4CRS (que garante que há uma interseção); na assinatura 3CRS, o equivalente seriam duas células *Inconclusivo* (que retorna um resultado inconclusivo). O resultado dos testes encontra-se na Tabela 13 e na Figura 37.

Tabela 13. Acessos ao disco para realizar as junções polígono x polígono (Relação entre arquitetura em três passos e arquitetura em dois passos)

Número máximo de células	3CRS / Dois passos (%)	4CRS / Dois passos (%)
250	23,94	22,45
500	19,31	18,44
1000	16,29	15,69
1500	15,46	14,99

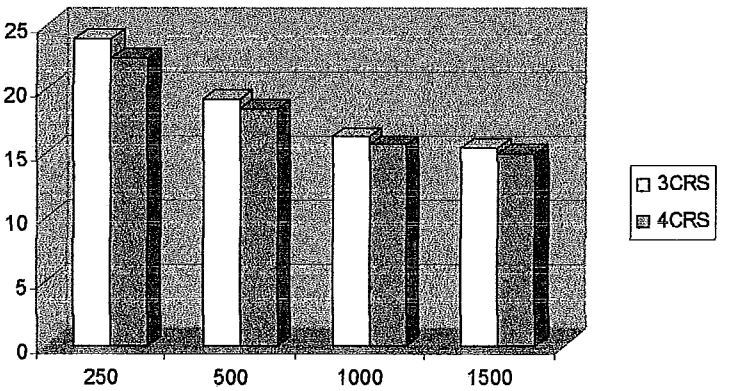


Figura 37. Acessos ao disco para realizar as junções polígono x polígono (Relação - % - entre arquitetura em três passos e arquitetura em dois passos)

5.5.2.4. Porcentagem de pares analisados no teste exato

Os testes demonstraram que uma porcentagem muito pequena dos pares envolvidos nas junções é analisada no teste exato. Tanto para a assinatura de 3 cores quanto para a assinatura de 4 cores, menos de 30% dos pares que seriam analisados no teste exato utilizando-se a arquitetura de dois passos foi realmente analisado no teste exato. Em alguns casos, este percentual chega a quase 10%. Isto significa que as assinaturas foram capazes de identificar se há ou não interseção na maioria dos casos. Nota-se também que neste caso o desempenho das duas assinaturas foi semelhante, com pequena vantagem para a 4CRS.

Tabela 14. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polígono (Relação entre arquitetura em 3 passos e arquitetura em dois passos)

Número máximo de células	3CRS / Dois passos (%)	4CRS / Dois passos (%)
250	26,92	24,42
500	18,62	17,16
1000	13,22	12,29
1500	10,99	10,23

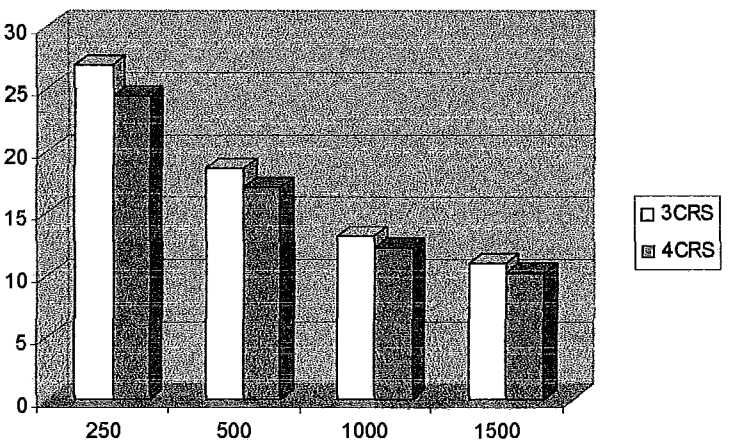


Figura 38. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polígono (Relação - % - entre arquitetura em três passos e arquitetura em dois passos)

5.5.2.5. Tempo de geração da assinatura

A geração da assinatura 3CRS é mais simples do que a geração da assinatura 4CRS, considerando o fato que na primeira, quando uma célula possui interseção com um segmento do objeto, a célula é marcada como *Inconclusiva*; na segunda, entretanto, ainda é necessário computar a área do polígono dentro da célula (o que corresponde a um *clipping* do polígono pela célula) para verificar se a célula é do tipo *Muito* ou *Pouco*. Este cálculo da área pode ser muito custo em relação ao resto da geração da assinatura. Os testes experimentais demonstraram que a assinatura 3CRS é gerada mais rapidamente, chegando em alguns casos a menos de 60% do tempo de geração da outra assinatura, conforme os

dados apresentados na Tabela 15. A Figura 39 apresenta um gráfico comparativo dos valores obtidos, de acordo com o número máximo de células utilizado.

Tabela 15. Tempo de geração da assinatura dos objetos envolvidos na junção polígono – comparação entre 3CRS e 4CRS

Número máximo de células	3CRS (s)	4CRS (s)	3CRS / 4CRS (%)
250	8,178	14,229	57,47
500	12,717	19,477	65,29
1000	21,166	29,121	72,68
1500	28,807	37,603	76,61

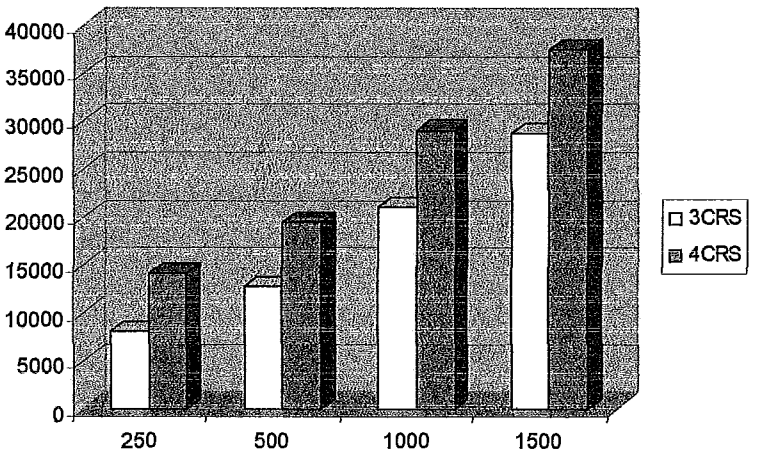


Figura 39. Tempo (ms) de geração da assinatura dos objetos envolvidos na junção polígono – comparação entre 3CRS e 4CRS

5.5.3. Polígono × Polilinha

O próximo teste realizado foi a junção de objetos do tipo polígono com objetos do tipo polilinha. Uma vez que assinaturas do tipo 4CRS não podem ser utilizadas para representar polilinhas, as comparações foram apenas entre a utilização da 3CRS como segundo passo da arquitetura MSQP e a arquitetura de dois passos.

5.5.3.1. Junções utilizadas

Neste teste, foi realizada a junção entre os polígonos representando os municípios brasileiros e as polilinhas geradas a partir dos municípios replicados (municípios'), conforme descrito na seção 5.1.

5.5.3.2. Tempo total de execução

O resultado dos testes realizados entre estes dois tipos de objetos demonstrou que a utilização da assinatura 3CRS representa uma diminuição considerável do tempo total de execução das junções, conforme descrito na Tabela 16. O tempo utilizado pela assinatura 3CRS variou entre 381 e 444 segundos (aproximadamente), variando de acordo com o número máximo de células utilizado, enquanto que a arquitetura de dois passos utilizou 853 segundos (aproximadamente). A economia no tempo foi de aproximadamente 50%. A Figura 40 apresenta um gráfico comparativo.

Tabela 16. Tempo gasto para realizar as junções polígono x polilinha

Método	Tempo (s)	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	444,468	52,09
3CRS com máximo de 500 células	396,680	46,49
3CRS com máximo de 1000 células	381,671	44,73
3CRS com máximo de 1500 células	389,335	45,63
Dois passos	853,303	

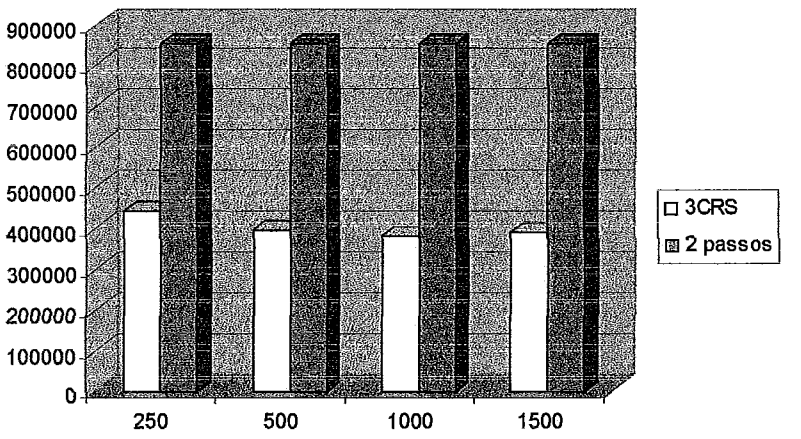


Figura 40. Tempo (ms) gasto para realizar as junções polígono x polilinha

5.5.3.3. Acessos ao disco

Assim como ocorrido no teste de junção entre polígonos, os experimentos demonstraram quem também houve uma redução significativa no número de acessos ao disco na comparação entre polígonos e polilinhas, conforme descrito na Tabela 17. O teste utilizando a assinatura 3CRS realizou entre 29 e 46 porcento do número de acessos ao disco utilizado pela arquitetura de dois passos (valores aproximados). Como pode ser observado no gráfico da Figura 41, a redução chega a mais de 70% (menos de 30% do número de acessos para arquitetura de dois passos).

Tabela 17. Acessos ao disco para realizar as junções polígono x polilinha

Método	Acessos	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	64.007.029	45,94
3CRS com máximo de 500 células	52.127.188	37,41
3CRS com máximo de 1000 células	44.315.308	31,81
3CRS com máximo de 1500 células	41.017.059	29,44
Dois passos	139.333.217	

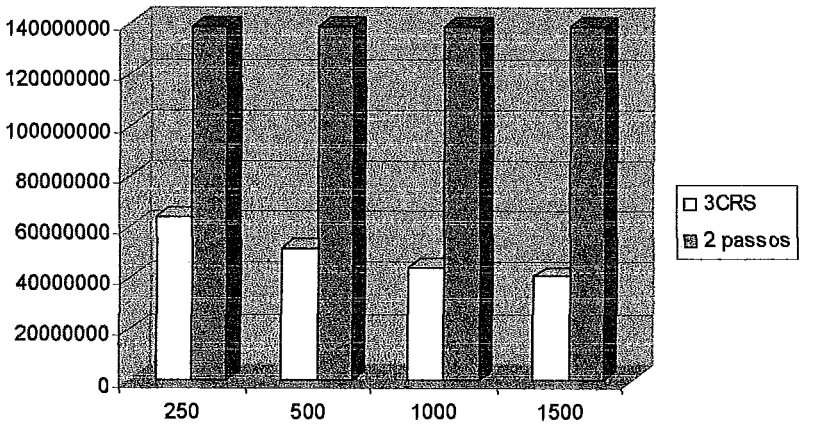


Figura 41. Acessos ao disco para realizar as junções polígono x polilinha

5.5.3.4. Porcentagem de pares analisados no teste exato

O número de pares analisados no teste exato, assim como no teste entre polígonos, também é significantemente reduzido no teste de junção entre polígono e polilinhas. A Tabela 18, que apresenta o resultado do teste experimental, demonstra que o ganho sobre número de pares analisados utilizando-se a assinatura foi de mais de 80% em alguns casos (para o uso de um máximo de 1000 ou 1500 células). A utilização da assinatura 3CRS fez com que no terceiro passo fossem avaliados apenas 14 a 34 porcento dos pares que seriam processados no teste exato utilizando a arquitetura de dois passos.

Tabela 18. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polilinha

Método	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	33,67
3CRS com máximo de 500 células	23,81
3CRS com máximo de 1000 células	17,25
3CRS com máximo de 1500 células	13,94

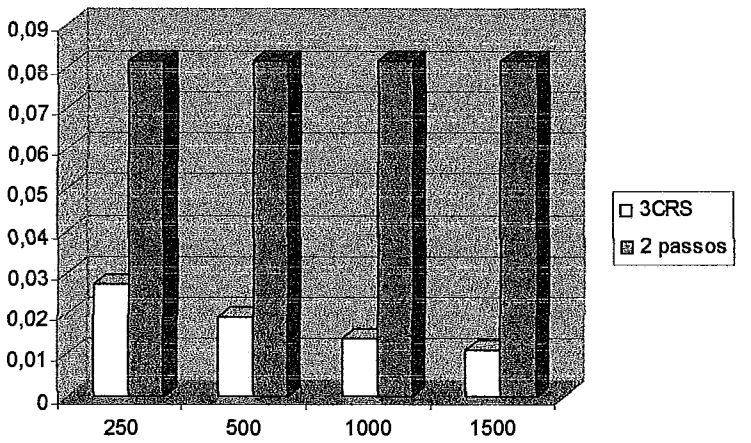


Figura 42. Porcentagem de pares analisados no teste exato para realizar as junções polígono x polilinha

5.5.3.5. Tempo de geração da assinatura

O tempo de geração das assinaturas nos testes realizados, conforme demonstra a Tabela 19, é muito pequeno em relação ao tempo utilizado para realizar as junções. O tempo de geração ficou em torno de 20 segundos, variando de acordo com o máximo de células utilizado (entre 10 e 35 segundos, aproximadamente). A Figura 43 mostra um gráfico para facilitar a comparação entre o tempo necessário para a geração das assinaturas para cada limite de células (250, 500, 1000, 1500).

Tabela 19. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x polilinha

Máximo de células	Tempo de geração (s)
250	10,355
500	15,274
1000	23,924
1500	35,096

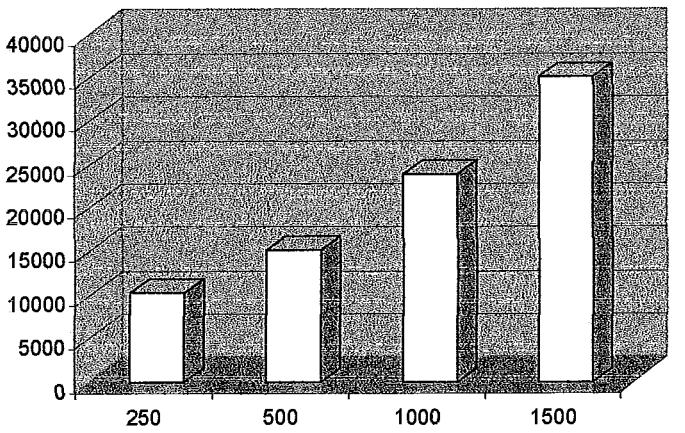


Figura 43. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x polilinha (ms)

Vale ressaltar que, enquanto o tempo para gerar a assinatura oscilou em torno de 20 segundos, o tempo para realizar a junção ficou em torno de 400 segundos utilizando assinatura e 850 segundos para a arquitetura de dois passos. Com isso, podemos afirmar que a assinatura pode ser calculada sob demanda (*on the fly*), e não previamente, e ainda assim a melhora no desempenho se mostra considerável. A Figura 44 exibe um gráfico comparativo, considerando para cada limite de células utilizados o tempo de geração da assinatura somado ao tempo da junção propriamente dito e para a arquitetura de dois passos apenas o tempo da junção. Observa-se que a geração da assinatura representa uma parte pequena do total.

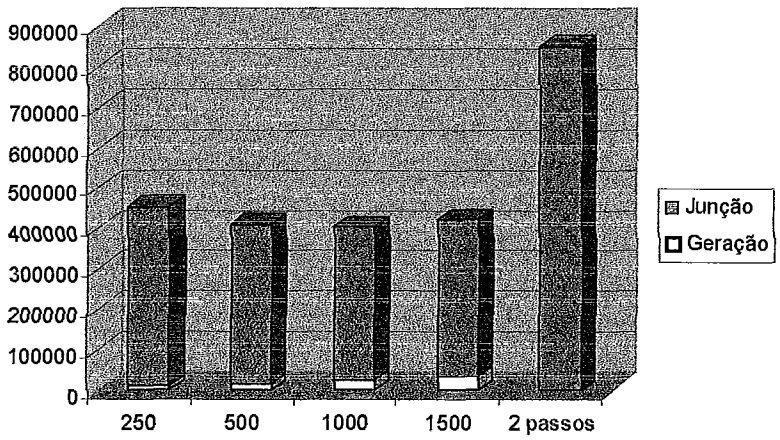


Figura 44. Tempo total da junção (ms), considerando a geração da assinatura *on-the-fly*

5.5.4. Polígono × Pontos

O teste experimental comparativo entre objetos do tipo polígono e objetos do tipo pontos também foi realizado. Este teste demonstrou mais uma vez a eficiência, sob as métricas analisadas, da utilização da assinatura como segundo passo da arquitetura MSQP, em comparação com a arquitetura tradicional de dois passos. Os resultados dentro de cada métrica são apresentados nas seções seguintes.

5.5.4.1. Junções utilizadas

Para a realização deste teste, foi executada a junção entre os polígonos que representam os municípios do Brasil e um conjunto de pontos gerado aleatoriamente, conforme descrito na seção 5.1.

5.5.4.2. Tempo total de execução

Em relação ao tempo total de execução, a junção utilizando a arquitetura MSQP com a assinatura de 3 cores apresentou uma melhora em relação à arquitetura de dois passos, embora a melhora não tenha sido tão expressiva quanto nos testes entre polígonos e entre polígonos e polilinhas. Isto se deve ao fato de que o objetivo da assinatura é evitar que o teste exato, mais custoso, seja executado. No caso de pontos, o teste exato não é tão custoso quanto os outros dois citados. Outro fator é que estes objetos não possuem área, logo não possuem células do tipo *Cheio*, o que leva a mais resultados inconclusivos.

Mesmo assim, a utilização da assinatura é recomendada, já que pode reduzir o tempo de comparação em cerca de 20%, conforme detalhado na Tabela 20. A junção utilizando a assinatura 3CRS utilizou entre 365 e 388 segundos, aproximadamente, enquanto a arquitetura de dois passos utilizou aproximadamente 465 segundos.

Tabela 20. Tempo gasto para realizar as junções polígono x pontos

Método	Tempo (s)	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	388,752	83,58
3CRS com máximo de 500 células	367,657	79,05
3CRS com máximo de 1000 células	367,357	78,98
3CRS com máximo de 1500 células	365,711	78,63
Dois passos	465,110	

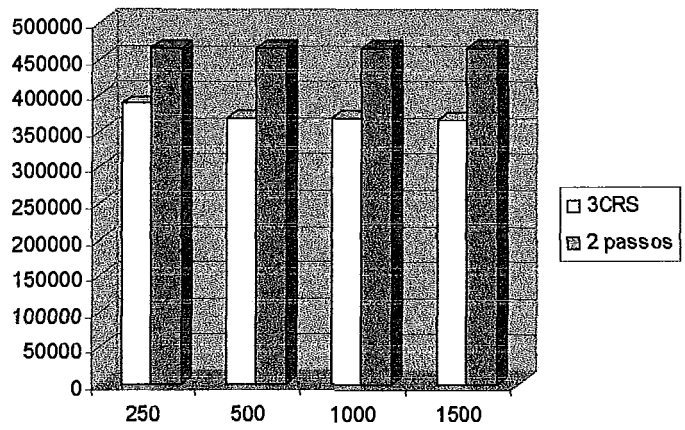


Figura 45. Tempo gasto para realizar as junções polígono x pontos (ms)

5.5.4.3. Acessos ao disco

Assim como ocorreu analisando-se a métrica de tempo total de execução, a quantidade de acessos ao disco também sofreu uma redução, embora não tão acentuada quanto a apresentada nos testes entre polígonos e entre polígonos e polilinhas. Isto se deve ao fato de que a representação de objetos do tipo pontos é muito pequena. Assim, apenas os objetos do tipo polígono presentes na junção apresentam uma relevância significativa no número de acessos ao disco. Os valores resultantes dos testes encontram-se na Tabela 21 e

na Figura 46. Os testes utilizando a assinatura 3CRS realizaram entre 29 e 31 milhões de acessos (aproximadamente), enquanto a arquitetura de dois passos realizou aproximadamente 32 milhões de acessos ao disco.

Tabela 21. Acessos ao disco para realizar as junções polígono x pontos

Método	Acessos	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	29.655.534	93,80
3CRS com máximo de 500 células	28.888.689	91,37
3CRS com máximo de 1000 células	29.614.335	93,67
3CRS com máximo de 1500 células	30.722.989	97,17
Dois passos	31.616.480	

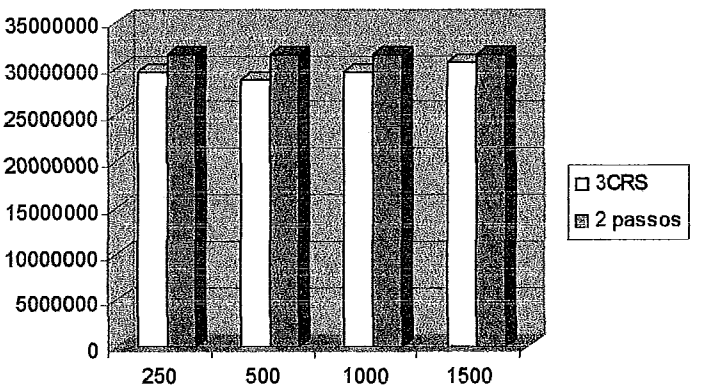


Figura 46. Acessos ao disco para realizar as junções polígono x pontos

5.5.4.4. Porcentagem de pares analisados no teste exato

Conforme indicam os resultados dos testes experimentais apresentados na Tabela 22 e na Figura 47, o número de pares onde foi necessário utilizar o teste exato foi cerca de 60% menor quando a assinatura 3CRS foi utilizada.

Tabela 22. Porcentagem de pares analisados no teste exato para realizar as junções polígono x pontos

Método	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	77,42
3CRS com máximo de 500 células	63,93
3CRS com máximo de 1000 células	53,99
3CRS com máximo de 1500 células	46,13

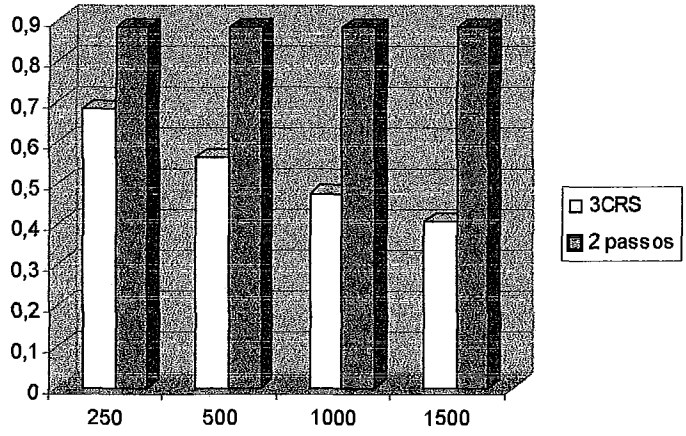


Figura 47. Porcentagem de pares analisados no teste exato para realizar as junções polígono x pontos

5.5.4.5. Tempo de geração da assinatura

O tempo de geração da assinatura apresentado foi muito pequeno, já que a geração para objetos do tipo pontos é muito rápida, uma vez que o objeto original é pequeno (rápido de se ler) e não há a necessidade de se calcular células do tipo cheio. Os valores obtidos para a geração das assinaturas encontram-se na Tabela 23 e na Figura 48, divididos de acordo com o número máximo de células utilizado (250, 500, 1000, 1500).

Tabela 23. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x pontos

Máximo de células	Tempo de geração (s)
250	6,931
500	8,858
1000	13,037
1500	19,094

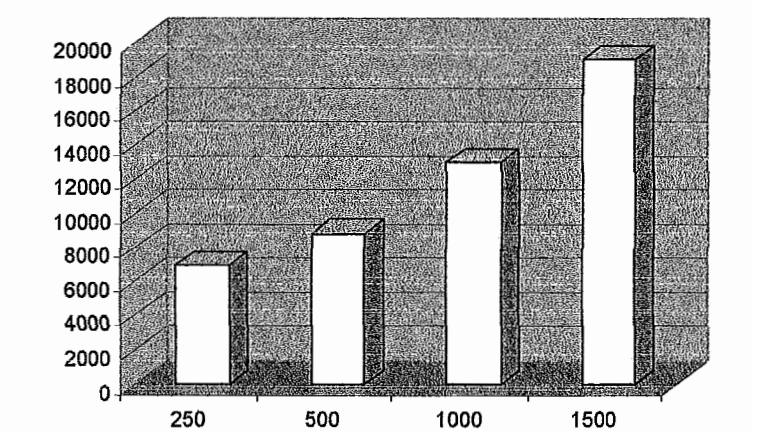


Figura 48. Tempo de geração da assinatura dos objetos envolvidos na junção polígono x pontos (ms)

5.5.5. Linha × Linha

O próximo teste realizado foi o teste de junção entre objetos do tipo polilinha. O teste foi executado e foram comparados os resultados utilizando-se a assinatura 3CRS e utilizando-se a arquitetura em dois passos. Os resultados são apresentados nas seções a seguir.

5.5.5.1. Junções utilizadas

Para a realização deste teste, foram realizadas as junções entre as polilinhas geradas a partir dos municípios brasileiros e as polilinhas geradas a partir da replicação dos municípios brasileiros, conforme descrito na seção 5.1.

5.5.5.2. Tempo total de execução

Os testes demonstraram que o tempo total de execução é reduzido quando a proposta da assinatura 3CRS é utilizada. Como demonstrado na Tabela 24. O tempo total foi reduzido em cerca de 15%, variando de acordo com o número máximo de células utilizado. A diferença entre as duas abordagens não foi tão expressiva quanto no caso do teste entre dois polígonos, uma vez que, como os objetos não possuem área e logo não possuem células do tipo *Cheio*, então o teste da assinatura não pode determinar com certeza que há interseção. O teste sobre as assinaturas pode apenas ter como resultado *Não há interseção* ou *Inconclusivo*. Além disso, o teste exato dos objetos é mais simples que o teste envolvendo dois polígonos, pois neste teste não há a necessidade de se avaliar se um objeto está totalmente contido no outro; apenas as arestas são testadas para verificar uma possível interseção. Assim, o tempo ganho evitando-se a execução do teste exato não é tão grande. Um gráfico comparativo do teste realizado encontra-se na Figura 49.

Tabela 24. Tempo gasto para realizar as junções polilinha x polilinha

Método	Tempo (s)	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	1.069,51	87,89
3CRS com máximo de 500 células	1.047,54	86,09
3CRS com máximo de 1000 células	1.032,60	84,86
3CRS com máximo de 1500 células	1.002,70	82,41
Dois passos	1.216,79	

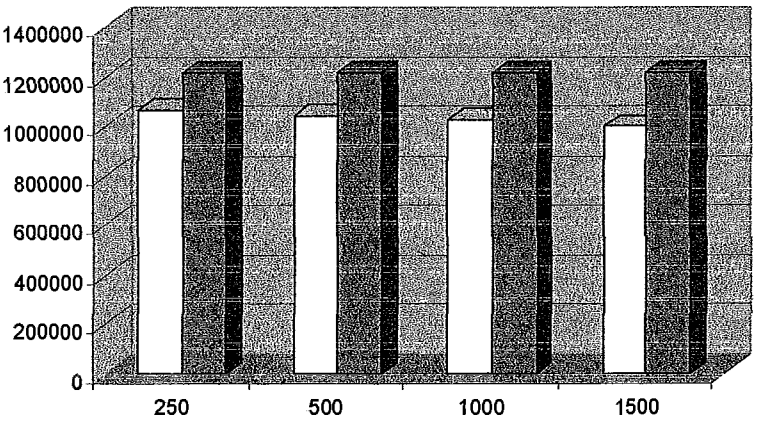


Figura 49. Tempo gasto para realizar as junções polilinha x polilinha (ms)

5.5.5.3. Acessos ao disco

O número de acessos ao disco sofreu uma queda entre 58 e 64 porcento (aproximadamente) nos testes realizados, quando a assinatura 3CRS foi utilizada. Os valores em detalhes encontram-se na Tabela 25 e na Figura 50.

Tabela 25. Acessos ao disco para realizar as junções polilinha x polilinha

Método	Acessos	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	77.781.967	64,49
3CRS com máximo de 500 células	73.548.009	60,98
3CRS com máximo de 1000 células	71.330.528	59,15
3CRS com máximo de 1500 células	70.429.188	58,40
Dois passos	120.602.280	

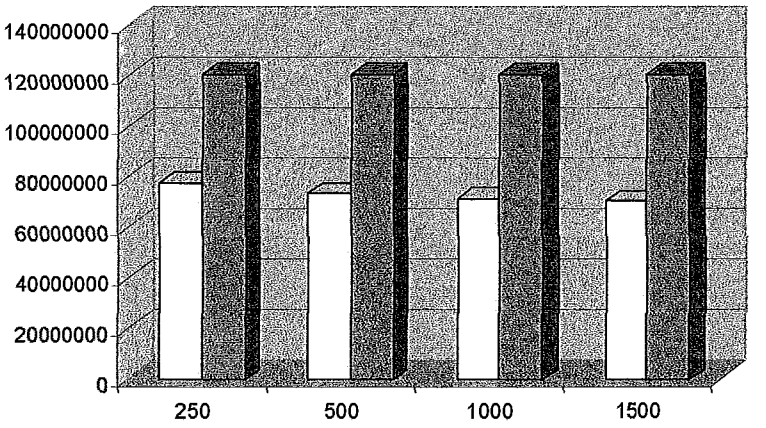


Figura 50. Acessos ao disco para realizar as junções polilinha x polilinha

5.5.5.4. Porcentagem de pares analisados no teste exato

Mesmo com a limitação do teste entre duas assinaturas 3CRS geradas a partir de polilinhas de não ser possível um resultado *Há interseção* (como não há área e logo não há células do tipo *Cheio* na assinatura, o teste pode apenas retornar *Não há interseção* ou *Inconclusivo*), a utilização da assinatura fez com que houvesse uma redução de 50 a 59 porcento (aproximadamente) do número de pares avaliados no teste exato. Os resultados do teste encontram-se na Tabela 26 e na Figura 51.

Tabela 26. Porcentagem de pares analisados no teste exato para realizar as junções polilinha x polilinha

Método	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	59,36
3CRS com máximo de 500 células	54,94
3CRS com máximo de 1000 células	51,96
3CRS com máximo de 1500 células	50,37

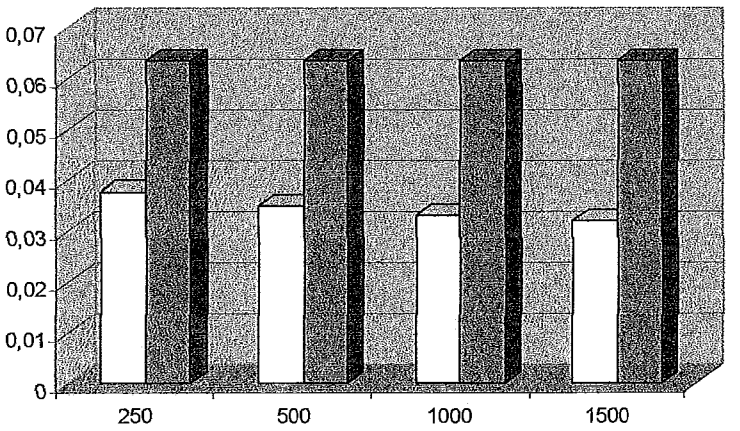


Figura 51. Porcentagem de pares analisados no teste exato para realizar as junções polilinha X polilinha

5.5.5.5. Tempo de geração da assinatura

O tempo de geração da assinatura foi muito pequeno em relação ao tempo total de execução da junção. Enquanto o tempo de geração variou entre 11 e 40 segundos, o tempo de execução variou entre 1.003 e 1.070 segundos (valores aproximados), conforme mostra a seção 5.5.5.2. Como a geração foi realizada apenas sobre objetos do tipo polilinha, não houve a necessidade de calcular as células do tipo *Cheio,* agilizando a geração. Os tempos utilizados para a geração nos testes encontram-se descritos na Tabela 27 e na Figura 52.

Tabela 27. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X polilinha

Máximo de células	Tempo de geração (s)
250	11,570
500	18,704
1000	27,704
1500	40,768

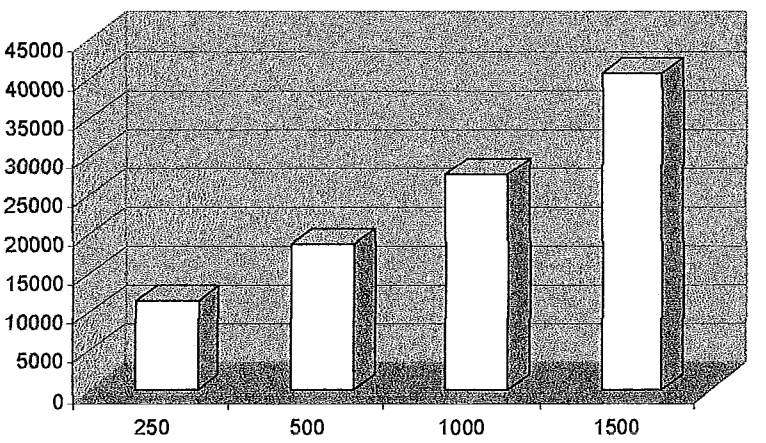


Figura 52. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X polilinha (ms)

5.5.6. Linha × Pontos

A próxima comparação realizada nos testes experimentais foi a junção entre objetos do tipo polilinha com objetos do tipo pontos, com a utilização da assinatura 3CRS e com a utilização da arquitetura de dois passos. Os resultados destas junções são apresentados nas seções a seguir.

5.5.6.1. Junções utilizadas

Para a realização deste teste, foram utilizadas polilinhas geradas a partir dos municípios brasileiros, contra grupos de pontos gerados de forma aleatórias, como descrito na seção 5.1.

5.5.6.2. Tempo total de execução

Da mesma forma como ocorreu no tempo total de execução na junção de polilinhas, a junção entre polilinhas e pontos utilizando a assinatura 3CRS representou uma melhora sobre a arquitetura de dois passos. Novamente a melhora no desempenho não foi tão expressiva quanto a observada entre polígonos, mas mesmo assim o tempo gasto com a utilização da assinatura 3CRS ficou entre 64 e 80 porcento (valores aproximados) do tempo

utilizando-se a arquitetura de dois passos. A Tabela 28 e a Figura 53 apresentam os resultados nessa métrica em detalhes.

Tabela 28. Tempo gasto para realizar as junções polilinha x pontos

Método	Tempo (s)	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	8.925,79	80,53
3CRS com máximo de 500 células	8.102,73	73,11
3CRS com máximo de 1000 células	7.725,44	69,70
3CRS com máximo de 1500 células	7.114,10	64,19
Dois passos	11.083,55	

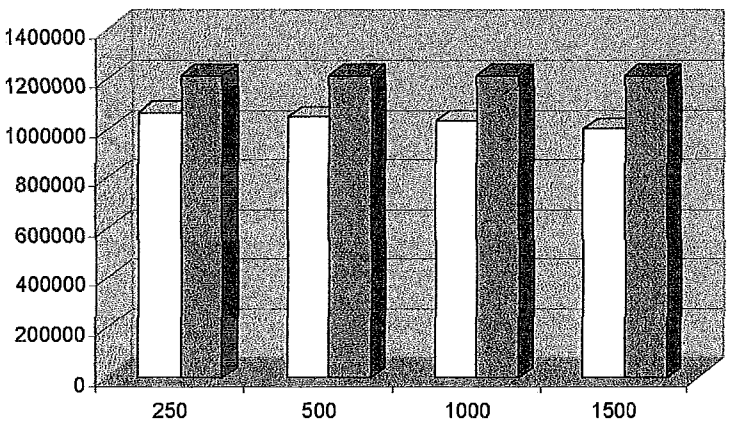


Figura 53. Tempo gasto para realizar as junções polilinha x pontos (ms)

5.5.6.3. Acessos ao disco

Novamente, foi observada uma redução do número de acessos ao disco utilizando-se a proposta apresentada. Com a utilização da assinatura 3CRS, o número de acessos ao disco ficou entre 81 e 92 porcento (valores aproximados) do número de acessos ao disco utilizado pela arquitetura de dois passos. A Tabela 29 e a Figura 54 apresentam os resultados do teste.

Tabela 29. Acessos ao disco para realizar as junções polilinha x pontos

Método	Acessos	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	339.855.852	91,75
3CRS com máximo de 500 células	323.835.141	87,43
3CRS com máximo de 1000 células	310.955.503	83,95
3CRS com máximo de 1500 células	299.515.988	80,86
Dois passos	370.411.103	

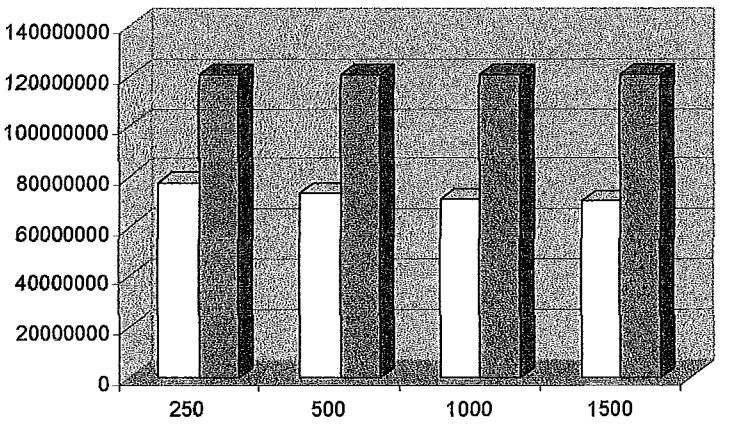


Figura 54. Acessos ao disco para realizar as junções polilinha x pontos

5.5.6.4. Porcentagem de pares analisados no teste exato

A assinatura 3CRS desempenhou seu papel de filtrar alguns pares para evitar que sejam analisados no teste exato. Como era de se esperar, novamente quanto maior o número de células, menor a quantidade de pares analisados no teste exato. O número de pares analisados utilizando-se a assinatura 3CRS ficou entre 59 e 83 porcento do número de pares no teste exato utilizando-se a arquitetura de dois passos (valores aproximados). Os valores em detalhes sobre esta métrica são apresentados na Tabela 30 e na Figura 55.

Tabela 30. Porcentagem de pares analisados no teste exato para realizar as junções polilinha x pontos

Método	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	82,68
3CRS com máximo de 500 células	71,76
3CRS com máximo de 1000 células	64,53
3CRS com máximo de 1500 células	58,19

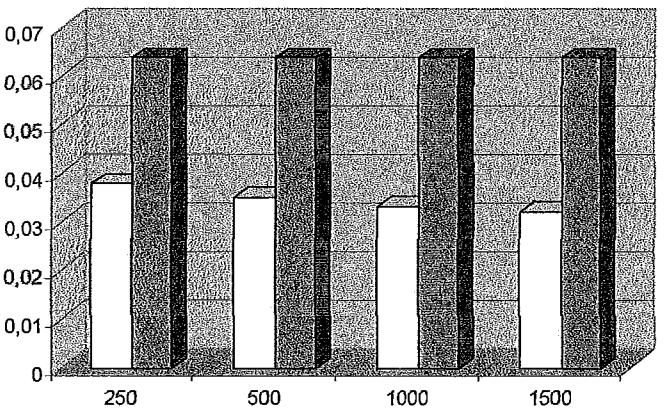


Figura 55. Porcentagem de pares analisados no teste exato para realizar as junções polilinha X pontos

5.5.6.5. Tempo de geração da assinatura

Analogamente ao observado no tempo de geração da assinatura para a junção de objetos do tipo polilinha, a geração envolvendo polilinha e pontos foi muito rápida em comparação ao tempo utilizado para realizar a junção. O tempo de geração ficou entre 6,83 e 17 segundos, enquanto o tempo da junção foi em média 8.590 segundos (aproximadamente). Os valores do tempo de geração encontram-se na Tabela 31 e na Figura 56.

Tabela 31. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X pontos

Máximo de células	Tempo de geração (s)
250	6,83
500	8,26
1000	12,16
1500	17,00

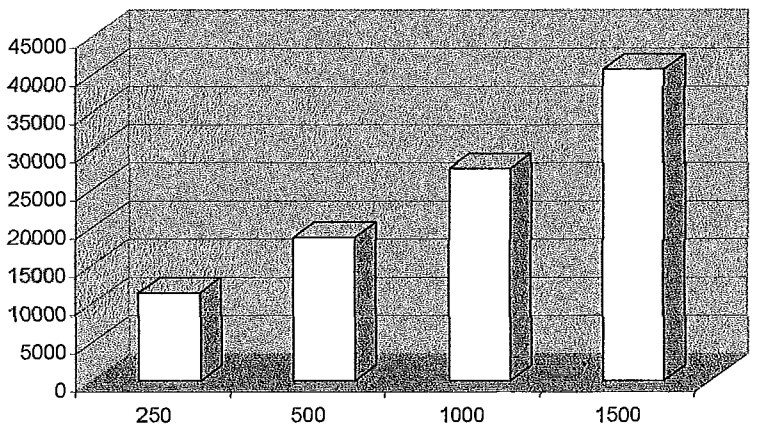


Figura 56. Tempo de geração da assinatura dos objetos envolvidos na junção polilinha X pontos (ms)

5.5.7. Pontos × Pontos

O último teste realizado foi a junção de objetos do tipo pontos, comparando a utilização da assinatura 3CRS com a arquitetura de dois passos. Este teste demonstrou uma diferença fundamental em relação aos outros testes: apesar de também ser uma comparação possível, para interseções entre dois objetos do tipo pontos, normalmente não é aconselhável o uso da assinatura 3CRS. Isto se dá pelo fato de que o teste exato de interseção entre dois objetos do tipo pontos é muito simples (basta comparar a igualdade dos pontos, não é necessário calcular interseção de segmentos de reta como no caso de polígonos e polilinhas). Como o objetivo da assinatura é evitar o teste exato, este objetivo

não representa um ganho significativo neste caso. Na verdade, o próprio teste da assinatura pode representar um cálculo extra, aumentando o tempo total da junção.

5.5.7.1. Junções utilizadas

Para este teste, foram utilizados dois conjuntos de pontos gerados aleatoriamente, conforme indicado na seção 5.1.

5.5.7.2. Tempo total de execução

O teste exato de interseção entre dois objetos do tipo pontos é muito simples. Assim, o tempo necessário para avaliar as assinaturas dos objetos pode ser muito próximo ao tempo utilizado para executar o teste exato. Como objetos do tipo pontos não possuem área, as suas assinaturas não possuem células do tipo *Cheio*. Desse modo, o teste das assinaturas não pode ter como resultado *Há interseção* (apenas pode resultar *Inconclusivo* ou *Não há interseção*). Com isso, muitos pares acabam sendo analisados na assinatura e também no teste exato. Estes fatores fazem com que em alguns casos o tempo gasto utilizando a assinatura 3CRS pode ser maior do que o tempo gasto utilizando a arquitetura de dois passos. Estes casos foram identificados nos testes experimentais, para o máximo de 500, 1000 e 1500 células. Para estes casos, o tempo utilizando a assinatura ficou entre 5.000 e 7.000 segundos, enquanto a arquitetura de dois passos necessitou de 4.300 segundos (valores aproximados). Os valores obtidos nos testes são apresentados na Tabela 32 e na Figura 57.

Tabela 32. Tempo gasto para realizar as junções pontos x pontos

Método	Tempo (s)	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	4.280,18	97,88
3CRS com máximo de 500 células	4.973,29	113,74
3CRS com máximo de 1000 células	5.303,46	121,29
3CRS com máximo de 1500 células	7.102,50	162,43
Dois passos	4.372,68	

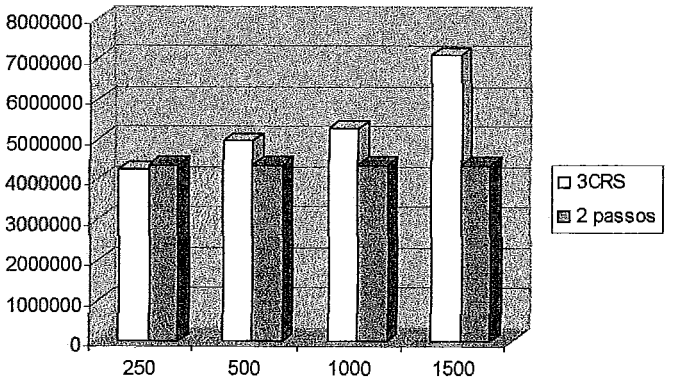


Figura 57. Tempo gasto para realizar as junções pontos x pontos (ms)

5.5.7.3. Acessos ao disco

Como a geração dos pontos para os testes foi aleatória, então o número de células to tipo *Inconclusivo* nas assinaturas foi muito grande. Assim, a grande maioria dos pares foi analisado também no teste exato (seção 5.5.7.4). Com isso, são necessários acessos ao disco para recuperar a assinatura e outros acessos para recuperar também a representação exata do objeto. Neste teste, a exceção foi para um máximo de 1500 células. Para este caso, muitos pares foram identificados no teste das assinaturas e a representação exata não foi acessada. Apesar da representação de objetos do tipo pontos ser simples (como os pontos são em duas dimensões, são necessários dois números para cada ponto), para a representação da assinatura, são utilizados apenas dois bits para cada célula. Assim, a recuperação das assinaturas necessita de menos acessos ao disco que a representação do objeto original.

Tabela 33. Acessos ao disco para realizar as junções pontos x pontos

Método	Acessos	Relação 3CRS / Dois passos (%)
3CRS com máximo de 250 células	2.455.466.912	99,94
3CRS com máximo de 500 células	2.437.169.277	99,19
3CRS com máximo de 1000 células	2.389.189.649	97,24
3CRS com máximo de 1500 células	930.223.553	37,86
Dois passos	2.457.055.578	

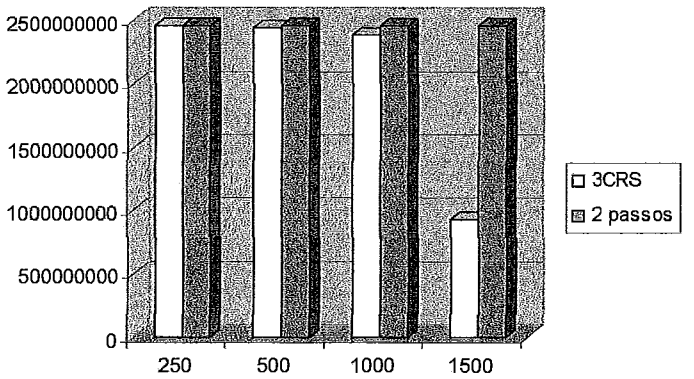


Figura 58. Acessos ao disco para realizar as junções pontos x pontos

5.5.7.4. Porcentagem de pares analisados no teste exato

Para o teste de pontos, os objetos foram gerados de forma aleatória, distribuídos em um espaço determinado, o que fez com que cada grupo possuísse aproximadamente o mesmo MBR. Com isso, o teste de MBR não foi capaz de eliminar nenhum par. Assim, para a arquitetura de dois passos, todos os pares foram analisados no teste exato. Para a utilização da assinatura 3CRS, o resultado não foi muito diferente, uma vez que pontos muito espalhados no espaço resultaram em muitas células do tipo *Inconclusivo*, e o teste entre duas células do tipo *Inconclusivo* fazem com que o par seja analisado no teste exato. A porcentagem de pares analisados utilizando-se a assinatura 3CRS (Tabela 34) ficou entre 85,6 e 99,9 porcento (aproximadamente). A Figura 59 apresenta um gráfico comparativo entre a arquitetura de dois passos e a arquitetura MSQP com a assinatura 3CRS como filtro.

Tabela 34. Porcentagem de pares analisados no teste exato para realizar as junções pontos x pontos

Método	Porcentagem
3CRS com máximo de 250 células	99,99
3CRS com máximo de 500 células	99,81
3CRS com máximo de 1000 células	99,33
3CRS com máximo de 1500 células	85,59
Dois passos	100,00

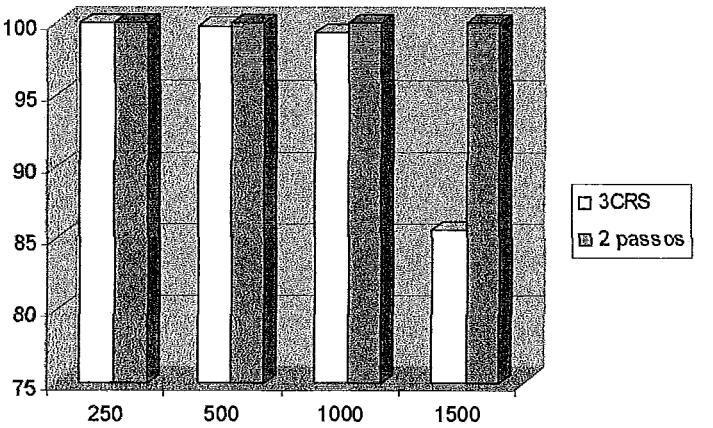


Figura 59. Porcentagem de pares analisados no teste exato para realizar as junções pontos X pontos

5.5.7.5. Tempo de geração da assinatura

Da mesma forma que a geração para objetos do tipo polilinha, a geração para objetos do tipo pontos utilizou um tempo muito pequeno em relação ao tempo total da junção. Enquanto o tempo gasto na junção ficou em torno de 4.700 segundos (em média), o tempo de geração (Tabela 35) variou entre 6 e 19 segundos (aproximadamente). A Figura 60 apresenta um gráfico comparativo entre os números máximos de células utilizados.

Tabela 35. Tempo de geração da assinatura dos objetos envolvidos na junção pontos X pontos

Máximo de células	Tempo de geração (s)
250	6,50
500	7,21
1000	14,81
1500	19,11

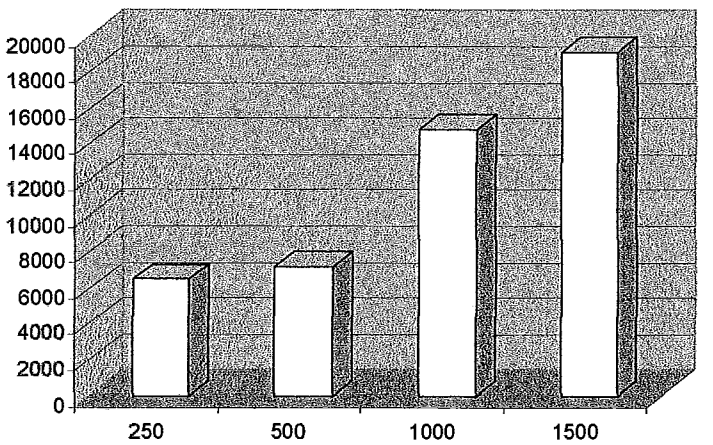


Figura 60. Tempo de geração da assinatura dos objetos envolvidos na junção pontos X pontos (ms)

6. Conclusão

O processamento de operações espaciais é muito custoso. A literatura apresenta diferentes abordagem para o processamento de operações espaciais. Este trabalho propôs uma nova assinatura raster para aproximação de objetos espaciais, Assinatura Raster de 3 Cores (*Three-Color Raster* Signature – 3CRS). Esta proposta consiste em discretizar o espaço que contém o objeto em células de uma grade, marcando cada célula com uma dentre três cores, de acordo com a ocupação da célula pelo objeto sendo representado. O objetivo é que esta representação possa ser utilizada para agilizar junções espaciais. Para este fim, esta representação pode ser utilizada no segundo passo da Arquitetura de Processamento de Consultas em Múltiplos Passos (MSQP) (BRINKHOFF *et al.*, 1994b). A definição da assinatura, assim como os algoritmos para geração da mesma e para a comparação entre duas assinaturas foram apresentados na seção 4. Para a implementação de tais algoritmos, foi utilizado um banco de dados extensível, tanto em termos de tipos de dados, quanto em termos de operações. O banco de dados escolhido para tal foi o sistema SECONDO (GÜTING *et al.*, 2000; GÜTING *et al.*, 2005). Neste banco de dados implementamos novos tipos de dados para o armazenamento e recuperação das assinaturas geradas para os objetos. Além disso, implementamos operações para a geração e comparação das assinaturas. Mais detalhes sobre o sistema SECONDO foram discutidos na seção 3.

Os testes experimentais realizados e apresentados em detalhes na seção 5 demonstraram que, para comparações entre objetos do tipo polígono, esta nova proposta apresenta um desempenho semelhante ao desempenho da assinatura 4CRS (ZIMBRAO *et al.*, 1998), em termos de custo de CPU (tempo de execução) e número de acessos ao disco. Em relação ao custo de CPU, as assinaturas apresentaram uma redução de cerca de 70% sobre o tempo total. Sobre o número de acessos ao disco, a utilização das assinaturas representou uma redução de cerca de 80%. Além das vantagens apresentadas pela assinatura 4CRS, a representação 3CRS também apresentou um melhor desempenho para a geração da assinatura, com uma média de 30% de economia no tempo necessário para a geração da assinatura de quatro cores.

Outra vantagem da proposta é a sua flexibilidade, dado que pode ser utilizada para representar diferentes tipos de dados espaciais, tais como polígonos, polilinha e pontos. A

representação de polilinhas e pontos foi avaliada contra o processamento em dois passos nos testes experimentais, cujos resultados são apresentados na seção 5. Como resultado dos experimentos foi obtido um melhor desempenho, em termos de custos de CPU e número de acessos ao disco, nas junções espaciais para estes tipos de objetos. Junções envolvendo polilinhas necessitaram de um tempo de em média cerca de 50% do tempo de execução das junções em dois passos. A redução no número de acessos ao disco foi de cerca de 65%, em média. Para objetos do tipo pontos, a melhora apresentada nos testes experimentais para o tempo foi de cerca de 20%, enquanto a redução no número de acessos ao disco foi em média 6%. A Tabela 36 resume a melhor abordagem para as junções entre cada combinção de objetos espaciais. Para as junções polígono X polilinha, polígono X pontos e polilinha X polilinha, a arquitetura de 3 passos utilizando a assinatura 3CRS é a melhor abordagem. Para a junção entre polígonos, a arquitetura de 3 passos utilizando assinatura 4CRS é a melhor abordagem. Para a junção entre pontos, a arquitetura de 2 passos é a melhor abordagem.

Tabela 36. Melhor abordagem para os diferentes tipos de junção espacial

	Polígono	Polilinha	Pontos
Polígono	4CRS	3CRS	3CRS
Polilinha	3CRS	3CRS	3CRS
Pontos	3CRS	3CRS	2 passos

A operação que foi principalmente tratada neste trabalho foi a operação de junção espacial com operador de interseção. Entretanto, a assinatura 3CRS pode ser utilizada para outras operações. Como trabalhos futuros, sugerimos a utilização da representação proposta para reduzir o custo de CPU e o número de acessos ao disco na operação *diferente*. Para comparar dois polígonos a fim de identificar se são diferentes entre si, as assinaturas também podem ser analisadas: se possuírem resolução diferente, então a resposta de que os objetos são diferentes é imediata. Caso a resolução seja a mesma, então cada célula é comparada com a correspondente. Se pelo menos uma possuir uma cor diferente, então podemos afirmar que os objetos são diferentes. O contrário, entretanto, não é verdadeiro. Se todas as células possuírem a mesma cor que as suas correspondentes, então os objetos originais devem ser analisados, já que não é possível garantir que o conteúdo de duas

células *Inconclusivas* seja o mesmo. Esta operação traz um resultado exato. Outras operações, apresentadas em AZEVEDO *et al.* (2006) e AZEVEDO *et al.* (2005) para serem utilizadas com a assinatura 4CRS, trazem resultados aproximados, apresentando um determinado intervalo de confiança. Estas operações também podem ser adaptadas para a utilização com a assinatura 3CRS. Entre estas operações, destacam-se o cálculo da área aproximada de um polígono, a distância, o diâmetro, o perímetro e contorno, soma, diferença, o mais próximo, entre outros.

Outra possibilidade do uso da proposta a ser explorada em trabalho futuro é para objetos tridimensionais, ou ainda n-dimensionais. Para tal, as células bidimensionais seriam substituídas por cubos ou hipercubos. O algoritmo de geração das assinaturas deve ser alterado, mas o algoritmo de comparação é o mesmo utilizado para a representação bidimensional. Além da possibilidade da utilização em bancos de dados espaciais, as assinaturas tridimensionais podem ser utilizadas em aplicações que não necessitam de resultados exatos, mas exigem resultados rápidos, como ocorre em jogos. Este tipo de aplicação se beneficiaria da assinatura 3CRS em testes de colisão, que são freqüentemente utilizados.

Além dos trabalhos futuros propostos, outra implementação sugerida é um algoritmo para identificar o número de células ideal para representar o objeto. Uma idéia simples para ser utilizada para polígonos, derivada da proposta sugerida em AZEVEDO (2005), é iniciar a geração com apenas uma célula e a cada iteração aumentar o número até que o somatório das áreas das células *cheio* se aproxime da área do objeto original em um limite definido.

Bancos de dados espaciais freqüentemente realizam operações subseqüentes, onde o resultado de uma junção é utilizado como entrada da próxima, o que é chamado de multi-join. A capacidade da assinatura 3CRS de ser calculada sob demanda pode ser utilizada neste contexto. Para tal, a assinatura pode ser gerada sobre o resultado de cada junção para ser utilizada na junção seguinte.

Referências

ARONOFF, S., 1989, *Geographic information systems: a management perspective*, WDL Publications.

AZEVEDO, L.G., 2001, *Filtros Raster para Junções de Polilinhas*, Dissertação de Mestrado, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

AZEVEDO, L.G., 2005, *Processamento Aproximado de Consultas em Bancos de Dados Espaciais Usando Assinaturas Raster*, COPPE / PESC, UFRJ, Rio de Janeiro.

AZEVEDO, L.G., ZIMBRÃO, G., DE SOUZA, J.M., 2006, "Approximate Query Processing in Spatial Databases Using Raster Signatures". In: *VII Brazilian Symposium on GeoInformatics (GeoInfo 2006)*, Campos do Jordão.

AZEVEDO, L.G., ZIMBRAO, G., DE SOUZA, J.M., *et al.*, 2005, "Estimating the Overlapping Area of Polygon Join", *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*.

AZEVEDO, L. G., GÜTING, R. H., RODRIGUES, R. B., ZIMBRÃO, G., DE SOUZA, J. M., 2006, "Filtering with Raster Signatures", *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems, Arlington, Virginia, EUA*, p. 187-194

BATORY, D.S., BARNETT, J.R., GARZA, J.F., *et al.*, 1988, "GENESIS: An Extensible Database Management System", *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, v. 14, n. 11.

BOISSONNAT, J.D., PREPARATA, F.P., 1997, "Robust Plane Sweep for Intersecting Segments", *SIAM Journal on Computing*.

BRINKHOFF, T., KRIEGEL, H.P., 1994a, "Approximations for a Multi-Step Processing of Spatial Joins", *Proc. Int. Workshop on Advanced Research in Geographic Information Systems, Monte Verita, Ascona, Switzerland*, pp. 25-34.

BRINKHOFF, T., KRIEGEL, H.P., SCHNEIDER, R., 1993, "Comparison of approximations of complex objects used forapproximation-based query processing in spatial database systems", *Data Engineering, 1993. Proceedings. Ninth International Conference on*, pp. 40-49.

BRINKHOFF, T., KRIEGEL, H.P., SCHNEIDER, R., *et al.*, 1994b, "Multi-step processing of spatial joins", *ACM SIGMOD Record*, v. 23, n. 2, pp. 197-208.

CAREY, M.J., DEWITT, D.J., 1996, "Of Objects and Databases: A Decade of Turmoil", *VLDB*, v. 96, pp. 3–6.

CAREY, M.J., DEWITT, D.J., FRANK, D., *et al.*, 1986, "The architecture of the EXODUS extensible DBMS", *Proceedings on the 1986 international workshop on Object-oriented database systems*, pp. 52-65.

DUCZMAL, L., CANÇADO, A.L.F., TAKAHASHI, R.H.C., 2006, "Geographic Delineation of Disease Clusters through Multi-Objective Optimization". In: *GeoInfo - Simpósio brasileiro de geoinformática*, Campos do Jordão.

FREISEISEN, W., 1998, "A Generic Plane-Sweep for Intersecting Line Segments". In: Technical Report RISC-Linz TR-98-18, University of Linz, Linz, Austria

GORDON, S.R., GOODWIN, C.W.H., XIONG, D., 1994, "Draft Final Report on Status of Spatial/Map Databases", *Oak Ridge National Laboratory*.

GUTING, R.H., 1994, "An introduction to spatial database systems", *VLDB Journal*, v. 3, n. 4, pp. 357-399.

GÜTING, R.H., 1993, "Second-order signature: a tool for specifying data models, query processing, and optimization", *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pp. 277-286.

GÜTING, R.H., DE ALMEIDA, V.T., ANSORGE, D., *et al.*, 2005, "Secondo: An Extensible DBMS Platform for Research Prototyping and Teaching", *Proc. 21st Intl. Conf. on Data Engineering (ICDE)*, pp. 1115-1116.

GÜTING, R.H., DIEKER, S., 2000, "Plug and Play with Query Algebras: SECONDO. A Generic DBMS Development Environment", *Proc. IDEAS*, pp. 380-392.

HUDAK, P., 1989, "Conception, Evolution, and Application of Functional Programming Languages", *ACM Computing Surveys*, v. 21, n. 3.

IBGE, C.D., 2000, "Malha Municipal Digital do Brasil 1997", *IBGE–Cidades@ Acesso em*, v. 31, n. 03, pp. 2003.

KOTHURI, R.K., RAVADA, S., 2001, "Efficient Processing of Large Spatial Queries Using Interior Approximations", *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001)*, pp. 404–421.

MEDINA, N.D.L.O., LAPA, R., SANTOS, R.D., *et al.*, 2006, "Desenvolvimento de um SIG para reconfiguração de redes de energia elétrica com interface integrada". In: *GeoInfo - Simpósio brasileiro de geoinformática*, Campos do Jordão.

NEWMAN, W.M., SPROULL, R.F., 1979, "Principles of interactive computer graphics McGraw-Hill", *New York*.

OLSON, M.A., BOSTIC, K., SELTZER, M., 1999, "Berkeley DB". In: *Proc. of USENIX Technical Conference*, Monterey, CA.

ONG, J., FOGG, D., STONEBRAKER, M., 1983, "Implementation of data abstraction in the relational database system INGRES", *ACM SIGMOD Record*, v. 14, n. 1, pp. 1-14.

ORENSTEIN, J.A., 1986, "Spatial query processing in an object-oriented database system", *Proceedings of the 1986 ACM SIGMOD international conference on Management of data*, pp. 326-336.

PRESS, I., 1997, "Extending Informix Universal Server: Data Types", *Informix Software, Inc.*

RODRÍGUEZ-LUACES, M., 2000, "A Spatio-Temporal Algebra Implementation", *Proc. 5th World Conf. on Integrated Design and Process Technology.*

SAMET, H., 1990, *The design and analysis of spatial data structures*, Addison-Wesley Reading, Mass.

SCHWARZ, P., CHANG, W., FREYTAG, J.C., *et al.*, 1986, *Extensibility in the Starburst database system*, IEEE Computer Society Press Los Alamitos, CA, USA.

STONEBRAKER, M., ROWE, L.A., 1986, "The design of POSTGRES", *Proceedings of the 1986 ACM SIGMOD international conference on Management of data*, pp. 340-355.

TAO, Y., SUN, J., PAPADIAS, D., 2003, "Selectivity estimation for predictive spatio-temporal queries", *Data Engineering, 2003. Proceedings. 19th International Conference on*, pp. 417-428.

ZHU, H., SU, J., IBARRA, O.H., 2000, "Toward spatial joins for polygons", *Scientific and Statistical Database Management, 2000. Proceedings. 12th International Conference on*, pp. 231-244.

ZIMBRAO, G., DE SOUZA, J.M., 1998, "A Raster Approximation For Processing of Spatial Joins", *Proceedings of 24rd International Conference on Very Large Data Bases (VLDB'98)*, pp. 558–569.

ZIMBRÃO, G., SOUZA, J.M., MONTEIRO, R.S., *et al.*, 2000, "Filtro Raster para Junção de Polilinhas", *Anais do XV Simpósio Brasileiro de Banco de Dados-João Pessoa, PA, Brasil, Outubro*.