

Algoritmos exatos para a manipulação de mapas na TerraLib

Vinícius Lopes Rodrigues¹, Ângelo Sousa Cavalier¹, Marcus Vinícius Alvim Andrade¹, Gilberto Ribeiro de Queiroz²

¹Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa – MG – Brasil

²Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 515, 12201 São José dos Campos – SP – Brasil

{vlopes,angelo,marcus}@dpi.ufv.br, gribeiro@dpi.inpe.br

***Abstract.** The goal of this paper is to approach exact (roundoff error free) algorithms for geometric manipulation of geographical data. The main focus of this article is the map overlay algorithm, having TerraLib, a spatial component library with geometric algorithms support, as a development environment.*

***Resumo.** O objetivo deste trabalho é abordar algoritmos exatos (livre de erros de arredondamento) para a manipulação geométrica de dados geográficos. O principal foco deste artigo é o algoritmo de sobreposição de mapas, tendo como ambiente de desenvolvimento a TerraLib, uma biblioteca de componentes espaciais com suporte a algoritmos geométricos.*

1. Introdução

Sistemas de Informação Geográfica (SIG) são sistemas utilizados para representação computacional e análise de informações localizadas no espaço, possuindo aplicações em diversas áreas, como meio ambiente, saúde, gestão municipal e governamental [Câmara et al, 1996]. Inserido neste contexto encontra-se o projeto TerraLib, que consiste em uma biblioteca de código fonte aberto que possibilita o desenvolvimento de aplicativos geográficos customizados [Câmara, 2000].

Como qualquer SIG, o núcleo da TerraLib contém um módulo geométrico responsável por dar suporte às operações geométricas requisitadas pelas camadas superiores do SIG, como sobreposições e consultas. Neste módulo são definidos a representação geométrica e os diversos algoritmos que realizam as manipulações geométricas. Uma das grandes dificuldades deste módulo está no tratamento dos erros de arredondamento [Yap e Dubé, 1995; Wallner et al., 2000], pois os valores numéricos manipulados por tais algoritmos estão associados a uma representação topológica e as variações causadas pelos erros de arredondamento podem gerar inconsistências entre os valores e as respectivas representações.

Neste artigo apresentamos o trabalho que está sendo realizado para ampliar o conjunto de operações espaciais disponíveis na TerraLib, bem como o esforço empregado para torná-la cada vez mais robusta, evitando erros de arredondamento que possam levar a resultados inconsistentes, devido à propagação destes e à variação de decisão [Daltio et. al., 2004] baseada em informações suscetíveis a tais erros. No que

segue, apresentamos um algoritmo de sobreposição de mapas a ser incorporado a esta biblioteca, que realiza o tratamento de erros de arredondamento através do paradigma da computação exata [Yap e Dubé, 1995].

É importante ressaltar que a operação de sobreposição é uma das mais importantes para realização de análises espaciais, sendo utilizada em situações em que é necessário combinar ou comparar dados colocados em camadas de informações distintas. Por exemplo, considere-se uma consulta como “calcular a área de desmatamento ocorrido dentro de cada reserva indígena”. Para executar esta análise, é necessário combinar uma camada de objetos poligonais (os limites de reservas indígenas) com outra (o mapa de desmatamento), para se obter uma nova camada, a partir da qual é possível obter a análise desejada (área de desmatamento nas reservas indígenas). Esta operação recebe entradas de tamanho considerável, merecendo um estudo para que sua eficiência seja satisfatória.

Na implementação da operação de sobreposição será utilizado o algoritmo proposto por Kriegel et. al. (1991), pois ele tem a vantagem de processar as camadas a serem sobrepostas como um todo e não como polígonos individuais. Além disso, este algoritmo permite a combinação de várias camadas simultaneamente.

Este artigo encontra-se estruturado da seguinte forma: a seção 2 apresenta a representação adotada para as coordenadas dos elementos geométricos e a metodologia utilizada para os erros de arredondamento; a seção 3 apresenta uma revisão do algoritmo de varredura, que serve de base para o algoritmo de sobreposição apresentado na seção 4. A seção 5 apresenta uma discussão sobre os resultados esperados e o estado atual de desenvolvimento deste novo operador.

2. Metodologia para tratamento dos erros de arredondamento

Na versão atual da TerraLib, as primitivas geométricas foram construídas utilizando-se tolerâncias (ϵ) para tentar contornar os problemas dos erros de arredondamento. Uma vez definida esta tolerância, a igualdade entre dois valores ocorre se a diferença entre eles é menor que ϵ . Entretanto, esta estratégia não garante que os erros de arredondamento sejam tratados de forma efetiva.

Neste trabalho, para tentar contornar esses problemas, a metodologia utilizada foi representar os dados espaciais, como vértices e segmentos, utilizando coordenadas homogêneas [Andrade, 1999; Stolfi, 1991]. Resumidamente, em coordenadas homogêneas um ponto do plano é representado pela tripla $[x, y, w]$, sendo que se $w \neq 0$, então este ponto corresponde ao ponto de \mathbb{R}^2 com coordenadas cartesianas $(x/w, y/w)$. Se $w = 0$, o ponto corresponde ao ponto no infinito na direção do vetor (x, y) . Note que, para todo $\lambda \neq 0$, $[x, y, w]$ e $[\lambda x, \lambda y, \lambda w]$ representam o mesmo ponto.

Analogamente, uma reta é representada pelos seus coeficientes homogêneos $\langle A, B, C \rangle$ e, neste caso, ela consiste de todos os pontos $[x, y, w]$ tais que $Ax + By + Cw = 0$. Da mesma forma, para todo $\lambda \neq 0$, $\langle A, B, C \rangle$ e $\langle \lambda A, \lambda B, \lambda C \rangle$ representam a mesma reta. Esta representação permite decidir facilmente de que lado da reta está um determinado ponto. Dizemos então que a reta divide o plano em duas

partes: os pontos $[x, y, w]$ tais que $Ax + By + Cw > 0$ e os pontos $[x, y, w]$ tais que $Ax + By + Cw < 0$.

É importante ressaltar que os pontos cujos valores das coordenadas são números racionais podem ser representados com coordenadas inteiras na representação homogênea, sendo que a mesma observação também vale para as retas. Assim, as operações básicas entre esses elementos podem ser implementadas sem o uso de divisão, o que permite representação e cálculo exatos dos dados geométricos. A representação em coordenadas homogêneas também provê outras vantagens, como maior simplicidade das operações geométricas básicas.

O cálculo do ponto de interseção entre dois segmentos é um exemplo da maior simplicidade das operações geométricas em coordenadas homogêneas. Se dois segmentos se interceptam, este ponto de interseção corresponde ao ponto de interseção entre as retas suporte destes segmentos. Este ponto de interseção pode ser obtido facilmente da seguinte forma: supondo que os dois segmentos têm retas suporte com coeficientes $\langle r_x, r_y, r_w \rangle$ e $\langle s_x, s_y, s_w \rangle$ então as coordenadas do ponto de interseção são: $[r_x s_w - s_y r_w, r_w s_x - r_x s_w, r_x s_y - r_y s_x]$.

Para evitar os erros de arredondamento, os objetos serão representados utilizando coordenadas e coeficientes inteiros, ou seja, serão considerados apenas pontos e retas com coordenadas e coeficientes racionais. Desta forma, os pontos (que na base de dados estão representados em ponto flutuante) são convertidos utilizando um valor de precisão previamente estabelecido. Por exemplo, supondo que seja desejada uma precisão milimétrica, então um ponto com coordenadas cartesianas (12.478, 37.957) será convertido para [12478, 37957, 1000].

Assim, tendo os pontos representados por coordenadas inteiras e como todas as operações utilizam apenas operadores inteiros (+, -, *), então fica garantida a exatidão das operações.

No entanto, esta representação obriga a um cuidado especial com a possibilidade de ocorrências de *overflow* (quando um valor é maior que o máximo valor representável). Para tratar esta questão, foram desenvolvidas técnicas com intuito de evitar e contornar tais situações. Uma dessas técnicas foi implementar uma classe com operadores aritméticos básicos sobrecarregados que levantam uma exceção quando existe a possibilidade de ocorrência de um *overflow*. Assim, este fato pode ser tratado adequadamente pelas funções que utilizam essa classe. Esses tratamentos tentam diminuir os valores das coordenadas a valores que não geram *overflow*. Um método utilizado é a translação dos pontos para o menor valor de x dentre todas as abscissas e o menor valor de y dentre todas as ordenadas. Essa translação também é executada durante o pré-processamento dos algoritmos, tornando as coordenadas de entrada menores, diminuindo a probabilidade de *overflows*. Ao final da execução do algoritmo, os pontos são transladados de volta ao seu sistema de coordenadas original. Há outras formas de tratamento de *overflows* como, por exemplo, a bisseção dos segmentos na obtenção do ponto de interseção.

3. Interseção de segmentos com o paradigma da varredura

Uma implementação exata do algoritmo para obter a interseção entre um conjunto de segmentos foi elaborada por Daltio et. al. (2004) e é baseada no algoritmo de Bentley e Ottman (1979), que consiste em imaginar a existência de uma reta vertical r que varre o plano da esquerda para a direita, identificando e tratando alguns eventos. Cada um destes eventos serve de gatilho para que um conjunto de operações seja realizado. A reta r também contém as informações sobre os segmentos ativos, ou seja, os segmentos que são interceptados por r em um dado instante. Devido ao fato de os segmentos ativos serem armazenados em ordem crescente pelo valor de sua ordenada, o teste da interseção entre dois segmentos é necessário apenas entre dois segmentos ativos consecutivos.

Uma lista E é utilizada para armazenar os eventos, que são mantidos ordenados crescentemente pelo valor da abscissa. Na prática, a varredura consiste em percorrer a lista E e para cada elemento desta lista executar um conjunto de operações definido para cada evento. O algoritmo utiliza também uma lista L para indicar os segmentos ativos a cada instante. Uma alteração na posição relativa dos segmentos na lista L somente ocorre quando há um evento. Quando um evento extremo esquerdo é tratado, isso significa que o segmento correspondente àquele evento passa a estar ativo, enquanto no evento extremo direito, o segmento deixa a lista de segmentos ativos. Quando ocorre uma mudança de estado em L , o segmento que acarretou essa mudança deve sofrer um teste de interseção contra seus novos vizinhos. Caso encontre uma nova interseção, esta é reportada no resultado, além de ser inserida como um evento a ser tratado posteriormente. Quando um evento de interseção é tratado, os segmentos que o formam mudam de posição na ordem de L , e conseqüentemente são testados contra seus novos vizinhos. Ao término da agenda E , o algoritmo termina.

Uma importante característica desse algoritmo, que é em grande parte responsável por sua eficiência, é a manutenção dos segmentos ativos o que reduz consideravelmente os testes de interseção entre segmentos, pois restringe o número de segmentos a serem processados em um dado instante.

4. Sobreposição de mapas

Atualmente está sendo desenvolvida uma implementação para a operação de sobreposição, baseada no algoritmo de Kriegel et. al. (1991) que também segue o princípio da varredura utilizado pelo algoritmo de interseção de Bentley e Ottman.

Este algoritmo também contém uma lista dos segmentos ativos e uma lista de eventos, diferenciando apenas a complexidade de tratamento de eventos e a classificação destes, que agora é generalizado, ou seja, classificado em apenas um tipo. O evento é representado por um vértice e contém duas listas: a lista dos segmentos à sua esquerda e a lista dos segmentos à sua direita. Tais listas são utilizadas no tratamento dos eventos para associar os segmentos às regiões que estes delimitam.

Além disso, cada segmento possui um apontador para uma estrutura adicional, chamada de R , que é responsável por guardar os pontos relacionados às regiões associadas aos segmentos. Cada segmento ativo s possui basicamente dois apontadores para a estrutura R , $A(s)$ e $B(s)$, que representam as listas de pontos das regiões acima e

abaixo do segmento em questão, respectivamente, assim como o identificador da região delimitada por estes segmentos. A lista $A(s)$ de um segmento s é representada por um apontador para a cauda da lista de pontos que delimitam a região acima de s e a lista $B(s)$ é representada por um apontador pra cabeça da lista de pontos que formam a região abaixo de s .

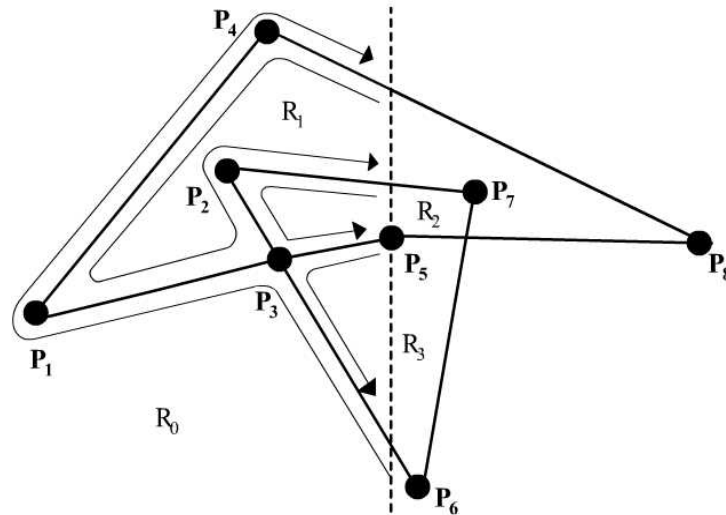


Figura 1. Exemplo de execução da sobreposição

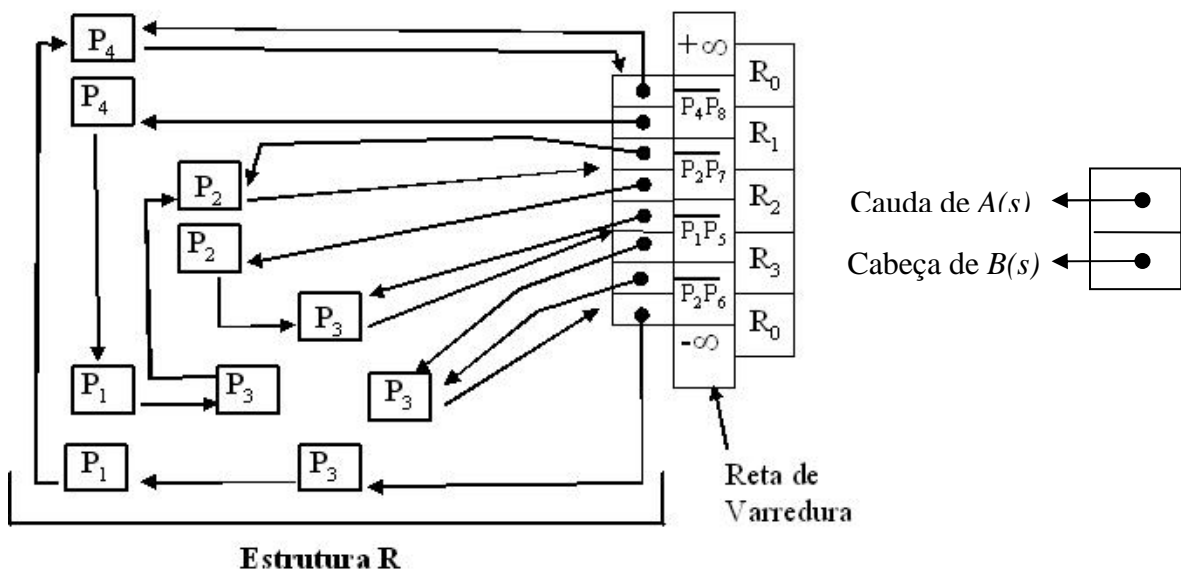


Figura 2. Representação do estado das estruturas envolvidas no algoritmo, considerando a execução mostrada na Figura 1.

A idéia principal do algoritmo é imaginar setas que circundam as regiões formadas pela sobreposição dos polígonos, como mostradas na Figura 1. Estas setas são uma maneira didática de se descrever as listas de pontos da estrutura R , mostrada na Figura 2. À medida que os eventos são tratados e os segmentos vão entrando e saindo da lista de segmentos ativos, essas listas vão sendo preenchidas e sendo atribuídas às regiões nomeadas entre dois segmentos ativos. Quando a região é fechada, ou seja, não

há possibilidade de nenhum segmento estar associado a ela, a região é reportada como parte do resultado final.

Observe que os possíveis erros de arredondamento ficam restritos ao processo de cálculo dos pontos de interseção entre os segmentos que formam as regiões. Portanto, para eliminar tais erros, este algoritmo utiliza uma abordagem semelhante à descrita na seção anterior, computando as interseções durante o tratamento dos eventos.

Atualmente, a TerraLib utiliza para operações de sobreposição de polígonos o algoritmo proposto por Margalit e Knott (1989), que descreve uma série de etapas para determinar a interseção, união e diferença de polígonos, inclusive para polígonos com buracos. Esse algoritmo, após um pré-processamento para orientar os polígonos adequadamente, calcula as interseções entre os polígonos, e assim é realizada a fragmentação das fronteiras. Esta fragmentação consiste em dividir o polígono em várias linhas que o formam, tomando referência para a divisão os pontos de interseção encontrados. De posse dos fragmentos de cada polígono, estes são classificados em relação à pertinência de cada fragmento ao polígono oposto. A partir dessa classificação, é decidido quais fragmentos farão parte da solução final.

As principais diferenças entre o algoritmo que está sendo implementado e a solução atualmente disponível na TerraLib são:

- A formação de regiões ocorre durante a computação dos pontos de interseção dos segmentos, dispensando operações de localização (como no caso da localização dos fragmentos), o que certamente levará a uma melhora considerável no desempenho global da operação de sobreposição;
- Estão sendo adotados métodos para tratamentos de erros de arredondamento, por exemplo, conforme os discutidos na Seção 2;
- Possibilidade de realização da sobreposição de mais de dois mapas “simultaneamente”, isto é, durante um mesmo processamento.

5. Considerações finais

No presente momento, os algoritmos descritos neste trabalho estão sendo desenvolvidos como parte de uma biblioteca acoplada à TerraLib, chamada GeoComp (Figura 3). Do ponto de vista do programador TerraLib, a interface desta biblioteca utiliza as estruturas do núcleo (kernel) da TerraLib, como o suporte de banco de dados, as estruturas de camadas de informação e temas e o modelo de geometrias. Internamente, ela possui rotinas para a conversão dos dados de coordenadas cartesianas para coordenadas homogêneas e vice-versa. Desta forma, quando alguma operação geométrica é requisitada, os dados manipulados pela TerraLib (em coordenadas cartesianas com ponto flutuante) são automaticamente convertidos para coordenadas homogêneas (exatas). Daí, são manipulados (de maneira exata) pelos respectivos algoritmos deste módulo e ao final, são convertidos para coordenadas cartesianas e retornados para os componentes da TerraLib que requisitaram a operação.

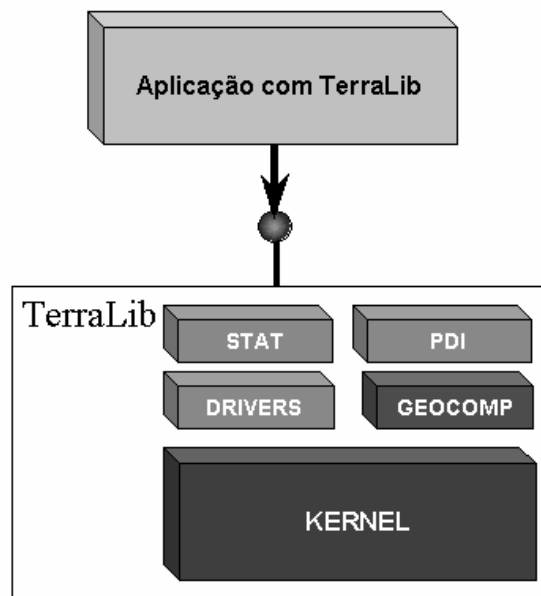


Figura 3. Organização da biblioteca GeoComp dentro da TerraLib

Nas próximas etapas do trabalho serão realizados testes práticos para avaliar o desempenho e a confiabilidade da operação de sobreposição de mapas com aritmética exata. Além de comparar o tempo para a realização das operações considerando as operações geométricas originais e a implementação exata, um outro importante objetivo dos testes é identificar situações em que os operadores originais geram resultados inconsistentes e que são corretamente obtidos pela implementação exata.

Um detalhe sobre a implementação do algoritmo de sobreposição é que a princípio ele tem como resultado todos os polígonos identificáveis pela sobreposição dos polígonos de entrada. Um trabalho futuro é elaborar uma classificação para as regiões encontradas, tornando possível gerar resultados de operações de interseção, união e diferença através da seleção das regiões adequadas para a saída de cada operação.

Além disso, outro objetivo é incorporar outras operações exatas à TerraLib. e também reescrever outras partes já existentes utilizando o paradigma de computação exata. Dessa forma, após a realização de testes e verificação da confiabilidade, o módulo GeoComp poderia ser integrado ao núcleo da TerraLib.

Bibliografia

- Andrade, M. V. A. (1999) “Representação e manipulação exatas de mapas esféricos”, PhD thesis, Instituto de Computação - UNICAMP.
- Bentley, J. L. and Ottmann, T. A. (1979) “Algorithms for reporting and counting geometric intersections”, In: IEEE Transactions on Computers, v. C-28, n. 9, p. 643-647.
- Câmara, G., Casanova, M. A., Hemerly, A. S., Magalhães, G. C. and Medeiros, C. M. B (1996) “Anatomia de sistemas de informação geográfica”, Campinas, Unicamp.

- Câmara, G., Souza, R. C. M., Pedrosa, B. M., Vinhas, L., Monteiro, A. M. V., Paiva, J. A. Carvalho, M. T. and Gatass, M. “TerraLib: technology in support of GIS innovation.” In: Anais, Workshop Brasileiro de Geoinformática, 2., 2000, São Paulo, São Paulo: INPE, 2000, p. 126-133.
- Daltio, J., Andrade, M. V. A., and Queiroz, G. R. (2004) “Tratamento de Erros de Arredondamento na Biblioteca TerraLib”, VI Simpósio Brasileiro de Geoinformática, Campos do Jordão, SP.
- Kriegel, H. P., Brinkhoff, T., and Schneider R. (1991) “An Efficient Map Overlay Algorithm Based on Spatial Access Methods and Computational Geometry”, Workshop Proceedings, Capri.
- Margalit, A. and Knott, G. D. (1989) “An algorithm for computing the union, intersection or difference of two polygons”, Computers & Graphics, v. 13, n. 2, p. 167-183.
- Nievergelt, J. and Preparata, F. P. (1982) “Plane Sweep Algorithms for Intersecting Geometric Figures”, ACM, v. 25, n. 10, p. 739-747, Out.
- Queiroz, G. R. (2003) “Algoritmos Geométricos para Bancos de Dados Geográficos: Da Teoria à Prática na TerraLib”, Tese de Mestrado, Inpe.
- Stolfi, J. (1991) “Oriented Projective Geometry - A framework for geometric computations”, Academic Press.
- Wallner, J., Krasauskas, R. and Pottmann, H. (2000) “Error Propagation In Geometric Computations”.
- Yap, C. and Dubé, T. (1995) “Exact computation paradigm”, World Scientific Press, pp. 452-492.