

# Criação e Manutenção de Subdivisões Espaciais

PAULO ROMA CAVALCANTI<sup>1</sup>  
PAULO CEZAR PINTO CARVALHO<sup>1</sup>  
LUIZ FERNANDO MARTHA<sup>2</sup>

<sup>1</sup>IMPA–Instituto de Matemática Pura e Aplicada  
Estrada Dona Castorina, 110  
22460 Rio de Janeiro, RJ, Brasil  
pcezar@visgraf.impa.br

<sup>2</sup>Departamento de Engenharia Civil e  
ICAD–Laboratório de CAD Inteligente  
Pontifícia Universidade Católica do Rio de Janeiro  
Rua Marquês de São Vicente, 225  
22453 Rio de Janeiro, RJ, Brasil  
lfm@icad.puc-rio.br

**Abstract.** This paper deals with the problem of creating and maintaining a spatial subdivision, defined by a set of surface patches. The main goal is creating a library of functions which provides a layer of abstraction capable of hiding the geometric and topological problems which occur when one creates and manipulates spatial subdivisions. The study of arbitrary spatial subdivisions extends and unifies the techniques used in nonmanifold solid modeling.

## Introdução

O presente trabalho trata do problema de criar e manter uma subdivisão do espaço tridimensional, definida por um conjunto de retalhos de superfície. O objetivo principal é a criação de uma biblioteca de funções capaz de criar uma camada de abstração que esconda os problemas topológicos e geométricos envolvidos na criação e manipulação de uma subdivisão arbitrária do espaço.

A motivação para este trabalho, que estende para o espaço tridimensional as técnicas para subdivisões planares apresentadas em [ROMA91], vem do fato de que as técnicas tradicionais de modelagem de sólidos são inadequadas para lidar com várias aplicações de interesse. Por exemplo, muitas vezes é necessário representar objetos constituídos por diversos materiais com diferentes propriedades (por exemplo: circuitos de semi-condutores, motores, estruturas de concreto e aviões) ou objetos com diversas regiões (por exemplo: malhas para modelos de elementos finitos). Sistemas tradicionais de modelagem têm também dificuldade para modelar relacionamentos de contato entre sólidos, cruciais em planejamento de montagens, robótica, geologia e em diversas outras áreas. O tratamento de subdivisões arbitrárias do espaço permite lidar, de uma maneira

uniforme e coerente, com todos estes problemas.

O estudo de métodos de representação e modelagem baseados em subdivisões arbitrárias do espaço engloba e unifica, de maneira natural, as chamadas técnicas de modelagem *nonmanifold* (isto é, a modelagem de objetos que não são necessariamente variedades do espaço tridimensional).

## Subdivisões Espaciais (SEs)

Um exemplo de uma subdivisão espacial simples está apresentada na fig. 1. Nesta subdivisão, o espaço é dividido em duas regiões limitadas ( $r_1, r_2$ ) e uma região ilimitada ( $r_{ext}$ ). Cada região é delimitada por um conjunto de cascas que, por sua vez, são formadas por um conjunto de faces (e/ou arames) encadeadas.

Um problema de interesse consiste em, dado um conjunto de retalhos de superfície (ou simplesmente retalhos), obter a SE determinada por estes retalhos. Um retalho é uma superfície algébrica<sup>(1)</sup>, orientável, conexa, sem singularidades e com bordo. Neste trabalho, descreve-se a estrutura do SSE (Sistema de Subdivisão Espacial), que é capaz de criar e manter uma subdivisão espacial, permitindo a inserção de novos retalhos em tempo real.

(1) Neste trabalho, são consideradas apenas superfícies descritas por variedades algébricas.

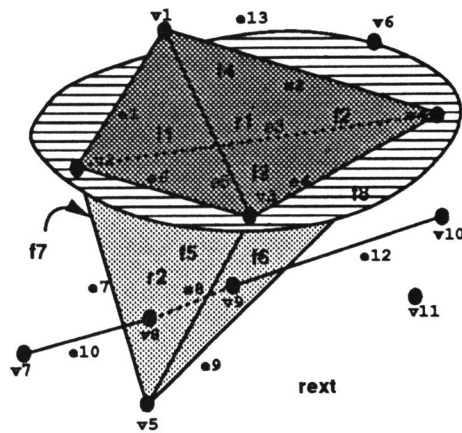


Figura 1 - Uma subdivisão do  $\mathbb{R}^3$ .

### Representação de uma SE

Embora uma descrição geométrica completa possa representar todas as informações sobre as formas geométricas dos elementos de uma SE e suas respectivas localizações no espaço, é aconselhável combinar, na representação, informações topológicas e geométricas. Por exemplo, a partir da descrição geométrica de dois retalhos de superfície, sua curva de interseção pode ser determinada sempre que necessário. No entanto, esta determinação requer um processamento que, em geral, é complicado. Assim, uma representação para SEs que contenha de forma explícita todas as interseções não só permite que os cálculos destas interseções sejam feitos uma única vez, no momento da sua criação, mas também evita a propagação de erros numéricos.

A idéia básica, portanto, é considerar que uma subdivisão espacial decompõe o espaço em um conjunto de porções disjuntas, denominadas células [ROMA92], cada uma delas homogênea em dimensão, e satisfazendo a condição de que a interseção das fronteiras de suas células é necessariamente igual à união de outras células da decomposição. Rossignac e O'Connor [ROSS90] descrevem um modelo matemático, denominado SGC (*Selective Geometric Complex*), que formaliza de modo apropriado este conceito. Representar a topologia de uma SE significa representar as informações de adjacência entre as suas células (isto é, informações sobre proximidade e ordenação de células).

Como vai ser visto adiante, a grande vantagem de armazenar explicitamente a topologia de uma SE em sua representação é que isto possibilita a construção de algoritmos geométricos muito mais eficientes. A nomenclatura utilizada para descrever a topologia de uma SE é a seguinte:

- Um **vértice** é uma célula, de dimensão 0, que corresponde a um único ponto do  $\mathbb{R}^3$ . Não po-

dem existir dois vértices com a mesma localização geométrica.

- Uma **aresta** é uma célula de dimensão 1 que corresponde a um segmento de curva homeomorfo a um segmento de reta. Um **arame** é uma aresta que não pertence a uma face. A fronteira de uma aresta é formada por dois vértices não necessariamente distintos. Estes dois vértices são ditos ligados.
- Um **ciclo** é um conjunto conexo e ordenado de vértices e arestas (alternadamente) na fronteira de uma face. Um ciclo pode consistir de um único vértice, sendo chamado, neste caso, de ciclo pontual.
- Uma **face** é uma célula de dimensão 2 que corresponde a uma porção conexa e limitada de uma superfície no  $\mathbb{R}^3$ . Sua fronteira é composta por um ou mais ciclos. Um dos ciclos contém os demais e representa a sua fronteira externa. Os outros ciclos (se existirem) são chamados anéis e representam fronteiras internas (furos).
- Duas faces  $S$  e  $Q$  de uma SE estão **encadeadas** quando existe uma seqüência de faces  $F_1, F_2, \dots, F_n$  na SE tais que  $F_1 = S$ ,  $F_n = Q$  e, para cada  $i = 1, 2, \dots, n-1$ , as fronteiras de  $F_i$  e  $F_{i+1}$  têm interseção não vazia. Se cada uma destas interseções contiver pelo menos uma aresta, diz-se que  $S$  e  $Q$  estão fortemente encadeadas.
- Uma **casca** é um subconjunto de células compondo uma componente conexa da fronteira de uma região. Ela pode ser formada por um conjunto de faces encadeadas, apenas por arames, ou, até mesmo, por um único vértice, sendo chamada, neste caso, de casca pontual.
- Uma **região** é uma célula de dimensão 3 que corresponde a um volume conexo do  $\mathbb{R}^3$ . Sua fronteira é formada por uma ou mais cascas. Uma das cascas contém as demais e representa a sua fronteira externa. As outras (se existirem) representam fronteiras internas (cavidades).

Situações onde mais de duas faces incidem na mesma aresta, ou vários volumes estão conectados por um único vértice, são comuns em uma SE. Por isso, uma representação para SEs deve ser capaz de armazenar de alguma forma este tipo de condição.

Diversos trabalhos apresentam métodos para representar subdivisões espaciais. Rossignac e O'Connor [ROSS90] abordam o problema geral de representar objetos  $n$ -dimensionais, possivelmente com estruturas interiores. Várias estruturas de dados, em geral apresentadas no contexto de modelagem sólida *nonmanifold* [WEIL86, DOBK87, LASZ87, LIEN88], se prestam a representar, de forma bastante geral, as informações de adjacência de objetos no  $\mathbb{R}^3$  que

não são necessariamente homogêneos em dimensão. Neste trabalho, utiliza-se a estrutura proposta em [WEIL86].

**A Estrutura de Dados RED**

Kevin Weiler [WEIL86] criou uma estrutura de dados capaz de armazenar os relacionamentos de adjacência de uma SE. Esta estrutura de dados é conhecida por estrutura *radial-edge* ou RED pelo fato de armazenar explicitamente a lista de faces, ordenadas radialmente em torno de uma aresta (fig. 2). Weiler criou a RED para ser utilizada em modelagem *nonmanifold* e provou que ela é completa, ou seja, permite que se extraia qualquer relacionamento de adjacência presente em uma SE.

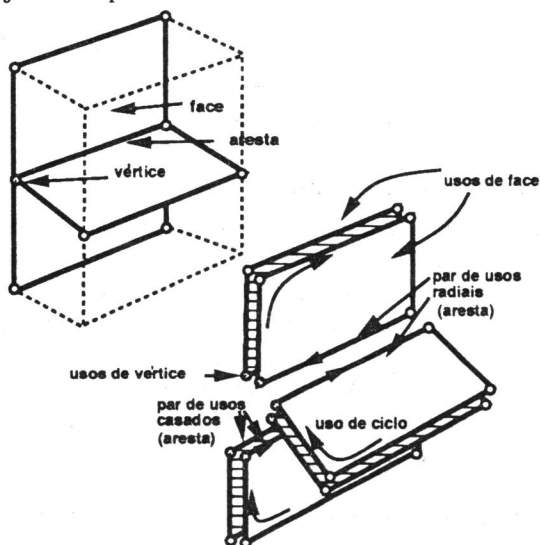


Figura 2 - Usos de elementos topológicos na RED.

A RED utiliza o conceito de *uso* de um elemento topológico para armazenar a topologia de uma SE. Um uso pode ser visto como a ocorrência de um elemento topológico em um relacionamento de adjacência com um elemento de dimensão superior. Assim, a RED armazena explicitamente os dois usos (lados) de uma face pelas duas regiões (não necessariamente distintas) que a compartilham. Cada uso de face é limitado por um ou mais usos de ciclo, que por sua vez são formados por uma seqüência alternada de usos de aresta e usos de vértice (fig. 2). Usos de vértice são utilizados para captarem condições *nonmanifold* nos vértices.

Os ciclos presentes no uso de uma face por uma região devem ser orientados de modo consistente em toda a estrutura, de modo a assegurar sua consistência global. Admite-se, neste trabalho, que o

ciclo externo de um uso de face seja sempre orientado no sentido horário quando observado da região a que se refere, enquanto ciclos internos são sempre orientados no sentido anti-horário (fig. 2). Com esta convenção, o “volume com sinal” [MANT88] de cada região limitada será sempre positivo. Este fato é explorado mais adiante para manter a consistência da estrutura quando a inclusão de uma nova face provoca a criação de uma nova região.

A RED apresenta uma descrição hierárquica de uma SE, partindo de níveis mais altos em dimensão (regiões) para os mais baixos (vértices) (fig. 3). Todos os elementos topológicos são mantidos em listas circulares duplamente encadeadas e possuem ponteiros para atributos.

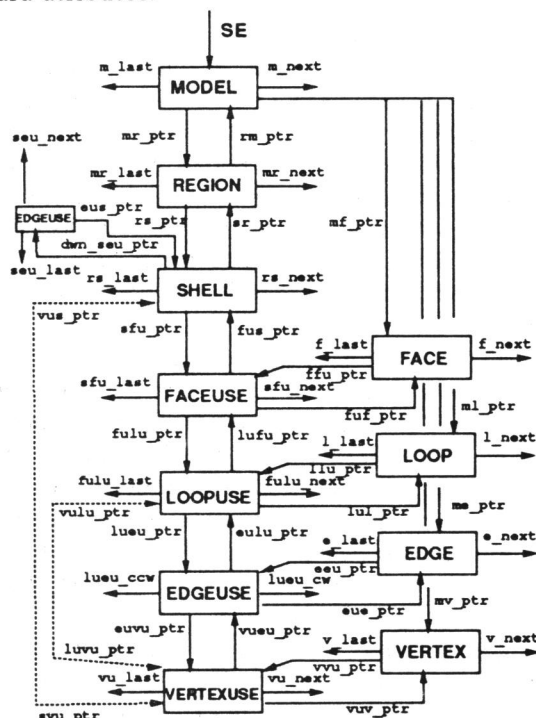


Figura 3 - Hierarquia da RED.

**Operadores *nonmanifold***

Estruturas de dados topológicas são bastante complexas e não devem ser manipuladas diretamente. Weiler [WEIL86] introduziu um conjunto de operadores, chamados neste trabalho WOps, que fornecem um meio, de relativo alto nível, para manipular a RED<sup>(2)</sup>. Estes operadores estão divididos em dois grupos. O primeiro possui operadores que agem exclusivamente sobre as faces de uma SE e são análogos aos apresentados em [ROMA91]. O segundo possui operadores que são capazes de criar arames e adicionar faces que são “costuradas” em arestas ou arames

(2) Embora atualmente já existam trabalhos que apresentam operadores *nonmanifold* de uma forma mais rigorosa (por exemplo, [MURA90]), ainda não existe um trabalho definitivo sobre o assunto.

especificados. Considerações a respeito de um conjunto mínimo de operadores podem ser encontradas em [TING90].

Os WOps do segundo grupo, e os seus respectivos inversos, são os seguintes:

<b>mmr</b>	( <i>make SE and region</i> )
<b>km</b>	( <i>kill SE</i> )
<b>msv</b>	( <i>make shell and vertex</i> )
<b>mev</b>	( <i>make wire edge and vertex</i> )
<b(kv)< b=""></b(kv)<>	( <i>kill vertex</i> )
<b>me</b>	( <i>make wire edge</i> )
<b>ke</b>	( <i>kill edge</i> )
<b>mf</b>	( <i>make face</i> )
<b>kf</b>	( <i>kill face</i> )
<b>esplit</b>	( <i>split edges</i> )
<b>join</b>	( <i>join edges</i> )

O WOp **mmr** cria uma SE com uma única região ilimitada. O WOp **msv** cria uma nova casca pontual em uma dada região. O WOp **mev** cria um arame e um vértice em uma dada região. O arame é delimitado por um vértice já existente e pelo vértice criado. O WOp **me** cria um arame entre dois vértices de uma dada região. O WOp **esplit** divide uma aresta (ou um arame) em duas, criando um novo vértice entre elas.

O WOp **mf** insere uma nova face  $f$ , na SE. Os vértices e arestas que formam a fronteira de  $f$  devem existir na SE. Através de um algoritmo de caminhamento, este operador detecta o fechamento de uma nova região. Por adjacência de arestas, são percorridos e marcados todos os usos de face que podem ser atingidos a partir de um dos dois usos da face criada. Ao final, se o outro uso não estiver marcado, uma nova região foi formada.

Um processo semelhante é usado para, caso seja formada uma nova região, distribuir os usos de face entre a região antiga e a nova. Se as faces que delimitam estas regiões não estiverem fortemente encadeadas (estando conectadas às suas cascas apenas por vértices ou arames), alguns usos de face simplesmente não são percorridos e, conseqüentemente, passam a não pertencer a uma região (fig. 4). Em função disto, foi adicionado um parâmetro de saída ao WOp especificado por Weiler. Este novo parâmetro é uma lista (possivelmente vazia) com os usos de face não classificados. Esta classificação deve ser feita, posteriormente, baseada em geometria.

O operador **mf** também foi estendido para ser capaz de criar faces com fronteiras desconexas (multi-ciclos). Este novo operador elimina as cascas que passaram a estar conectadas pela adição da nova face. Permite-se, também, que os ciclos contêm arestas pendentes.

Em cada WOp foi inserido um trecho de código

encarregado de chamar um procedimento que registra uma descrição do WOp e os respectivos parâmetros. Isto permite que possam ser executados algoritmos de *undo*, *redo* ou de inversão, utilizada no armazenamento permanente da SE, conforme visto adiante.

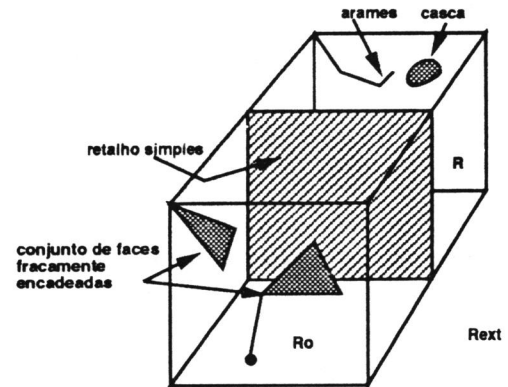


Figura 4 - Criação de uma nova região.

### Inserção de Retalhos

Neste trabalho considera-se a criação de SEs através da inclusão de um retalho de superfície por vez, atualizando-se a estrutura a cada inserção. A inserção de um novo retalho é, pois, a operação fundamental a ser executada pelo SSE.

Um retalho de superfície é uma superfície algébrica, orientável, sem singularidades, conexa e com bordo. Um procedimento para inserir retalhos em uma SE deve garantir que, após a inserção, a SE continue consistente topológica e geometricamente. A consistência topológica é mantida, naturalmente, se forem utilizados WOps de forma apropriada. Para isto, o retalho deve ser subdividido em um conjunto de retalhos inteiramente contidos em regiões da SE, chamados retalhos simples.

Para subdividir um retalho  $K$  é necessário descobrir que faces  $f_i$  da SE são cortadas e os segmentos de curva determinados por cada interseção (fig. 5).

Estes segmentos são usados para refinar  $K$  e cada  $f_i$ . Este refinamento pode ser feito inserindo cada segmento de curva nas faces apropriadas. Como o suporte de uma face é homeomorfo ao  $\mathbb{R}^2$ , o método descrito em [ROMA91], para inserir um segmento simples em uma face (plana), pode ser adaptado.

### Inserção de Retalhos Simples

Uma vez que se tenha subdividido um retalho e as faces por ele interceptadas, criando um conjunto de retalhos simples que encaixam na SE, cada um deles deve ser inserido na SE. O problema se reduz, então, em como inserir um retalho simples  $S$ . Para isso é necessário determinar:

- que vértices  $v_i$  e arestas  $a_i$  da SE serão “costurados” ao bordo de  $S$ .
- para cada  $a_i$ , encontrar a face que deve suceder  $S$  no ciclo ordenado de faces ao redor de  $a_i$ .
- que região  $R_o$  contém  $S$ .

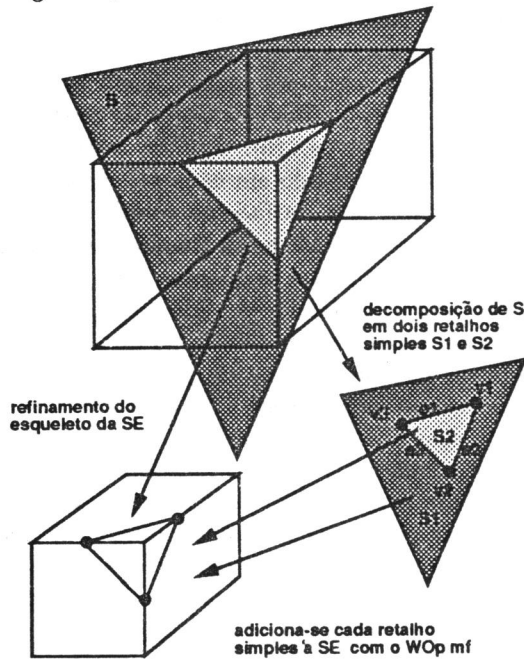


Figura 5 - Inserção de um retalho em uma SE.

O primeiro passo é encontrar os vértices da SE que coincidem geometricamente com vértices de  $S$ . Estes vértices são armazenados em uma lista  $V_l$ . O segundo passo é encontrar as arestas da SE que coincidem geometricamente com arestas de  $S$ . Somente arestas que ligam vértices armazenados em posições consecutivas de  $V_l$  precisam ser consideradas. Estas arestas são armazenadas em uma lista  $E_l$ .

Ao final deste processo, se  $E_l$  estiver vazia é porque  $S$  não está conectado a SE, ou está conectado a ela apenas por vértices. Em ambos os casos,  $R_o$  é a região que contém um ponto qualquer de uma aresta de  $S$ . Se  $E_l$  não estiver vazia, então, para cada aresta  $a_i$  armazenada em  $E_l$ , deve ser encontrada a face que sucede  $S$  no ciclo ordenado de faces ao redor de  $a_i$ . Quando as faces são planas, basta considerar cada vetor  $V(f)$  perpendicular a  $a_i$  e contido em uma face  $f$  incidente em  $a_i$ . A face  $f'$ , que sucede  $S$ , é aquela cujo  $V(f')$  forma ângulo orientado mínimo com  $V(S)$ . Esta face é armazenada em uma lista  $F_l$ . A face  $f''$ , que antecede  $S$ , é aquela cujo  $V(f'')$  forma ângulo orientado máximo com  $V(S)$ .  $R_o$  é a região delimitada por cada face que sucede e antecede  $S$  no ciclo ordenado de faces em torno de cada  $a_i$  (fig. 6).

Quando as faces não são planas, este método pode ser adaptado considerando  $V(f)$  perpendicular

ao vetor tangente à face  $f$  em um ponto qualquer de  $a_i$ . Isto funciona porque, se duas superfícies algébricas se interceptam ao longo de um trecho de uma aresta, a curva de interseção e a aresta têm suportes comuns.

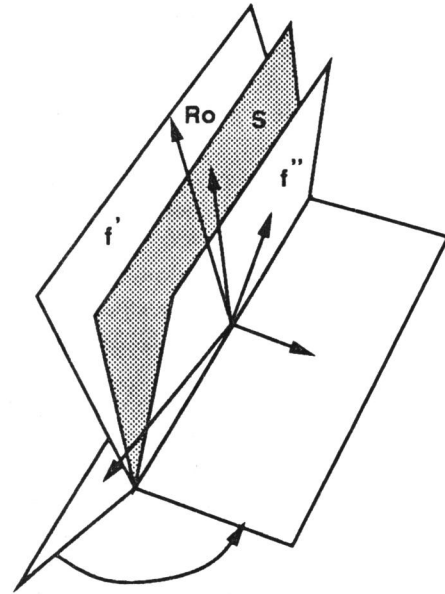


Figura 6 - Ordenação de faces ao redor de arestas.

O próximo passo para a inserção de  $S$  é adicionar cada  $v_i$  e cada  $a_i$  não presentes na SE, utilizando os WOps apropriados ( $me$  ou  $mev$ ), criando arames.

Finalmente, o retalho simples é inserido utilizando o operador  $mf$ .

### Criação de Uma Nova Região

A adição de  $S$  pode ocasionar a criação de uma nova região  $R$  na SE. Isto ocorre quando a inclusão do retalho  $S$  provoca a subdivisão de  $R_o$  e de uma de suas cascas (a casca de construção). É necessário, então, distribuir as diversas cascas de  $R_o$  entre  $R_o$  e  $R$ , de maneira a manter a consistência entre geometria e topologia.

Inicialmente, é necessário distinguir, entre as duas porções em que a casca de construção foi subdividida, qual delas é a casca que limita a nova região  $R$ . Para tal, escolhe-se arbitrariamente uma das cascas e calcula-se o “volume com sinal” da região que ela determina. Se o sinal for positivo, a escolha da casca foi correta. Senão, adota-se como nova casca para  $R$  a outra porção da casca de construção.

Além desta casca externa inicial,  $R$  pode ainda (fig. 4):

- possuir um conjunto de faces conectadas a ela apenas por vértices ou arames (e por isso não foram acrescentadas à sua casca externa).
- conter algumas cascas ou arames de  $R_o$ .



Para cada um destes casos, os seguintes procedimentos devem ser executados para corrigir a representação:

- verificar, para cada face de  $R_o$  que esteja conectada à casca de construção apenas por vértices ou arames, se um ponto qualquer de alguma de suas arestas está contido em  $R$ . Se estiver, colocá-la na casca externa de  $R$ . Caso contrário, deixá-la na porção da casca de construção que permanece em  $R_o$ .
- verificar se um ponto qualquer de cada arame da casca de construção está contido em  $R$ . Se estiver mover o arame para  $R$ .
- verificar se um vértice qualquer de cada casca de  $R_o$  (com exceção da sua casca externa e da casca de construção) está contido em  $R$ . Se estiver, movê-la para  $R$ .

O problema de testar a inclusão de um ponto  $p$  em uma região  $R$  é resolvido através de técnicas semelhantes às usadas para a versão bidimensional do problema [ROMA91]. Assim, pode-se empregar um algoritmo baseado na paridade do número de interseções de uma semi-reta partindo de  $p$  com as faces de  $R$  (com o cuidado de tratar as dificuldades introduzidas pela possibilidade de a semi-reta escolhida encontrar uma aresta ou um vértice na fronteira de  $R$ ). Pode-se, também, utilizar um algoritmo baseado na soma dos ângulos sólidos definidos por  $p$  e pelas faces de  $R$ . Verifica-se, em primeiro lugar, se  $p$  está na fronteira de  $R$ . Se não estiver, projeta-se cada face da fronteira na esfera de raio um, centrada em  $p$ , e calcula-se o somatório de todas as áreas destas projeções. Se o valor absoluto do resultado for  $4\pi$ ,  $p$  está no interior da região e se for nulo, está no exterior. Deve-se frisar que o algoritmo funciona mesmo que a região seja delimitada por várias cascas ou a sua fronteira não seja convexa. No entanto, apesar de sua elegância, o algoritmo pode apresentar problemas numéricos se  $p$  estiver muito próximo da fronteira de  $R$ .

### Interseção de Faces

O passo mais oneroso na inserção de um novo retalho é o cálculo das interseções do retalho com as faces já existentes, de modo a permitir sua decomposição em retalhos simples. A complexidade deste passo depende da geometria das faces. Embora o SSE seja concebido para lidar com superfícies de geometrias arbitrárias, em sua implementação atual ele lida apenas com faces planas e arestas retilíneas. A seguir, descrevemos resumidamente um algoritmo capaz de obter a interseção de duas faces planas.

Em linhas gerais, dadas duas faces planas  $A$  e  $B$ , cujas fronteiras são formadas por um ou mais ciclos, o

objetivo é determinar que vértices devem ser criados (em  $A$  e  $B$ ) e entre que pares de vértices devem ser criadas novas arestas, de forma a compatibilizar as faces (fig. 7).

O algoritmo percorre cada (uso de) aresta da fronteira de cada face (arestas pendentes são percorridas duas vezes, uma em cada sentido) e subdivide a reta  $L$  de interseção dos planos das faces em vários segmentos determinados por pares de pontos de interseção consecutivos. Em seguida, cada um destes segmentos é classificado em cada face como estando dentro da face, fora da face ou coincidente com alguma aresta. Com base nesta classificação, é possível determinar quais destes segmentos geram novas arestas em cada uma das faces. Por exemplo, para que um segmento da reta  $L$  determine uma aresta a ser criada em  $A$ , é preciso que ele seja interno a  $A$  e, ao mesmo tempo, seja interno a  $B$  ou esteja contido em uma de suas arestas (este é o caso, por exemplo, do segmento 2 da fig. 7, que corresponde a uma aresta a ser criada em  $A$ ). Critérios análogos existem para determinar quais dos pontos de interseção gerados sobre  $L$  correspondem a vértices a serem criados em cada face. Uma descrição detalhada do algoritmo pode ser encontrada em [ROMA 92].

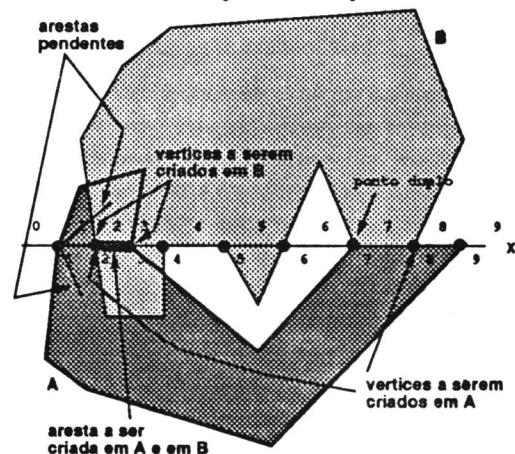


Figura 7 - Interseção de duas faces  $A$  e  $B$ .

O algoritmo acima supõe que as duas faces não sejam coplanares. Quando as duas faces estão contidas em um mesmo plano, a interseção não mais ocorre sobre uma reta. Neste caso, basta apenas criar, em cada face, vértices e arestas correspondentes aos trechos das arestas da outra face que lhe são interiores.

### Estimativa do Tempo de Criação de uma SE

A princípio, considere-se que a SE é criada a partir de retalhos simples que encaixam na SE. Isto ocorre, por exemplo, quando se tem a descrição geométrica

de cada face de uma SE, desejando-se apenas determinar sua estrutura topológica. Neste caso, deve-se verificar para cada novo retalho  $S$ , quais dos seus vértices já existem na SE. Supondo-se que  $S$  possui  $nv_s$  vértices e a SE possui  $nv_{se}$  vértices, então, na pior das hipóteses, se farão  $(nv_s * nv_{se})$  comparações. A identificação das arestas de  $S$  que coincidem com arestas da SE é mais simples. Consideram-se apenas vértices consecutivos de  $S$  que foram identificados com vértices da SE e, para cada um deles, verificam-se quais são as arestas da SE que incidem nos vértices identificados. Aquela que incidir em ambos os vértices é a aresta procurada. Esta busca é proporcional ao número de arestas incidentes nos vértices, que, tipicamente, não costuma ser um número muito grande. Feita a identificação das arestas, resta ainda determinar qual a região que contém  $S$ . Utilizando a informação de faces ao redor de arestas (ordenadas radialmente), esta determinação é proporcional ao número de faces ao redor de uma aresta, o que, também, não costuma ser um número elevado.

Note-se que, se os retalhos simples estão fortemente encadeados, pode-se executar este processo apenas para o primeiro retalho  $S_i$ . Os próximos retalhos devem ser obtidos por adjacência de aresta com  $S_i$ . Desta forma, já se tem antecipadamente uma aresta da SE coincidente com uma aresta de cada novo retalho. Isto permite que, para a identificação dos vértices, a busca seja feita apenas entre os vértices da região que contém cada novo retalho.

Se os retalhos não forem simples, a complexidade do algoritmo de criação de uma SE depende da complexidade dos algoritmos geométricos empregados para se determinar a interseção de duas faces. Esta complexidade, por sua vez, depende da geometria das faces. De todo o modo, o tempo de processamento dependerá do número de pares de faces cuja interseção deva ser examinada. Se  $F$  é o número final de faces da SE, é possível construir a SE efetuando-se  $O(F^2)$  interseções de faces. Claramente, esta não é a forma mais eficiente. Para melhorar o tempo de inserção, foi criado em [ROMA92] um algoritmo recursivo que explora o encadeamento das faces criadas ao se introduzir um novo retalho. O algoritmo resultante efetua apenas  $O(F)$  interseções de faces para cada casca fortemente encadeada da SE.

### Arquitetura de um Sistema de SE

A metodologia adotada na implementação do sistema de subdivisão espacial SSE é análoga à ado-

tada em [ROMA91] para o EDP (Editor de Diagramas Planares). Existem camadas de *software* com funcionalidades bem definidas: uma camada responsável pelo tratamento da topologia utilizando WOps (esta camada desconhece completamente que tipo de geometria está sendo utilizada); uma camada responsável pelo gerenciamento e manipulação de entidades geométricas (MGG); e por fim uma camada responsável pela interação com o usuário.

### Armazenamento de SEs

A estrutura RED é composta por porções de memória alocadas dinamicamente, recuperadas através de ponteiros. O armazenamento desta estrutura não é imediato. Uma forma elegante de efetuar tal operação é adaptando o algoritmo de inversão utilizado em [ROMA91]. O objetivo é desfazer completamente a SE, eliminando primeiro todas as suas faces, em seguida eliminando todos os seus arames e por fim todas as cascas pontuais<sup>(3)</sup>. Em disco são gravados, então, os WOps inversos daqueles utilizados na destruição da SE. A execução destes WOps na ordem inversa recria a SE. Deve-se notar, porém, que para isto é necessário salvar também os atributos geométricos e os atributos do usuário associados a cada entidade topológica.

Como os WOps inversos aos que destroem a SE devem ser executados na ordem inversa, é aconselhável armazená-los já de acordo com esta ordenação, ao invés de ler o arquivo de trás para frente. Para tal, é utilizada uma pilha. Cada rotina que implementa um WOp deve empilhar um registro com a descrição do WOp inverso, com seus respectivos parâmetros, sempre que estiver sendo executado o algoritmo de inversão.

### Funções Necessárias a um Sistema de SE

Uma das preocupações no desenvolvimento do SSE, já presente em [ROMA91], foi estruturá-lo de modo que, conceitualmente, ele seja capaz de lidar com quaisquer curvas e superfícies. Para tal, foi identificado um conjunto de funções, descritas abaixo, que devem ser fornecidas quando um novo tipo de geometria for considerado (funções 1-8).

- 1) Dado um par de células de mesma dimensão, verificar se elas são geometricamente idênticas.
- 2) Dadas duas faces, determinar se elas se interceptam e, em caso afirmativo, retornar uma lista com os vértices e arestas a serem criados em cada face.
- 3) Dada uma célula, retornar as coordenadas de um ponto no seu interior.

<sup>(3)</sup> Utilizando o WOp **kf**, pode-se reduzir qualquer SE a uma única região que contém apenas um conjunto de arames.

- 4) Dada uma célula e as coordenadas de um ponto, verificar se o ponto está no seu interior.
- 5) Dada uma célula, calcular uma medida com sinal da sua extensão. Para uma aresta, a medida é o seu comprimento; para uma face, a sua área; e para uma região, o seu volume.
- 6) Dada uma aresta  $a$  de uma SE e uma face  $F$  a ser inserida na SE, tendo  $a$  como uma de suas arestas, retornar a face que se segue a  $F$  no ciclo ordenado de faces ao redor de  $a$ .
- 7) Dado um par de células  $C_1$  e  $C_2$ , de mesma dimensão, que foram produzidas como resultado da divisão de uma célula  $C$ , distribuir os atributos geométricos de  $C$  entre  $C_1$  e  $C_2$ .
- 8) Dado um par de células  $C_1$  e  $C_2$ , com o mesmo suporte e uma célula  $c$  pertencente à interseção das fronteiras de  $C_1$  e  $C_2$ , combinar os atributos geométricos de  $C_1$  e  $C_2$  para formarem os atributos geométricos da célula resultante da eliminação de  $c$ .
- 9) Dada uma célula, alocar ou liberar a área de memória correspondente aos seus atributos.
- 10) Dado um arquivo e uma célula, gravar todos os seus atributos e uma indicação de quantos registros foram gravados.
- 11) Dado um arquivo e a indicação de quantos registros devem ser lidos, ler todos os atributos da célula e retornar um ponteiro para a área que os contém.
- 12) Dada uma célula, desenhá-la.

Algumas destas funções podem não ser facilmente implementáveis (por exemplo a 2), para certas geometrias, sendo necessária a utilização de métodos aproximativos. As funções 9-11 existem para permitir o armazenamento permanente e a recuperação de uma SE. A função 12 existe para que seja possível exibir graficamente uma SE.

### Aplicações e Conclusões

A principal contribuição deste trabalho é a apresentação de uma metodologia para lidar com subdivisões do espaço tridimensional. Tal metodologia permitiu a implementação do SSE, que é um sistema capaz de gerar uma descrição geométrica e topológica de uma subdivisão espacial a partir dos retalhos de superfície que a definem. O SSE consiste de uma biblioteca de funções que, chamada por um programa de aplicação, permite que o mesmo possa manipular subdivisões espaciais. Uma situação em que é necessário manipular SEs ocorre na criação interativa de malhas tridimensionais para elementos finitos. Esta técnica foi utilizada, por exemplo, em [MART89] para modelagem tridimensional de propagação de fraturas.

Uma outra aplicação das técnicas aqui apresentadas é na modelagem de objetos compostos por diversos materiais. A cada elemento topológico (região, face, aresta ou vértice) pode-se associar um atributo que representa o material correspondente. Para este propósito, o que se deseja é modelar agregados de objetos (sólidos possivelmente combinados com partes de dimensão inferior). No entanto, a forma de criação de uma SE aqui apresentada (isto é, através de retalhos de superfície) não é a mais natural. Requicha e Rossignac [ROSS91] descrevem um esquema de representação, chamado de CNRG — *Constructive Non Regularized Geometry*, que define operações de conjunto, análogas às operações CSG, para agregados de objetos. Em [ROMA92] mostra-se que os operadores CNRG podem ser implementados no SSE, de forma a obter-se um esquema que mantém uma representação explícita das células que compõem um objeto não homogêneo, permitindo ao mesmo tempo que ele possa ser criado de maneira construtiva, a partir de operações do tipo CSG. Esta técnica de modelagem, em que SEs mais complexas são obtidas a partir de SEs mais simples combinadas através de operadores CNRG, pode ser vista como uma extensão do uso de operadores booleanos em modelagem *manifold* [MANT88] ou *nonmanifold* [HOFF89, CROC91].

### Referências

- [CROC91] Crocker, G.A. & Reinke, W.F. - *An Editable Nonmanifold Boundary Representation*. IEEE Computer Graphics and Applications, Vol. 11, No. 2, March 1991.
- [DOBK87] Dobkins, D.P. & Laszlo, M.J. - *Primitives for the Manipulation of Three-dimensional Subdivisions*. Third ACM Symposium on Computational Geometry, Waterloo, Canada, Jun 1987, pp 86-99.
- [HOFF89] Hoffmann, C. - *Geometric and Solid Modeling: an Introduction*. Morgan & Kaufmann, 1989.
- [LASZ87] Laszlo, M.J. - *A Data Structure for Manipulating Three-dimensional Subdivisions*. Ph.D. Thesis, Report CS-TR-125-87, Dep. of Computer Science, Princeton University, Aug 1987.
- [LIEN88] Lienhardt, P. - *Extension of the Notion of Map and Subdivisions of a Three-dimensional Space*. Cinquieme Symposium sur les Aspects Théoriques de L'Informatique, STACS 88, Bordeaux, Feb 1988.
- [MANT88] Mäntyllä, M. - *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Maryland, 1988.



- [MART89] Martha, L.F. - *Topological and Geometrical Modeling Approach to Numerical Discretization and Arbitrary Fracture Simulation in Three Dimensions*. Ph.D. Thesis, Cornell University, Ithaca, N.Y., 1989.
- [MURA90] Murabata, S. & Higashi, M. - *Non-Manifold Geometric Modeling for Set Operations and Surface Operations*. Rensselaer Polytechnic Institute, Jun 1990.
- [ROMA91] Roma, P. Cavalcanti & P.C. Carvalho & Martha, L.F. - *Criação e Manutenção de Subdivisões Planares*, Anais do IV SIBGRAPI, 1991.
- [ROMA92] Roma, P. Cavalcanti - *Criação e Manutenção de Subdivisões do Espaço*, Tese de Doutorado, PUC-Rio, Dep. Informática, 1992.
- [ROSS90] Rossignac, Jarek R. & O'Connor, Michael A. - *SGC: A Dimension-independent Model for Pointsets with Internal Structures and Incomplete Boundaries*. Geometric Modeling for Product Engineering, North Holland 1990, pp 145-180.
- [ROSS91] Rossignac, Jarek R. & Requicha, A.G. - *Constructive Non-regularized Geometry*. Computer Aided Design, Vol.23, No.1, 1991, pp 21-32.
- [TING90] Ting, Wu Shin - *Considerations About a Minimal Set of Non-Manifold Operators*. Technische Hochschule Darmstadt - GRIS Wilhelminenstrabe, 7 DA-6100, FRG, Jun 1990.
- [WEIL86] Weiler, K. - *Topological Structures for Geometric Modeling*. Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, N.Y. Aug 1986.