

# Modelagem Baseada em Sistemas Físicos: Pesquisa de Vizinhança em Octrees para Implementação do Método Multipólos Rápido

**Artemis Moroni**

Fundação CTI - Centro Tecnológico para Informática  
CP 6162 - artemis@cti.ansp.br

**Léo Pini Magalhães**

UNICAMP - FEE - DCA  
CP 6101 - leopini@dca@fee.unicamp.br

**Cláudio L. Lucchesi**

UNICAMP - IMECC - DCC  
CP 6065 - lucchesi@dcc.unicamp.br

Abstract. This paper describes the algorithm for obtaining neighbour cells in a octree, used in the development and implementation of the PARTIC system. The PARTIC system is a 3-D implementation of the Fast Multipole Method for geometric modeling of isopotential surfaces using electrical particles systems.

## 1.0 Introdução

A visualização e simulação de processos físicos e a modelagem de fenômenos naturais tem ocupado um espaço crescente na Ciência da Computação. O computador tem sido usado para visualizar deformações em corpos sob tensão ou o comportamento de fluidos e outros materiais líquidos em situações específicas. Os métodos comuns da Computação Gráfica incluindo modelagem de sólidos e técnicas de "rendering" não são sempre adequados para tais aplicações especiais. Novas abordagens surgem agrupadas sob o termo *modelagem baseada em sistemas físicos*. Diferente da modelagem de sólidos, as primitivas usadas correspondem a entidades físicas discretas, como átomos, moléculas, estrelas ou galáxias. A idéia principal é descrever o comportamento de corpos complexos por partículas e regras simples associadas a estas partículas. Estas regras podem corresponder a leis de Newton para modelar o comportamento físico, mas podem descrever um comportamento social ou outros tipos de sistemas [Müller (1990)].

Este artigo descreve o algoritmo usado na implementação do Método Multipólos Rápido para modelagem de superfícies isopotenciais. Tal sistema foi desenvolvido na Fundação CTI - Centro Tecnológico para Informática, devendo incorporar-se ao sistema ProSim em desenvolvimento no Grupo de Computação de Imagens do DCA - FEE - UNICAMP, e com o apoio do DCC - IMECC - UNICAMP.

## 1.1 Sistemas de partículas

A representação através de sistemas de partículas difere em três aspectos básicos das representações normalmente usadas em síntese de imagens. Primeiro, um objeto é representado não por um conjunto de primitivas de

superfície, tais como polígonos ou "patches", que definem sua fronteira, mas como nuvens de primitivas de partículas que definem seu volume. Segundo, um sistema de partículas não é uma entidade estática. As partículas mudam de forma e movimento dinamicamente. Novas partículas "nascem" e outras "morrem" no decorrer do tempo. Terceiro, um objeto representado por uma partícula não é determinístico, desde que seu conteúdo e forma não estão completamente especificados.

Fenômenos tais como nuvens, fumaça, água e fogo quando modelados são denominados objetos "fuzzy". Estes objetos não têm superfícies lisas, bem definidas e brilhantes; pelo contrário, suas superfícies são irregulares, complexas e de baixa definição. Para a modelagem de tais objetos, a abordagem utilizando sistemas de partículas oferece algumas vantagens importantes, em relação às técnicas clássicas orientadas para superfície. Primeiro, uma partícula (um ponto no espaço tridimensional) é muito mais simples de ser gerada do que um polígono, a mais simples das representações de superfície. Uma segunda vantagem é que a definição do modelo é procedural, e é controlada por números randômicos. Portanto, obter um modelo altamente definido não necessariamente requer uma quantidade muito maior de esforço humano, como é frequentemente o caso de sistemas baseados em superfície. Também porque é procedural, um sistema de partículas pode ajustar seu nível de detalhamento para acomodar um conjunto específico de parâmetros de visualização. Da mesma forma que superfícies fractais, a ampliação de uma determinada região pode revelar mais e mais detalhes. E finalmente, sistemas de partículas modelam objetos vivos, i. e., que mudam de forma em um determinado período de tempo. É difícil representar dinâmicas assim complexas com técnicas de modelagem baseadas em superfícies [Reeves (1983)].

## 1.2 Objetos Definidos Por Funções Potenciais

Para aplicar um conjunto definido de regras a um sistema de partículas, as possíveis interações entre as partículas tem que ser determinadas. Como o número de partículas em um sistema em geral é muito grande, faz-se necessário um controle rápido de interação. O Método Multipólos Rápido [Greengard (1988)] sugere uma abordagem para acelerar os cálculos das interações de partículas para uma classe específica de problemas, onde as interações são expressas por funções potenciais.

## 2. O Método Multipólos Rápido (MMR)

A idéia básica em todas as abordagens para otimizar o algoritmo de complexidade  $N$  para cálculos potenciais é reduzir o número de objetos que participam do cálculo. Uma maneira é agrupando as partículas em buquês e calculando a interação de uma partícula com um buquê ao invés de com cada partícula individual. O potencial de um buquê de partículas pode ser expresso por uma função matemática. A qualquer momento, o potencial em uma posição do espaço com respeito a um certo grupo pode ser determinado avaliando-se a função correspondente. As funções de vários buquês distintos podem ser calculadas em uma posição específica e somadas para obter o potencial para um determinado ponto. Esta é uma maneira mais rápida de determinar o potencial do que o método à força bruta, desde que menos objetos estão envolvidos nos cálculos. O agrupamento de partículas, expressando o potencial desses grupos como expansões e usando uma estrutura de dados hierárquica para subdivisão da região, são as idéias fundamentais do algoritmo MMR. Esta abordagem resulta num algoritmo da ordem  $O(N)$  para cálculos no espaço.

A grosso modo, o algoritmo pode ser dividido em duas partes, o pré-cálculo e a avaliação.

### 2.1 O Pré-cálculo

No pré-cálculo o espaço computacional é subdividido de maneira a acomodar adequadamente os buquês de partículas. As partículas podem ser agrupadas levando-se em conta as propriedades das expansões para cálculos potenciais. Uma abordagem natural é agrupar as partículas com respeito à sua posição no espaço, o que requer um esquema de subdivisão adequado e estruturas de dados que reflitam esse esquema, como uma octree. Ainda, a estrutura hierárquica permite a escolha de um nível adequado de subdivisão adaptativamente ("branch-on-need octree").

### 2.2 A Avaliação

A. MORONI, L. P. MAGALHÃES, C. L. LUCCHESI

O próximo passo resulta no cálculo das expansões multipólos para cada nó (célula) na árvore, necessárias para a avaliação do potencial em cada região. A expansão multipólos nas células folhas deve ser determinada, descrevendo o efeito de todas as partículas contidas na célula na região exterior à célula. Esta região é o centro da célula mãe. Após o cálculo das expansões multipólos nas células folhas, elas devem ser transladadas para o centro de suas antecessoras e somadas. A seguir, cada célula no próximo nível de refinamento faz o mesmo com suas expansões multipólos, i. e., as células mães transladam suas expansões multipólos para suas antecessoras. Quando a raiz da árvore é alcançada, ela conterá a expansão multipólos descrevendo o efeito de todas as partículas armazenadas na árvore. A implementação da operação de translação para as células antecessoras usa o próprio mecanismo de retorno da árvore.

Após todas as expansões multipólos terem sido determinadas, as expansões locais representando o efeito das células não-vizinhas sobre uma dada célula na árvore são calculadas. Células não-vizinhas são as células de mesmo nível que estão à distância de uma célula da célula pesquisada. Pretende-se então avaliar o quanto o campo magnético das partículas fora da célula "influem" na célula. Para cada célula na árvore, as expansões multipólos de todas as células não-vizinhas são convertidas em uma expansão local e somadas. Esta expansão local é transladada para cada filha. As filhas convertem as expansões locais das células não-vizinhas em expansões locais, adicionando-as à expansão local recebida de sua mãe. Isto é feito recursivamente até que as folhas sejam encontradas. Como resultado, a expansão local de uma folha representa o efeito de todas as partículas das células não-vizinhas sobre aquela folha.

Finalmente, para avaliar o potencial em uma certa posição no espaço, a folha que contém aquela região deve ser encontrada. A expansão local deve ser calculada naquela posição e deve ser adicionada ao resultado do cálculo direto com as partículas das células vizinhas.

Ao final da avaliação tem-se calculado para cada célula o valor do campo potencial no centro da célula. Este valor será usado para a modelagem das superfícies isopotenciais. A descrição das equações usadas para a avaliação das expansões encontra-se descrita em [Müller, (1990)]. As equações utilizam funções de Legendre como soluções generalizadas da Equação de Laplace.

Para a avaliação do MMR, é importante que tanto a determinação do conjunto de células vizinhas e não-vizinhas bem como o seu acesso seja extremamente eficiente, pois para o cálculo das expansões locais, a partir do nível 2 na árvore (nível 0 = raiz), para cada célula são acessadas todas as células não-vizinhas, até o penúltimo nível na árvore. Ao nível das folhas, são necessárias as informações de todas as células vizinhas de cada célula.

A sessão a seguir trata do método implementado para a obtenção das células vizinhas e não-vizinhas.

### 3. Estruturas Hierárquicas Para Subdivisão Do Espaço

Octrees são estruturas de dados hierárquicas baseadas na decomposição recursiva do espaço em (normalmente) oito subvolumes, onde a raiz da octree refere-se ao volume completo. Octrees são uma generalização natural das quadrees e têm sido usadas em aplicações 3-D, sendo particularmente apropriadas para representar volumes em visualização científica, onde os pontos (coordenadas) frequentemente definem uma decomposição espacial em regiões cúbicas regulares ou irregulares, disjuntas, espaciais. Se os nós da octree forem usados para armazenar informações relativas a todos os subvolumes contidos, é possível explorar o volume sem se examinar cada coordenada.

Assim, a octree facilita a subdivisão do espaço adaptativamente. O elemento inicial de uma octree é um cubo, um subconjunto do espaço tridimensional. O cubo pode ser subdividido em oito cubos iguais recursivamente, até que um nível desejável de refinamento seja alcançado.

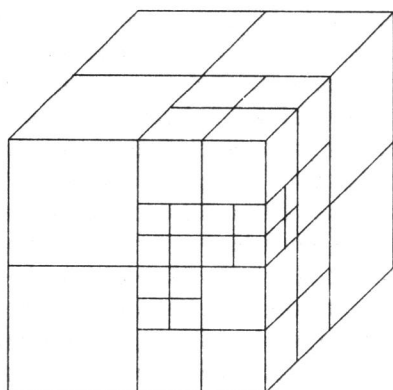


Fig. 3.0 - A Octree Adaptativa

Neste problema, a octree foi usada para representar o espaço onde estarão contidas as partículas elétricas. A raiz da árvore contém o espaço total e cada nó representa uma célula em um certo grau de refinamento, onde as folhas representam as células no mais alto grau. O critério para a subdivisão de uma célula no MMR é o número máximo de partículas em uma folha. Se o número de partículas em um nó é maior do que um valor pré-estabelecido, o nó é então subdividido em oito outros nós. Os nós não-terminais contém informações para o cálculo das expansões como as coordenadas polares do centro da célula, os valores das expansões multipólos e local nesta região, o número de partículas e listas das listas de partículas contidas nas células. Os nós-folha contém a lista e o número de partículas contidas na folha.

Os termos célula e nó serão usados indiscriminadamente neste texto para designar um nó genérico da árvore, sendo que o termo folha será usado para designar um nó terminal.

### 3.2 Relações de Vizinhança na Octree

As relações de vizinhança numa octree são um aspecto muito importante na avaliação do algoritmo MMR. Uma expansão multipólos pode ser calculada somente em uma posição suficientemente distante do centro da expansão, e a conversão de uma expansão multipólos em uma expansão local é válida apenas se as origens de ambas as expansões têm uma distância apropriada uma da outra. Isto pode ser expresso em termos de relações de vizinhança numa octree.

Define-se por vizinhas de uma célula (cubo) as células tendo uma face, aresta ou vértice em comum, no mesmo nível de refinamento. Uma célula não-vizinha é uma célula que não é vizinha de uma outra célula no mesmo nível. Desde que para o algoritmo MMR interessam apenas as células não-vizinhas que estão à distância de uma célula da célula pesquisada, define-se por células não-vizinhas apenas as células cujas mães (antecessoras) são vizinhas da mãe da célula pesquisada. Diz-se ainda que são células irmãs as células que têm a mesma antecessora.

A seguir, são mostradas numa quadtree as células (quadrados) vizinhas e não-vizinhas da célula A. Numa octree, as células são cubos nas posições correspondentes.

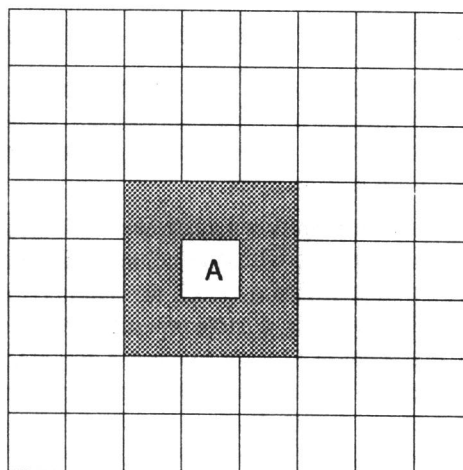


Fig. 3.1a - Células Vizinhas de A na Quadtree

No primeiro caso temos o quadrado Q1 de lado 3 circundando A, e um máximo de 8 vizinhas. No segundo caso, temos o quadrado Q2 de lado 5 circundando A, e um total de 16 vizinhas. Se denotarmos por S1 a área de Q1 e S2 a área de Q2, temos que  $NViz = S_1 - 1$ , onde NViz é o

número de células vizinhas e  $NNviz = S_2 - S_1$  onde  $NNviz$  é o número de células não-vizinhas.

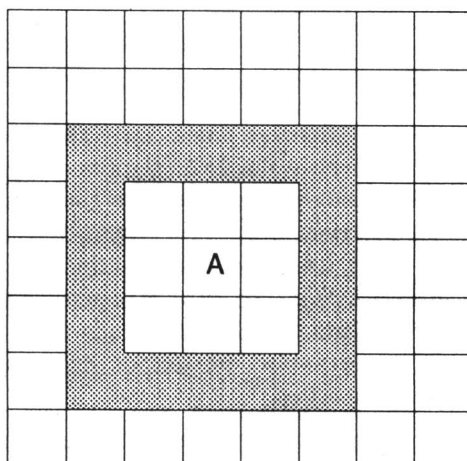


Fig. 3.1b - Células Não-Vizinhas de A na Quadtree

Analogamente, no caso da octree,  $NViz = V_1 - 1$  e  $NNviz = V_2 - V_1$ , onde  $V_1$  e  $V_2$  representam os volumes dos cubos de lados 3 e 5, respectivamente. Assim, numa octree cada célula tem um máximo de 26 vizinhas e 98 não-vizinhas e um mínimo de 0, caso seja uma célula de canto externo. Note que, devido ao elevado número de células não-vizinhas na octree é importante que a sua determinação e acesso seja extremamente eficiente, pois para a avaliação do MMR elas são calculadas para todas as células na octree.

### 3.3 Algoritmos de Pesquisa de Vizinhas

Os algoritmos existentes de pesquisa de vizinhas podem ser divididos em duas principais abordagens, bottom-up e top-down, supondo-se que não haja na árvore conexões entre células do mesmo nível.

Os algoritmos top-down atravessam a árvore da raiz até a célula, usando a trilha da célula para alcançá-la. Iniciando na raiz, a filha que está no caminho da célula procurada é selecionada, e isto é repetido até que a célula buscada seja atingida. O algoritmo tem a desvantagem de que o caminho até a célula deve ser conhecido. Para cada célula vizinha, a árvore deve ser atravessada a partir do topo novamente, o que pode ser bastante caro.

Uma abordagem diferente, o algoritmo de busca bottom-up, usa posições relativas ao invés de posições absolutas. A idéia básica é subir na octree até que uma antecessora comum seja encontrada, e então descer novamente em busca da célula vizinha. É óbvio que sempre se pode subir até a raiz da árvore e então iniciar a descida. De qualquer modo, o objetivo é encontrar a antecessora comum mais próxima, o que minimiza o número de nós que devem ser visitados. A partir daí, refaz-se o caminho usado para localizar esta célula, exceto que os movimen-

A. MORONI, L. P. MAGALHÃES, C. L. LUCCHESI são feitos "ao contrário", supondo-se um eixo imaginário passando pela célula antecessora.

A figura a seguir mostra a representação gráfica de uma quadtree. Cada célula (nó) numa quadtree corresponde a um quadrado no desenho.

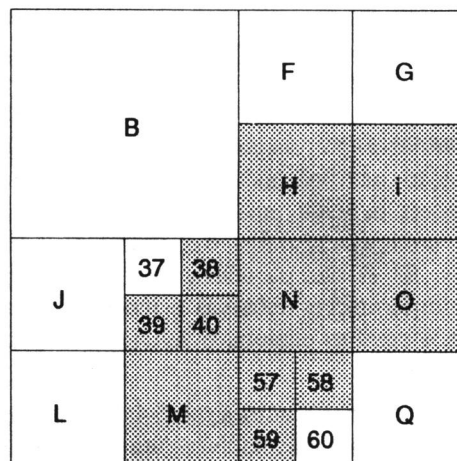


Fig. 3.2 - Representação Gráfica da Quadtree

Para orientação na quadtree, associa-se a cada canto da célula uma direção, conforme mostrado na Fig. 3.3. Assim, suponhamos que se queira encontrar a vizinha à Oeste da célula N na Fig. 3.2. A célula antecessora comum mais próxima é a primeira célula que é alcançada através de sua filha NE ou SE, ou seja, a primeira célula antecessora da qual N não é uma descendente à Oeste. A partir daí, o caminho é percorrido "ao contrário" em movimento descendente (em direção às folhas), tomando-se sempre as direções opostas das encontradas no movimento ascendente. No caso da vizinha à Oeste, as imagens reflexas de NO e SO são NE e SE, respectivamente. Então, a vizinha à direita da célula N na Fig. 3.2 é a célula K [Samet (1984)].

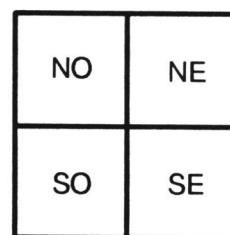


Fig. 3.3 - O esquema de Orientação Associado à Quadtree

Se a célula antecessora comum está próxima às folhas, o caminho de busca pode ser mais longo do que o caminho da raiz até a folha mais distante, como na abordagem top-down. Por outro lado, o caminho de busca é



muito mais curto quando a célula conectora está próxima da célula origem. A razão entre os caminhos mais longo e mais curto decrescem com a quantidade de filhos em um nó. Quanto mais filhos um nó pode ter, mais frequentemente pode ocorrer que o caminho de busca é mais curto que o caminho da raiz até o vizinho. De qualquer modo, a abordagem bottom-up é muito mais rápida do que a abordagem top-down, quando os vizinhos são requeridos para todos os nós [Müller (1990)].

### 3.4 A Octree vista como um 3-Cubo

Na literatura [Samet, 1989], em geral a pesquisa de vizinhança em octrees é tratada através de uma tabela de predicados que define relacionamentos tais como adjacente, oposto, face-em-comum, aresta-em-comum. Para a implementação do algoritmo MMR é muito importante que o cálculo e o acesso às células vizinhas seja extremamente eficiente. O método descrito a seguir permite a nível lógico a escolha do menor caminho (bottom up/top down) para o acesso a um nó vizinho, usando apenas operações de complementação booleana.

Na abordagem bottom-up para pesquisa de vizinhos foi mencionado o conceito de direção no exemplo sobre a quadtree (NE, NO, SE, SO). Na verdade, é necessário algum esquema de orientação para se percorrer a árvore, e nada mais natural do que usar os conceitos direita/esquerda na árvore binária ou uma analogia com os pontos cardeais no caso da quadtree.

No caso da octree isto torna-se muito mais complicado, pois sua representação espacial é um cubo. Assim, fica difícil associar um esquema de orientação intuitivo tal como os mencionados acima.

Se empregarmos um raciocínio análogo ao da quadtree à octree, e associarmos uma direção a cada vértice no cubo, obteremos um total de oito direções [Bell (1990)].

Buscando-se um esquema conhecido que permitisse organizar as direções na octree (cubo) de alguma maneira, optou-se por visualizá-la como um 3-cubo unário.

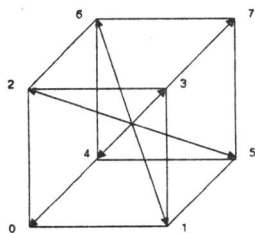


Fig. 3.4 - As direções no cubo

Por definição [Preparata], um 0-cubo unário é um vértice; um 1-cubo é um segmento; um 2-cubo é um quadrado e um 3-cubo é o próprio cubo. Para  $n = 4$ , a intuição humana falha com respeito à figura geométrica, mas é possível a representação de um  $n$ -cubo, para um  $n$  adequado, através de um grafo não-direcionado, cujos

vértices são os vértices do  $n$ -cubo e cujas arestas conectam os vértices que diferem em apenas uma coordenada (vértices adjacentes). Por exemplo,  $(1, 1, 0, 1)$  e  $(1, 0, 0, 1)$  são adjacentes porque eles diferem somente na coordenada  $x_3$  no conjunto de vértices  $V = \{ v \mid v = \{(x_4, x_3, x_2, x_1)\} \}$ . Os diagramas dos  $n$ -cubos para  $n = 1, 2, 3$  são mostrados na Fig. 3.5.

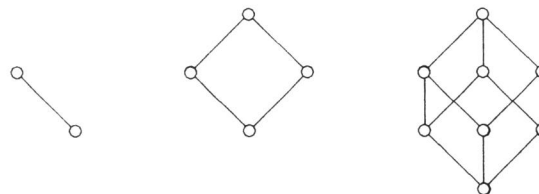


Fig. 3.5 - Os diagramas do 1-cubo, do 2-cubo e do 3-cubo

É exatamente a propriedade acima, ou seja, de dois vértices diferirem em uma única coordenada, que nos será útil na pesquisa de vizinhos ao esquematizarmos a octree como um 3-cubo.

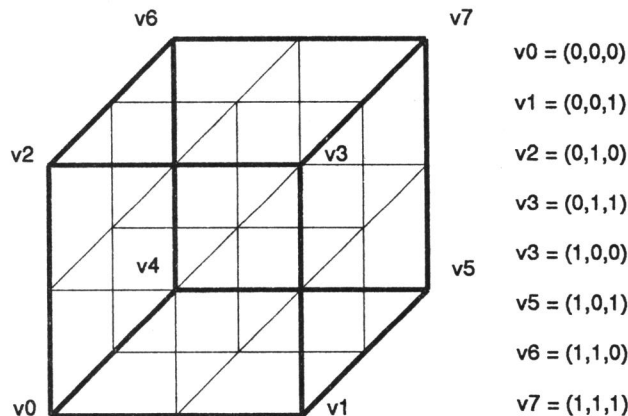


Fig. 3.6 - A octree vista como um 3-cubo

Na Fig. 3.6, é apresentado um cubo de lado 1, e vértices  $v_i$ ,  $i \in [0, 7]$ , e  $i$  é o octal correspondente à sequência de dígitos binários  $z_i y_i x_i$ . Os octantes da octree recebem a numeração do vértice em comum com o 3-cubo, e esse esquema é usado para orientação durante o percurso da octree, conforme mostrado na Fig. 3.7.

### 3.5 Algoritmos para Pesquisa de Vizinhos na Octree

A adoção de um esquema como o 3-cubo para a octree é bastante conveniente devido ao fato de que nesta representação os vértices adjacentes diferem em apenas uma coordenada. Baseado nesta propriedade, foram elaborados os algoritmos para obtenção dos vizinhos e não-vizinhos de uma dada célula.

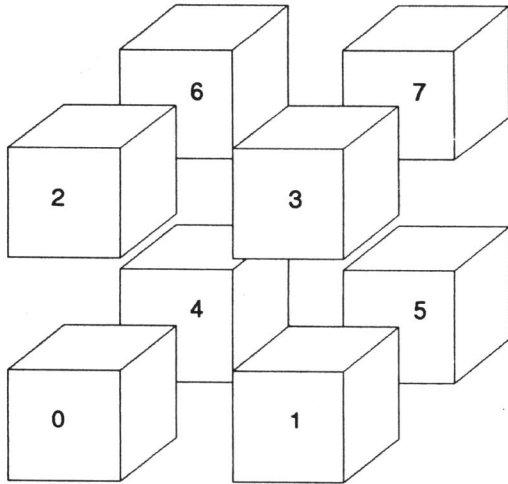


Fig. 3.7 - Octantes numerados de acordo com o 3-cubo

Na definição da estrutura de dados, a cada célula foi associada um campo *trilha*, que indica o caminho percorrido a partir da raiz para se atingir o nó. O campo trilha é um vetor onde cada elemento indica um octante e o índice o nível do octante. Assim, a sequência dos elementos do vetor indica os octantes percorridos até uma determinada célula. Por definição, a raiz está no nível 0.

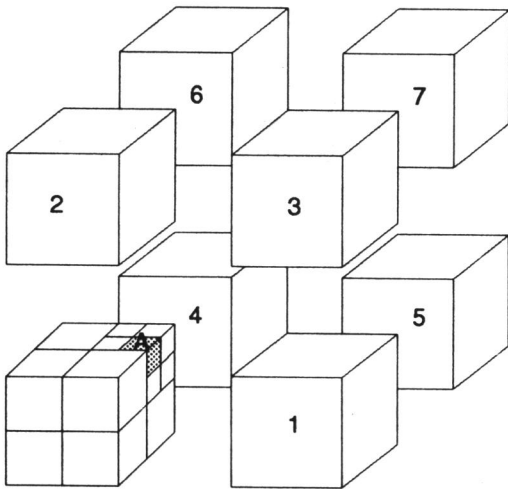


Fig. 3.8 - A célula A na Octree

No exemplo acima, a trilha associada à célula A é [073] pois o caminho da raiz até a célula é obtido tomando-se sucessivamente os octantes 0, 7, 3 dos octantes anteriores.

Serão apresentados a seguir os conceitos de células vizinhas de face, aresta e vértice. Por definição, todas as irmãs de uma célula são suas vizinhas, e os métodos descritos aplicam-se às células vizinhas não-irmãs. O objetivo é obter o cubo de lado 3 sugerido na seção 3.2,

onde a célula de trabalho é o cubo central, e todas que a circundam são suas vizinhas.

### 3.5.1 Vizinhas Imediatas

Por vizinhas imediatas definem-se as células que têm uma face em comum com uma dada célula e não são suas irmãs. Uma célula tem no mínimo zero vizinhos imediatos (no caso de ser uma célula de canto externo) e no máximo três. Tais células são também chamadas de vizinhas imediatas em relação a uma componente, x, y ou z.

A	z	y	x	z	y	x'	B
0	0	0	0	0	0	1	1
7	1	1	1	1	1	0	6
3	0	1	1	0	1	0	2

Fig. 3.9 - Decomposição das direções em x, y e z

Na Fig. 3.9, o caminho para se chegar da raiz à célula marcada é [073], o que indica a ordem em que os octantes foram tomados para se chegar à célula. Lembrando que a cada quadrante está associada uma direção, podemos decompor cada direção em suas componentes (z, y, x) conforme sugerido na seção 3.4.

Temos então que o caminho percorrido em relação à componente x é [011]. De maneira análoga à sugerida na seção 3.3, "escalamos" a árvore enquanto a componente x é diferente de 0, que é o oposto da direção através da qual a célula foi encontrada e que indica que chegamos a uma antecessora comum. A partir daí, "descemos" na árvore pelo caminho reflexo que é obtido invertendo os valores da componente x da direção e obtemos B = [162], que é a trilha da vizinha de face B da célula A em relação à componente x.

As células vizinhas imediatas são sempre obtidas através de uma das componentes da direção. Suponhamos que a célula em estudo seja a célula E = [033] e que se deseja calcular o vizinho imediato em relação à coordenada z.

E	z	y	x	z'	y	x	B
0	0	0	0	X	0	0	X
3	0	1	1	X	1	1	X
3	0	1	1	X	1	1	X

Fig. 3.10 - Decomposição das direções de E em x, y e z

Note que, se escalarmos a árvore a partir do nível 3 complementando os valores da coordenada z não obteremos o valor através do qual a célula foi encontrada, ou seja, 0 (Fig. 3.10). De fato, a célula não tem vizinho imediato em relação à componente z.

### 3.5.2 Vizinhas Relativas a Face

Na seção anterior vimos como obter uma célula vizinha imediata através de uma dada componente da direção. No exemplo, a célula B de trilha [162] foi obtida através da componente x da direção. Mas note na Fig. 3.12 que não apenas a célula B é vizinha de A, mas todas as células à esquerda na célula [16] são também vizinhas de A. São elas as células [160], [162], [164], [166]. Essas células são obtidas através de todas as combinações das coordenadas y e z no nível de refinamento a que pertence a célula, e são chamadas vizinhas relativas a face. No exemplo, o nível é 3.

B	z	y	x
1	0	0	1
6	1	1	0
c'	z'	y'	0

z'	y'	x	c'
0	0	0	0
0	0	0	2
0	1	0	4
0	1	0	6

Fig. 3.11 - O cálculo das vizinhas relativas a face

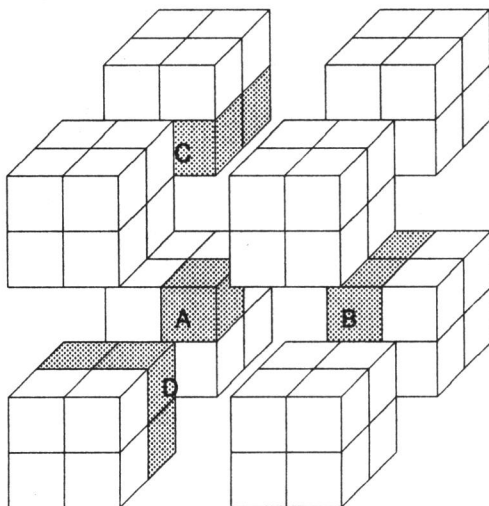


Fig. 3.12 - Vizinhas relativas a face da célula A

De modo análogo, são vizinhas imediatas as células C = [251] e D = [037], em relação às componentes y e z da direção; são vizinhas de face em relação à componente y as células [250], [251], [254], e [255] e em relação à componente z as células [034], [035], [036], [037].

As células B, C e D são ditas células imediatas de face. Define-se ainda, por vizinhas relativas a face, todas as células que podem ser obtidas através de uma face (Fig. 3.12). Uma célula tem no mínimo zero e no máximo doze vizinhas relativas a face.

### 3.5.3 Vizinhas Relativas a Aresta

De modo análogo, por vizinhas imediatas de aresta, define-se as células que têm uma aresta em comum com uma dada célula. Para se obter uma vizinha imediata de aresta, basta obter uma vizinha imediata de uma outra vizinha imediata, usando-se uma componente diferente da direção. Assim, no exemplo anterior, através da célula D = [037] que foi obtida pela componente z da célula A, obtemos por meio da componente y a célula E = [215], que é vizinha imediata de aresta da célula A. Note que poderíamos chegar à célula E através da célula C, usando a componente z. De qualquer maneira, para se chegar da célula A à célula E usam-se as componentes y e z da direção, em qualquer ordem.

Mas da mesma forma que no caso das vizinhas relativas a face, não apenas a célula D é vizinha da célula A, mas ambas as células inferiores posteriores do octante [21] também o são. Para se obter a trilha da outra célula, inverte-se no nível da célula (3) o valor da coordenada relativa à componente que não foi usada, no caso a componente x, obtendo-se a célula cuja trilha é [214].

São também vizinhas imediatas de aresta as células [126] e [340], bem como sua irmãs [124] e [344], obtidas através da complementação da componente da direção que não foi usada para se chegar à célula.

Define-se portanto vizinhas relativas a aresta todas as células obtidas através de uma vizinha de aresta. Uma célula tem no mínimo zero e no máximo seis células vizinhas relativas a aresta.

E	z	y	x
2	0	1	0
1	0	0	1
c'	z'	0	0

z'	y	x	c'
0	0	0	0
1	0	0	2

Fig. 3.13-Cálculo das viz. de aresta de A relativas a y e z

### 3.5.4 Vizinha Relativa a Vértice

A célula vizinha relativa a vértice é obtida através das componentes x, y e z da direção e é única quando existe. Dado que para se obter uma célula vizinha imediata de aresta usam-se duas componentes, basta portanto calcular

uma vizinha imediata de uma vizinha de aresta através da terceira componente. Note que, como as três componentes da direção são usadas no cálculo da vizinha de vértice, qualquer vizinha de aresta pode ser usada para obtê-la. No exemplo, a célula vizinha de aresta da célula A é a célula [304].

A célula vizinha de vértice tem em comum com a célula pesquisada um único vértice.

### 3.5.5 O Total de Vizinhas Obtidas

Os métodos descritos acima nos possibilitam obter 12 células vizinhas de face, 6 vizinhas de aresta e 1 célula vizinha de vértice, completando um total de 19 células vizinhas. Se adicionarmos a este número as 7 células irmãs, obteremos então as  $NViz = 26$  células vizinhas, conforme mencionado na seção 3.2. Desta forma, obtivemos o cubo vazado de lado 3 que envolve a célula pesquisada.

### 3.6 A Pesquisa de Não-Vizinhas na Octree

Vimos na seção anterior como obter as vizinhas de face, aresta e vértice. Mas no algoritmo MMR, não apenas as células vizinhas são importantes, como também as não-vizinhas, ou seja, as que estão à distância de uma célula da célula pesquisada.

Nessa pesquisa, o objetivo é obter as células externas do cubo de lado 5 mencionado na seção 3.2, que contém o cubo de lado 3 obtido com os métodos já descritos. Para tanto, são necessárias duas etapas, descritas a seguir.

#### 3.6.1 Obtenção das Células Não-Vizinhas Irmãs das Células Vizinhas

Esta etapa é imediata. Usando-se os métodos para obtenção de vizinhas de face, aresta e vértice, basta inserir na lista de células não-vizinhas as irmãs destas células que não são vizinhas, pois estão todas à distância de uma célula da célula pesquisada.

Dessa maneira, obtemos 26 células não-vizinhas, no caso da célula pesquisada ter todas as vizinhas, conforme mostrado na Fig. 3.14.

#### 3.6.2 Obtenção das Células Não-Vizinhas Através da Irmã Oposta

Define-se irmã oposta a célula irmã que tem um único vértice em comum com a célula pesquisada. Em outras palavras, está orientada em sentido oposto em relação às três componentes da direção no cubo que as contém. Portanto, todas as suas vizinhas, mostradas na Fig. 3.15, são não-vizinhas da célula pesquisada. Mas note no desenho da Fig. 3.16 que estas células não provêm todo o conjunto de células não-vizinhas esperado. Nosso ob-

*Anais do SIBGRAPI V, novembro de 1992*

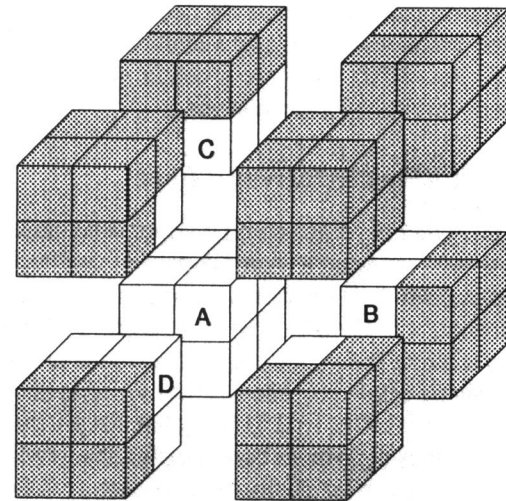


Fig. 3.14 - Células não-vizinhas irmãs das células vizinhas

jetivo é obter um cubo de lado 5, cujas células externas representam o conjunto das células não-vizinhas das células pesquisadas. Para tanto, é preciso calcular o conjunto de células adjacentes às células recém-obtidas.

No exemplo, a irmã oposta da célula  $A = [073]$  é a célula  $Z = [074]$ .

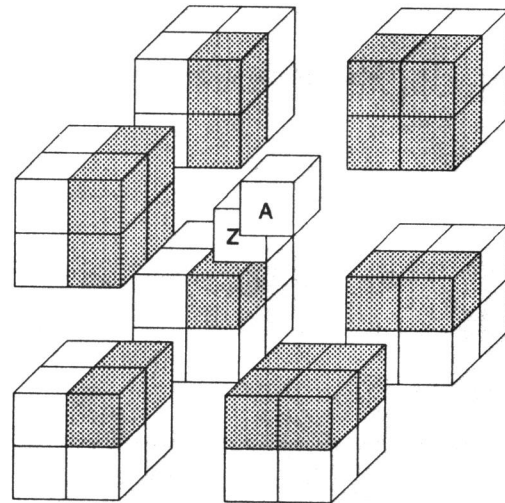


Fig. 3.15 - Células vizinhas da célula oposta

#### 3.6.2.1 Células Adjacentes

Acima, foram obtidas as células vizinhas da irmã oposta da célula A. Por células adjacentes definimos um conjunto de células que são adjacentes em relação à face, aresta e vértice às células mencionadas. Na verdade, são as vizinhas de um cubo maior, de dimensão 4, e os métodos para obtê-las são análogos aos já descritos.



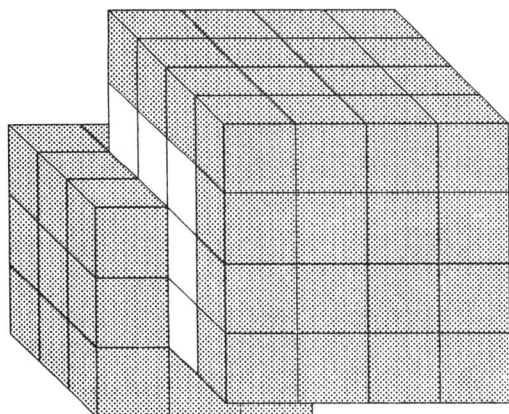


Fig. 3.16 - Células não-vizinhas já obtidas

Para se obter as células restantes, procede-se da seguinte maneira. Uma vez obtido um vizinho imediato, por exemplo a célula  $Y = [065]$  em relação à componente  $x$ , são calculadas as vizinhas de face  $[061]$ ,  $[063]$  e  $[067]$ . Note que sempre obteremos uma vizinha na mesma direção (3) da célula pesquisada, no caso a célula  $X = [063]$ . A partir desta célula calculamos as vizinhas imediatas de face em relação a  $y$  e a  $z$ , ou seja, em relação às componentes que não foram usadas no cálculo de  $Y$ , e a vizinha imediata de aresta em relação a  $y$  e  $z$ . Obtemos então as células  $[241]$ ,  $[027]$  e  $[205]$ .

Tomamos então todas as vizinhas de mesma face das mães das células encontradas, no caso as células  $[241]$ ,  $[243]$ ,  $[245]$ ,  $[247]$ ;  $[021]$ ,  $[023]$ ,  $[025]$ ,  $[027]$  e  $[201]$ ,  $[203]$ ,  $[205]$  e  $[207]$ . Assim, obtemos 16 células de face e adjacentes à célula irmã oposta. Procedendo da mesma forma em relação às outras faces, obtemos um total de 48 células não-vizinhas de  $A$ .

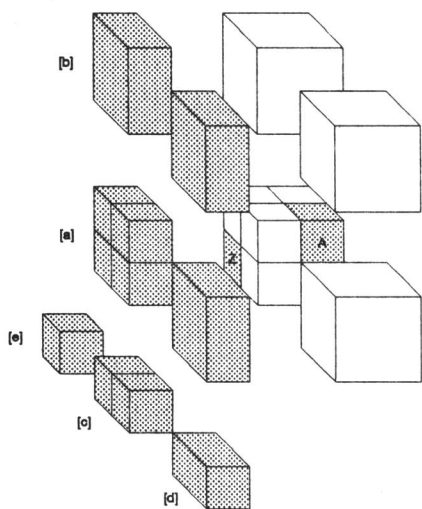


Fig. 3.17 - As vizinhas adjacentes

De modo análogo, calculamos as vizinhas de aresta da célula  $Z$  e as vizinhas adjacentes, obtendo 4 células por aresta e perfazendo um total de 12 células.

E finalmente, calculamos a vizinha de vértice da célula  $Z$ .

Desta forma, obtemos as 61 células não-vizinhas restantes, que somadas às 37 obtidas na etapa anterior, resultam no total esperado de 98 não-vizinhas. Estas células são também inseridas na lista de células não-vizinhas.

Obtemos então, o cubo de lado 5, cujas células externas são as células não-vizinhas da célula  $A$ .

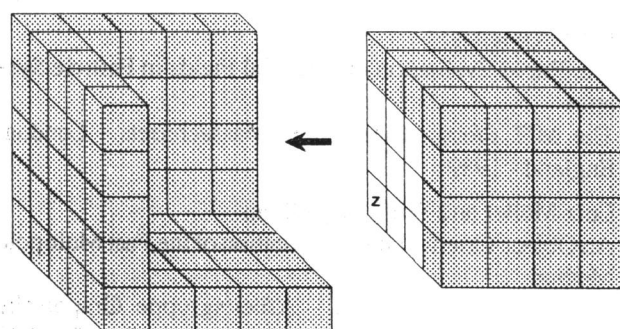


Fig. 3.18 - O conjunto de células não-vizinhas de  $A$

### 4.0 Conclusão

O sistema PARTIC foi desenvolvido na linguagem C, num computador VAX 785, e está sendo transportado para uma estação Sun, para avaliação de desempenho.

O sistema desenvolvido apresenta um avanço em relação ao descrito e implementado por [Müller (1990)] pois trata-se de uma implementação para o cálculo de superfícies isopotenciais tridimensionais, enquanto que as implementações anteriores eram bidimensionais. O algoritmo usado para navegação na octree permite a nível lógico a escolha do menor caminho (bottom up/top down) para se acessar um nó vizinho. Através da localização da antecessora comum, basta calcular e comparar o comprimento do caminho da raiz até o nó buscado, o que é dado pela profundidade do nó, com o dobro do comprimento do nó até a antecessora, antes de percorrer a árvore. A aplicação sucessiva do método permite obter uma célula a uma determinada distância. O algoritmo é extremamente eficiente pois os cálculos para a determinação da antecessora são efetuados através de operações booleanas.

### Agradecimentos

Agradeço a Marcos Antônio Rodrigues por suas valiosas sugestões durante o desenvolvimento deste trabalho.

Agradeço também a João Carlos Pinto, a Márcia C. C. Costa e a Jaime Khater por seu auxílio na elaboração deste artigo.

### **Bibliografia**

- S. B. M. Bell and D. C. Mason  
Tesseral Quaternions for the Octree  
The Computer Journal, Vol.33, No. 5, 1990
- H. H. Atkinson, I. Gargantini, and T. R. S. Walsh  
Filling by Quadrants or Octants  
Computer Vision, Graphics and Image Processing  
33, 138-155, 1986
- Leslie Greengard  
The Rapid Evaluation of Potential Fields in Particle Systems  
The MIT Press - Cambridge, Massachusets - London, England - 1989
- Lutz Müller and Wolfgang Müller  
An Object-Oriented Implementation of the Fast Multipole Method  
Fraunhofer Gesellschaft - Arbeitsgruppe Graphische Datenverarbeitung - Wilhelminenstr, 7 - D-6100 Darmstadt - West Germany Report No: TR-90030, 1990
- Hanan Samet  
The Quadtree and Related Hierarchical Data Structures  
Computing Surveys, Vol. 16, No. 2, June 1984
- Hanan Samet  
Neighbor Finding in Images Represented by Octrees  
Computer Vision, Graphics and Image Processing 46, 1989
- Preparata & Yeh  
Introduction to Discrete Structures  
Addison-Wesley Publishing Company, 1973