

## Otimização do Processo de Geração de Malhas Triangulares a partir de Polígonos

Prof. Dr. André Luiz Battaiola

Departamento de Computação / UFSCar  
Rod. Washington Luiz, km 235 - São Carlos - SP - 13565-905  
andre@dc.ufscar.br

Projeto AVVIC

Projeto ProTem-CNPq / Processo Institucional Número 680.063/95.1

**Abstract:** Triangle meshes represent a simple and an efficient form to describe 3D surfaces and many graphics libraries have functions to draw these meshes. Thus, as in terms of storage as in terms of performance of the exhibition process, it is important that these meshes are compact as much as possible. An efficient method to optimize these meshes is described in this paper.

**Resumo:** Malhas triangulares representam uma forma simples e eficiente de se descrever superfícies 3D e várias bibliotecas gráficas contém funções para a exibição destas malhas. Tanto em termos de armazenamento quanto em termos de performance do processo de exibição é importante que estas malhas sejam a mais compacta possível. Este artigo descreve um método eficiente de otimização destas malhas em relação a estes requisitos.

**Palavras Chaves:** Computação Gráfica, Visualização Científica e Volumétrica.

### 1. Introdução

O projeto AVVIC (Ambiente Virtual de Visualização Interativa Compartilhada) preve o uso conjugado de sistemas de visualização científica e de realidade virtual, de forma a possibilitar que as melhores características de cada um sejam aproveitadas em função do modo de uso. Por exemplo, os sistemas de visualização científica permitem de maneira mais apropriada a geração de gráficos científicos complexos com alta qualidade de exibição, enquanto os sistemas de realidade virtual permitem a navegação virtual.

O uso conjugado destes dois sistemas envolve a troca de objetos entre eles e a sua consequente manipulação. Neste texto, o objeto analisado é a isosuperfície, superfície que cruza uma grade tridimensional em pontos cujos valores são iguais a um valor específico denominado isovalor (fig. 1). Em visualização científica é comum o uso de isosuperfícies para a geração de gráficos complexos relacionados as áreas de medicina, de física, de meteorologia, etc.

Outro ponto a ser considerado em relação as duas áreas é um processo eficiente de animação. Neste caso, a taxa mínima de animação aceita é a de 8 quadros por segundo. Dois procedimentos simples normalmente seguidos para o aumento da velocidade de animação é armazenar a descrição dos objetos em memória e evitar repetidas inicializações (*overhead*) de funções de exibição. O escopo deste estudo é analisar um método que permita montar, armazenar e animar isosuperfícies.

### 2- Representação das Isosuperfícies

A partir de uma grade tridimensional de pontos, algoritmos como o dos cubos marchantes [LOREN87], [BATT92], interpolam uma malha poligonal ou triangular que representa uma isosuperfície (figs. 1 e 2). No caso dos cubos marchantes, uma configuração prévia é calculada para

os cubos da grade em função do valor 0 ou 1 associado aos oito vértices de cada cubo. O valor 0 significa ponto dentro da curva, dado que o valor deste ponto da grade é menor do que o isovalor. O valor 1 significa o oposto. Dado que cada cubo tem oito vértices e que é associado o valor 0 ou 1 a cada um deles, cada configuração de cubo é identificada por um número entre 0 a 255.

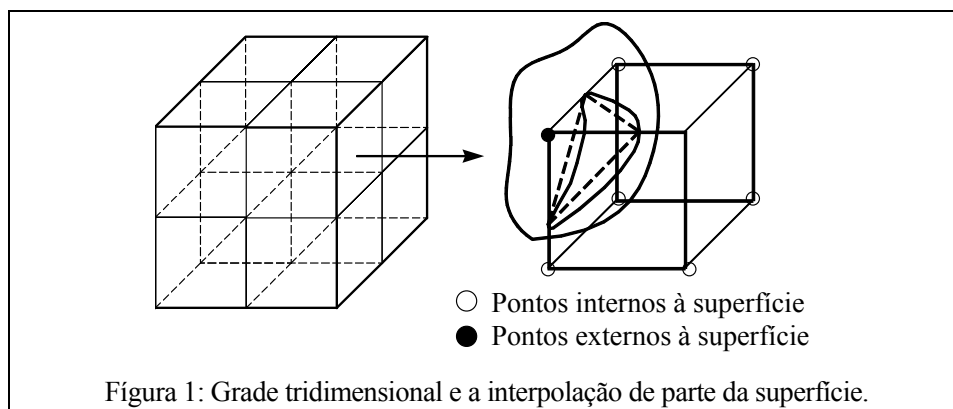
Isosuperfícies podem ser descritas através de malhas triangulares. Estas malhas são de grande interesse porque podem ser descritas de uma forma bastante eficiente [FLORI87], [FOLEY90]. No caso do exemplo na figura 2, 1, 2, 3, 4 e 5 são índices dos vértices dos triângulos, cujas coordenadas são conhecidas. A lista 1-2-3-4-5 indica que o triângulo A é definido pelos vértices 1-2-3, o B pelos vértices 2-3-4, o C pelos vértices 3-4-5, e assim sucessivamente no caso desta lista ser maior.

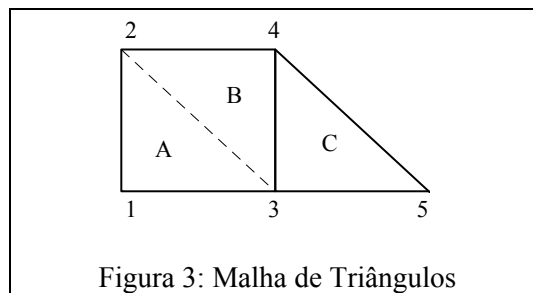
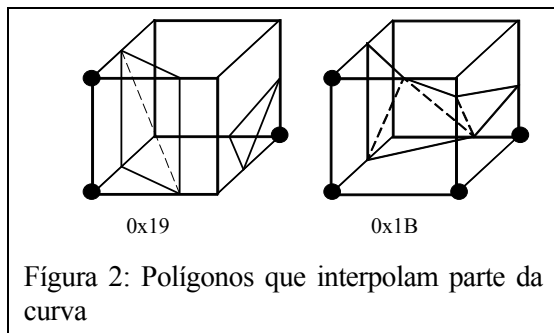
Para efeitos de animação de uma sequência de isosuperfícies relacionadas a diferentes instantes de tempo, é interessante armazená-las sequencialmente em memória (fig. 4). Cada superfície é composta de um vetor de índice de formação dos triângulos, dos vetores das coordenadas dos vértices e dos vetores das normais (fig. 5). Cada normal é associada ao vértice correspondente ao seu índice e elas são requeridas durante o processo de renderização.

Quanto menor as malhas menos espaço elas ocupam e conseqüentemente mais malhas podem ser armazenadas em memória. Quanto menor o número de malhas desconexas, menos vezes é chamada a rotina de exibição da biblioteca gráfica e, assim, menor o impacto do tempo requerido pelo processo de inicialização destas rotinas.

Além dos requisitos de eficiência do processo de exibição da malha, há outro relacionado com tempo gasto no processo de montagem da malha a partir dos polígonos determinados para cada cubo da grade. Isto implica que há um compromisso entre o tamanho da malha e o tempo gasto na sua montagem. O método descrito consegue balancear satisfatoriamente este compromisso.

Na otimização do processo de montagem das malhas considerou-se alguns mecanismos especiais de descrição das malhas triangulares que podem estar ou não presentes em diferentes bibliotecas gráficas. No método de otimização descrito, estes mecanismos estão incorporados, mas podem ser desabilitados no caso da biblioteca gráfica em questão não permití-los. Estes mecanismos estão descritos no item 3. A idéia do método é descrito no item 4 e medidas de sua eficiência no item 5. Conclusões estão no item 6 e a referência bibliográfica no item 7.





### 3. Mecanismos Especiais das Bibliotecas Gráficas

Os dois mecanismos especiais utilizados no método são a duplicação de triângulo e a linha de conexão (fig. 6). O primeiro mecanismo é mais fácil de ser suportado pelas bibliotecas gráficas, sendo o segundo usualmente menos aceito. Um simples teste permite verificar se a biblioteca passível de uso suporta estes mecanismos, sendo que bibliotecas como a XFDI [STELL90], a GL [GL90] e a OpenGL [OGL93a], [OGL93b], aceitam ambos. Note-se que o teste deve ser feito a partir da superfície renderizada.

A utilidade do primeiro mecanismo é permitir que se mude de direções durante o percurso pela malha. Assim, caso a malha de triângulos do objeto da figura 1 seja descrita pelos índices 1-3-2-4-3-5, o triângulo B é referenciado duas vezes, ou seja, 3-2-4 e 2-4-3, o que permite que se mude a direção de percurso e se defina o objeto através de uma única malha. Certamente, o objeto analisado não necessita deste recurso, pois pode ser definido através da malha 1-2-3-4-5. A descrição anterior só foi utilizada para exemplificar o uso do mecanismo.

O segundo mecanismo serve simplesmente para conectar malhas distintas no espaço, o que possibilita a descrição de uma superfície através de uma única malha.

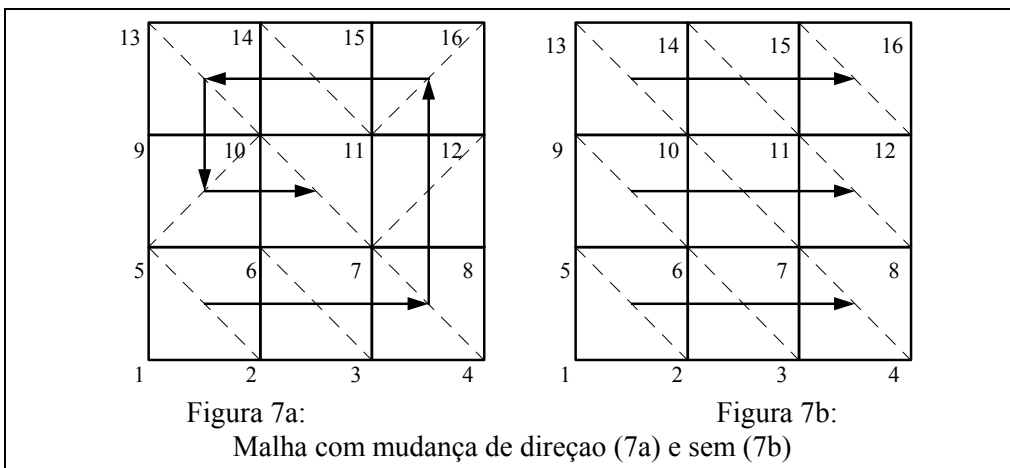
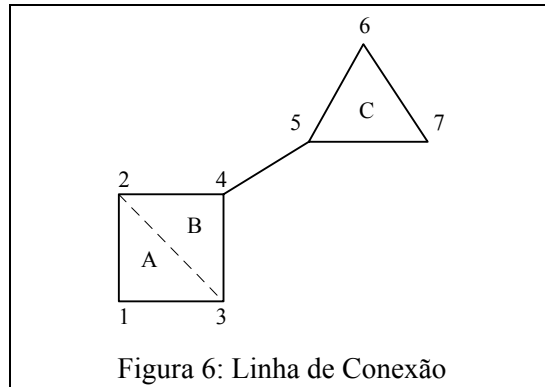
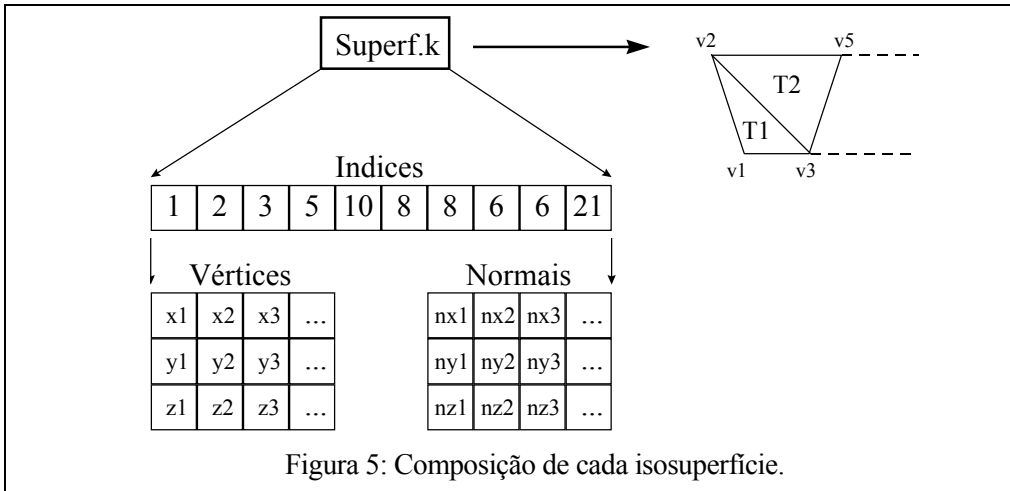
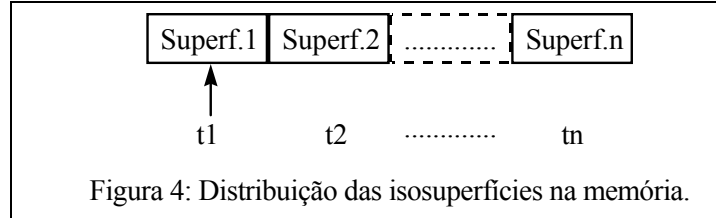
No caso de uma superfície plana descrita através de retângulos, a figura 7a apresenta a forma de descrição da malha desta superfície usando o primeiro mecanismo e a 7b a forma de descrição utilizando o segundo mecanismo. Note-se que, caso não seja possível o uso do mecanismo da linha de conexão, a definição da superfície segundo o esquema da figura 7a origina apenas uma malha, enquanto a definição pelo esquema da figura 7b origina três malhas. Como o interesse é analisar o uso destes mecanismos, será suposto que é possível o uso deste dois mecanismos. Assim, tem-se que as malhas das superfícies da figura 7 são dadas por:

7a: 1-5-2-6-3-7-4-8-7-12-11-16-15-11-14-10-13-9-10-5-6-10-7-11

7b: 1-5-2-6-3-7-4-8-8-5-5-9-6-10-7-11-8-12-12-9-9-13-10-14-11-15-12-16

Por este exemplo deduz-se que para o caso do mecanismo da mudança de direção (fig. 7a), o número de vértices da malha para uma superfície plana de dimensão 4x4 é 24. Para o caso de uma malha genérica “ $n_x \times n_y$ ”, o número de vértices da malha é determinado considerando-se inicialmente a descrição dos quadriláteros (células) da borda do plano. Assim, tomando-se como base o percurso sobre os quadriláteros da borda do plano na figura 7a e considerando-se que  $n_x \geq 4$  e  $n_y \geq 4$ , tem-se:

$$\begin{aligned}
 NV_{ct\_md}(n_x, n_y) &= (2 * n_x) + 1 + (2 * (n_y - 2)) + 1 + (2 * (n_x - 2)) + 1 + (2 * (n_y - 3)) \\
 &= 4n_x + 4n_y - 11
 \end{aligned}$$



Ressalte-se que  $NV_{ct\_md}(n_x, n_y)$  é o número de vértices da malha que descreve os quadriláteros da borda do plano na grade de dimensão  $n_x \times n_y$  usando o mecanismo da mudança de direção. No caso da grade ser quadrada, com dimensão “ $n \times n$ ”, tem-se:

$$\begin{aligned} NV_{ct\_md}(n) &= 8n - 11 \\ &= 2*(4n-8) + 5 \\ &= 1 + 2*(NQ_{ct}(n) + 2) \end{aligned}$$

$NQ_{ct}(n)$  é o número total de quadriláteros que formam a borda externa do plano. Considerando-se que no caso de uma borda interna, o ponto inicial já foi contado na borda anterior, tem-se:

$$NV_{ct\_md}(m) = 2*(NQ_{ct}(m) + 2)$$

Assim, o número total de vértices da malha,  $NTV_{md}(n)$ , no caso do uso do mecanismo de mudança de direção é dado por:

$$NTV_{md}(n) = 1 + NV_{ct}(n) + NV_{ct}(n-2) + \dots + NTV(k)$$

sendo  $NTV(k)$  o número total de vértices da malha mais interna da grade, considerando-se que o ponto inicial desta malha foi computado no número de vértices da malha da borda imediatamente externa a esta. O valor de “ $k$ ” é igual a “2” no caso de “ $n$ ” ser par e igual a “3” no caso de “ $n$ ” ser ímpar. Calculando-se a malha para “ $n$ ” igual a 2 e 3 tem-se:

$$\begin{aligned} NTV_{md}(3) &= 11 &&= 2*(NQ_{ct}(3) + 2) - 1 \\ NTV_{md}(2) &= 3 &&= 2*(NQ_{ct}(2) + 2) - 3 \end{aligned}$$

$NQ_{ct}(m)$  é o número dos quadriláteros de contorno de uma grade quadrada de dimensão “ $n \times n$ ”.  $NTQ(n)$  é o número total de quadriláteros desta grade. Logo no caso de “ $n$ ” ser par, tem-se:

$$\begin{aligned} NTV_{md}(n) &= NV_{ct}(n) + NV_{ct}(n-2) + \dots + NTV(2) \\ &= 1 + 2*(NQ_{ct}(n) + 2) + 2*(NQ_{ct}(n-2) + 2) + \dots + 2*(NQ_{ct}(2) + 2) - 3 \\ &= 2*(NQ_{ct}(n) + NQ_{ct}(n-2) + \dots + NQ_{ct}(2) + 2 + 2 + \dots + 2) - 2 \\ &= 2*NTQ(n) + (n/2)*4 - 2 \\ &= 2*(n-1)^2 + 2*n - 2 \\ &= 2*(n-1)*(n-1) + 2(n-1) \\ &= 2*(n-1)[(n-1) + 1] \\ &= 2*n*(n-1) \end{aligned}$$

No caso de “ $n$ ” ser ímpar, tem-se:

$$NTV_{md}(n) = NV_{ct}(n) + NV_{ct}(n-2) + \dots + NTV(3)$$

1

$$\begin{aligned}
&= 1 + 2*(NQ_{ct}(n) + 2) + 2*(NQ_{ct}(n-2) + 2) + \dots + 2*(NQ_{ct}(3) + 2) - \\
&= 2*[NQ_{ct}(n) + NQ_{ct}(n-2) + \dots + NQ_{ct}(3) + 2 + 2 + \dots + 2] \\
&= 2*[NTQ(n) + ((n-1)/2)*2] \\
&= 2*[(n-1)^2 + (n-1)] \\
&= 2*n*(n-1)
\end{aligned}$$

Assim:

$$NTV_{md}(n) = 2*n*(n-1) \quad n \text{ inteiro e } n \geq 2$$

Note-se que para “n = 4”, tem-se uma malha com 24 vértices, como exemplificado acima. Para “n = 50”, tem-se uma malha com 4900 vértices (tab. 1).

Para o caso de uso de linhas de conexão (fig. 7b), o número total de vértices da malha ( $NTV_{lc}(n_x, n_y)$ ) do plano em uma grade de dimensão “ $n_x \times n_y$ ”, é dado por:

$$NTV_{lc}(n_x, n_y) = (n_x - 1)*nv\_faixa\_h + nv\_lin\_conex$$

Sendo **nv\_faixa\_h** o número de vértices necessários para se definir os quadriláteros de uma faixa horizontal da grade e **nv\_lin\_conex** o número de vértices extras necessários para se compor a linha de conexão. Logo, para  $n_x$  e  $n_y \geq 3$ , tem-se:

$$\begin{aligned}
NTV_{lc}(n_x, n_y) &= (n_y - 1)*2*n_x + (n_y - 2)*2 \\
&= 2*(n_x*n_y - n_x + n_y - 2)
\end{aligned}$$

Se  $n_x$  igual a  $n_y$ , tem-se:

$$\begin{aligned}
NTV_{lc}(n) &= 2*(n^2 - 2) \quad n \geq 3 \\
&= 2*(n^2 - 1) - 2 \\
&= 2*(n+1)*(n-1) - 2
\end{aligned}$$

Assim, o número de vértices da malha do plano na grade 4x4 (fig. 7b) é 28. Para o caso de uma grade 50x50, o número de vértices é 4996.

Com base nestas fórmulas, tem-se que:

$$NTV_{md}(n) / NTV_{lc}(n) = [2*n*(n-1)] / [2*(n+1)*(n-1) - 2]$$

Para  $n \rightarrow \infty$  tem-se:

$$NTV_{md}(n) / NTV_{lc}(n) = 1$$

Logo, quanto maior a dimensão da grade, mais próximo o número de vértices obtido pelo método da mudança de direção fica do número de vértices obtido pelo método da linha de conexão. No entanto, o uso do método da mudança evita quebrar desnecessariamente a malha, o que é principalmente significativo quando o mecanismo da linha de conexão não pode ser utilizado. Além disto, a idéia é utilizar os dois

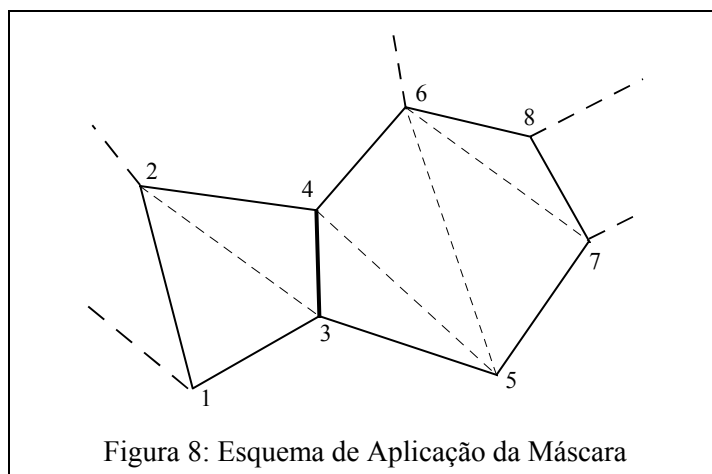
mecanismos simultaneamente. No item 5 são feitas outras considerações sobre a eficiência do processo em relação ao número de vértices da malha.

#### 4. Descrição do Método

A operação do algoritmo exige como dado de entrada a correlação entre vértices e polígonos, assim a tabela **Pol\_f\_Vert**, onde se tem a descrição dos polígonos em função dos vértices é suficiente. No entanto, para otimizar a performance do algoritmo, a tabela **Vert\_f\_Pol**, que relaciona vértices em função de polígonos, é utilizada para uma busca rápida dos polígonos que ainda não foram selecionados e também de seus vértices. Note-se que o algoritmo dos cubos marchantes, por exemplo, pode gerar estas tabelas.

Como tanto a velocidade de processamento quanto o tamanho da malha são requisitos para o método de geração das malhas, o uso de máscaras permite que, de forma simples, se satisfaça estes dois requisitos. A figura 8 ilustra o esquema de aplicação das máscaras. Por exemplo, considere-se que o percorrimento da malha tenha parado na aresta 3-4 do quadrilátero, ou seja, a malha está definida como ---1-2-3-4, e se deseja conectar a ela o hexágono. Neste caso, se aplica uma máscara sobre o hexágono, de tal forma que se define de imediato o caminho 4-5-6-7-8. Isto possibilita que se adicione a malha quatro triângulos de imediato e, conseqüentemente, se aumente a performance do processo de sua geração.

No caso de início de malha, a máscara pode ser utilizada para aumentar a chance de continuidade de percurso. Considerando-se os polígonos da figura 9, nota-se que várias malhas são passíveis de definição e, em particular, as malhas: 1-2-3-4-5-6-7 e 2-1-4-3-6-5-7. No entanto, como há um outro polígono conectado a aresta 7-8, a segunda definição não permite que se continue crescendo a malha através desta aresta. Para resolver este problema de forma eficiente basta apenas se examinar a aresta do segundo polígono que contém uma conexão e aplicar a máscara de forma inversa, conforme indicado pelas setas na figura 9.



Se o segundo polígono, pentágono na figura 9, conter várias arestas conectadas a outros polígonos, o algoritmo escolhe uma delas, sem se preocupar com a posterior continuidade da malha. Caso verificações posteriores ocorressem, o tempo de processamento iria aumentar e, como mencionado a seguir, o algoritmo estabelecido conseguiu um bom compromisso entre tamanho de malha e tempo de processamento.

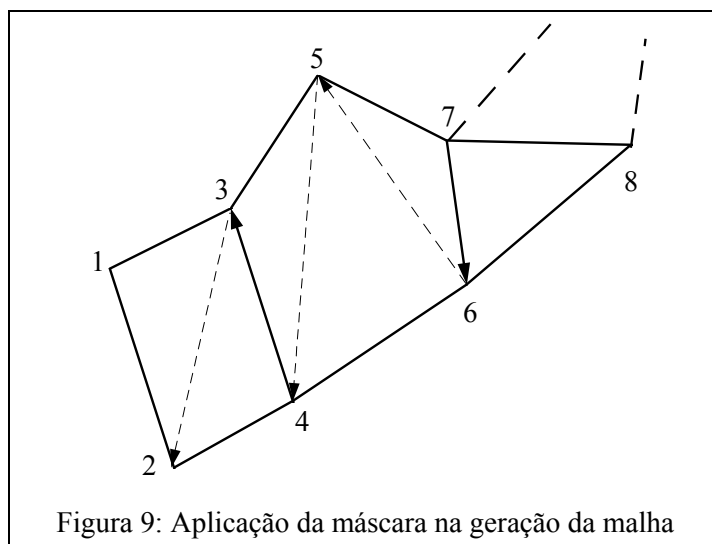
O uso da máscara neste processo implica em sua definição a priori. Para tanto, deve-se definir os polígonos a serem processados para uma aplicação em específico, por

exemplo, o algoritmo dos cubos marchantes trabalha com polígonos de 3, 4, 5 e 6 lados. Posteriormente, para cada polígono, fixa-se um vértice e se mapeia as 2 alternativas de percurso. Por exemplo, no caso do pentágono da figura 8, fixando-se o vértice 3, tem-se as seguintes opções de percurso: 3-4-5-6-7 e 3-5-4-7-6.

## 5. Medidas de Eficiência do Método

A eficiência do método foi considerada em termos da dimensão da malha gerada e do tempo gasto na sua geração, o que implica no balanceamento destes dois parâmetros.

A dimensão da malha pode ser diminuída quando se inicia a sua geração pelo vértice com o menor número de polígonos associados, dado que os outros vértices apresentarão um maior número de alternativas de movimentação. Observe-se que no caso do plano (fig. 7), a escolha do vértice 5 ao invés do 1, por exemplo, gera a sequência 5-1-6-2-7-3-8-4 para a faixa inferior do plano. A partir do vértice 4 não é possível se continuar a movimentação e uma linha de conexão tem que ser inserida para que se inicie o trajeto a partir da faixa imediatamente superior.



Para se precisar o impacto na performance do algoritmo da busca do vértice associado ao menor número de polígonos, foi medido o tempo de geração da malha para uma grade que contém um plano (tab. 1) e para outra que contém três esferas que se interceptam no espaço (tab. 2). Os testes foram feitos com grades de três diferentes dimensões. Note-se que em termos de complexidade, a malha associada a grade das esferas se aproxima bastante das malhas relacionadas a superfícies reais, como, por exemplo, isosuperfícies de pressão, umidade, temperatura, etc.

O tempo de execução do algoritmo não foi explicitado em termos absolutos nas tabelas porque este valor pode variar de máquina para máquina e porque também se considerou que o ponto a ser enfatizado é a relação entre os tempos com e sem busca do vértice com o menor número de polígonos. Ressalte-se, porém, que em estações de trabalho de porte médio, o algoritmo permite gerar a malha das esferas com dimensão 50x50x50 em tempo menor do que 1 segundo.

Pelas duas tabelas pode-se notar que a dimensão da grade aumenta muito pouco percentualmente quando não se faz a busca do vértice com o menor número de polígonos, 2% a mais no caso do plano da grade 50x50x50, no entanto, o tempo de

execução do algoritmo com a busca deste vértice aumenta significativamente, em torno de 4 vezes para o plano na mesma grade.

O aumento no tempo é devido a busca do vértice livre em uma das tabelas de dados associada ao método. Para se achar o vértice com o menor número de polígonos associados deve se pesquisar a tabela Pol\_f\_Vert (polígonos em função de vértices) a cada começo de um novo percurso. Quanto maior o número de vértices maior o tempo gasto na pesquisa desta tabela.

Duas razões são significativas para se ter grades com tamanho muito próximo quando se utiliza ou não a busca do vértice com o menor número de polígonos. Primeiramente, porque quando se alcança um vértice que não permite continuar caminhando pelos polígonos pode-se utilizar o recurso da linha de conexão, o qual tem um pequeno impacto em termos de número de vértices na dimensão total da malha. Em segundo lugar, como citado no item 4, o algoritmo utiliza o mecanismo das máscaras (aplicação inversa) para inicialmente decidir de forma eficiente qual o melhor caminho a seguir, o que, em geral, apresenta um bom resultado em termos de tamanho para as malhas geradas.

Devido a estes fatos, na implementação corrente do algoritmo em uso não utiliza a busca do vértice com menor número de polígonos, no entanto, este recurso pode ser facilmente habilitado.

Tabela 1 - PLANO				
Grade	Dimensão da Malha		Dif_Relat. (Dimensão)	Dif_Relat. (Tempo)
	Com Busca	Sem Busca	(CB/SB)	(CB/SB)
20x20x20	760	796	0,95	2,66
50x50x50	4900	4996	0,98	3,86
70x70x70	9660	9796	0,98	4,10

Tabela 2 - ESFERAS				
Grade	Dimensão da Malha		Dif_Relat. (Dimensão)	Dif_Relat. (Tempo)
	Com Busca	Sem Busca	(CB/SB)	(CB/SB)
20x20x20	3583	3647	0,98	1,30
50x50x50	25831	26001	0,99	1,89
70x70x70	51871	52060	0,99	2,39

## 6. Conclusões

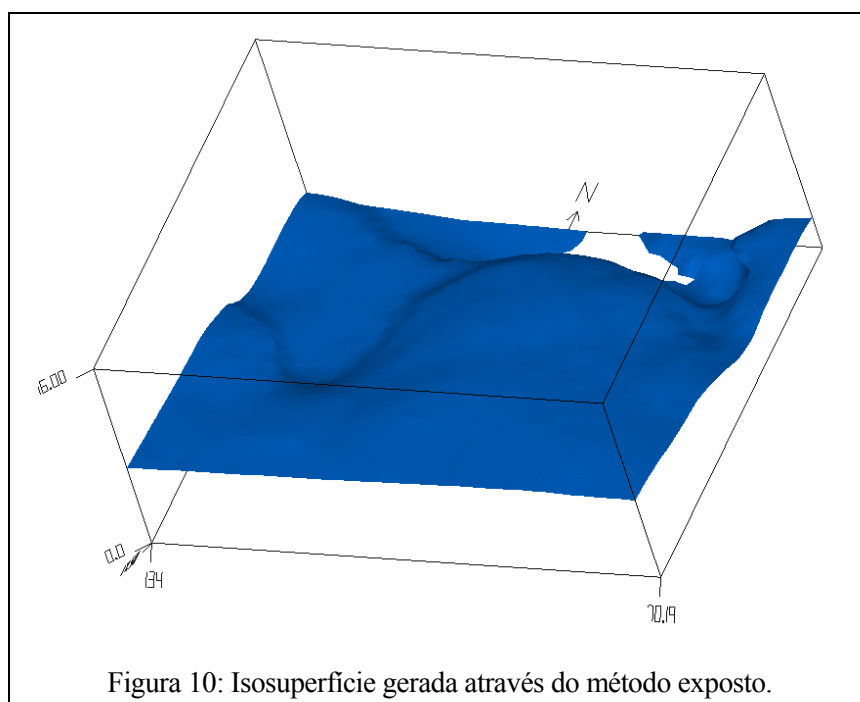
De acordo com o que foi exposto anteriormente, considerou-se que o objetivo de se elaborar um método eficiente para a geração de malhas triangulares a partir de um conjunto de polígonos foi atingido satisfatoriamente. A saída do montador da malha triangular pode ser utilizado diretamente em bibliotecas gráficas como a XFDI, GL e OpenGL, e possivelmente algumas outras. Note-se que como muitos pacotes gráficos estão sendo desenvolvidos sobre a OpenGL, este método pode ser útil para outras aplicações.

De acordo com testes preliminares realizados, constatou-se que a biblioteca VRML 1.0 não opera a contento com o mecanismos anteriormente citados, o que faz com que seja necessário extrair os triângulos da malha e fazer para a exibição deles separadamente. No entanto, o fato da malha estar armazenada compactadamente em

memória favorece algum ganho na eficiência do processo de animação. Além do que a superfície pode ser facilmente exportada para um sistema de visualização.

Em relação ao World ToolKit [WTK96], estudos ainda não foram realizados para precisar a possibilidade ou não do uso deste método.

O método descrito foi incorporado no gerador de isosuperfícies foi utilizado de forma integrada a um algoritmo dos cubos marchantes na geração de gráficos meteorológicos. A figura 10 exibe uma isosuperfície de temperatura.



## 7. Referências

- [BATT92] A. L. Battaiola, “Implementação de um Algoritmo Vetorizado para a Geração de Superfícies Tridimensionais”, Anais do SIBGRAPI V (1992), 145-154.
- [FLORI87] L. D. Floriani, “*Surface Representations Based On Triangular Grids*”, Springer-Verlag, 1987.
- [FOLEY90] J. Foley, A. van Dam, S. Feiner, J. Hughes, “*Computer Graphics - Principles and Practice*”, Addison Wesley, 1990, pag. 473-477, 871.
- [GL90] Silicon Graphics Corporation, “*Graphics Library Programming Guide*”, vol. I, II, 1990.
- [LOREN87] W. E. Lorensen, H. E. Cline, “*Marching Cubes: A High Resolution 3D Surface Construction Algorithm*”, Computer Graphics, vol.21, num.4, julho de 1987.
- [MATS96] S. N. Matsuba, B. Roehl, “*Using VRML*”, Que Corporation, 1996.
- [OGL93a] J. L. Neider, T. R. Davis, M. Woo, “*OpenGL Programming Guide*”, Addison Wesley, 1993.
- [OGL93b] J. L. Neider, T. R. Davis, M. Woo, “*OpenGL Reference Guide*”, Addison Wesley, 1993.
- [STELL90] Stellar Computer Inc, “*XFDI Users and Programming Guide*”, 1990.
- [WTK96] World ToolKit Reference Manual, Sense8 Corporation, 1996.