

Marching Cubes Multiresolução

SANDRO SANTOS ANDRADE¹, THOMAS DE ARAUJO BUCK¹

¹Universidade Federal da Bahia - Instituto de Matemática - Departamento de Ciência da Computação
Av. Ademar de Barros s/n, Ondina, 40170-110 Salvador, BA, Brasil
sandrosa, thomas@dcc.ufba.br

Abstract. A widespread approach to generating polygonal approximations of iso-surfaces in volume data is the *Marching Cubes* algorithm. This algorithm, however, has the disadvantage that the quantity of polygons generated is considerable. The *Multiresolution Marching Cubes* presented here reduces this quantity by a variable size marching cube. The resulting polygonal chain offers a wide spectrum for representing the desired surface.

Keywords. Iso-surfaces, polygonal approximation, multiresolution sampling.

1 Introdução

A obtenção de aproximações poligonais de iso-superfícies é uma alternativa largamente utilizada para visualização de volumes de dados. Dentre suas vantagens destacam-se uma considerável redução dos dados a serem analisados e a existência de suporte em *hardware* para renderização de malhas poligonais.

Em particular, no caso de aproximações por malhas de triângulos, tem-se que o cálculo das normais (e, conseqüentemente, da função de iluminação) fica favorecido, pelo fato de todos os pontos de um triângulo serem coplanares. Por este motivo, algoritmos para extração de malhas de triângulos a partir de volumes de dados são largamente utilizados.

Um dos mais difundidos algoritmos de extração de iso-superfícies é o algoritmo *Marching Cubes* (MC) [3]. Este algoritmo se destaca pelo fato de não necessitar de processamento de imagem para extração da iso-superfície. Porém, dentre as suas principais desvantagens, tem-se que o número de polígonos gerados é considerável. Em volumes médicos atinge-se facilmente 100k triângulos.

A nova abordagem aqui proposta promove a extração de malhas com reduzido número de polígonos sem comprometer de forma acentuada a representação da superfície em estudo.

2 Trabalhos correlatos

Como mencionado, o MC gera malhas com alto número de polígonos pois pode extrair até 4 triângulos por iteração. Para alguns volumes de tomografia computadorizada a extração de iso-superfícies usando o MC pode gerar malhas poligonais de 500k até 2000k [6]. Dessa forma, tem-se um alto custo computacional a nível de armazenamento e visualização

da superfície obtida.

Como soluções para este problema existem algoritmos executados após o MC que realizam a dizimação de triângulos irrelevantes à superfície [6]. Segundo seus autores, o número de triângulos pode ser reduzido em até 75% sem grandes conseqüências.

Este não é um processo muito inteligente, pois está-se gastando tempo de processamento para gerar triângulos que serão posteriormente eliminados.

Como ilustrado na figura 1, o objetivo básico do *Marching Cubes Multiresolução* (MCMR) é gerar uma quantidade reduzida de triângulos evitando, assim, o uso do algoritmo de dizimação.

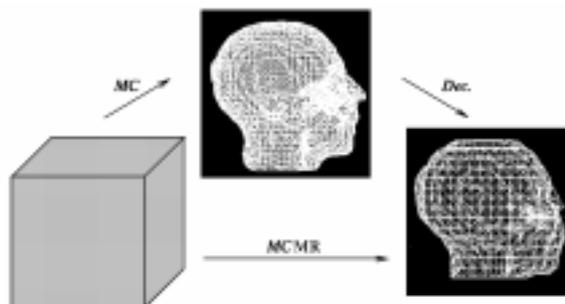


Figura 1: o objetivo do *Marching Cubes Multiresolução* é gerar uma quantidade reduzida de triângulos tornando o uso do algoritmo de dizimação desnecessário.

Um outro algoritmo de poligonização de iso-superfícies é o *Splitting-Box* [4]. O *Splitting-Box* gera malhas poligonais com reduzido número de polígonos, no entanto, possui o inconveniente de gerar quadriláteros, além de triângulos.

3 A abordagem proposta

A idéia básica do MCMR é seguir a mesma abordagem do MC com a vantagem de ter-se um cubo de tamanho variável n . Dessa forma, o cubo irá "marchar" de n em n linhas, colunas e níveis. Em outras palavras, aumentando-se o deslocamento do cubo ao longo das linhas, colunas e níveis do volume, reduz-se o número de triângulos gerados.

O algoritmo proposto recebe dois parâmetros: o valor da iso-superfície e o tamanho n do cubo.

A figura 2 apresenta algumas iterações do algoritmo ilustrando o seu funcionamento. Acima, com o cubo de tamanho 1, tem-se oito iterações e, conseqüentemente, um número maior de triângulos. Abaixo, com o cubo de tamanho 2, tem-se uma iteração somente, desprezando os *voxels* entre os extremos das arestas e reduzindo assim o número de triângulos gerados.

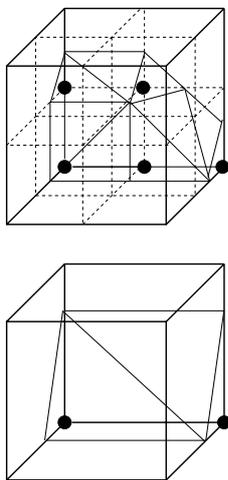


Figura 2: triângulos gerados com cubos em dois tamanhos. Nota-se claramente a redução na quantidade de triângulos gerados.

O tratamento dos 15 casos é feito de modo semelhante ao MC: cálculo do índice, utilização de *lookup tables* e interpolações de vértices e normais. Com esta abordagem, tem-se um sistema multiresolução de geometrias que não afeta consideravelmente a forma da superfície. Ao final consegue-se uma representação para uma iso-superfície com malhas mais detalhadas e malhas menos detalhadas.

4 Implementação

Seguindo o paradigma de orientação à objetos, foram implementadas as seguintes classes:

List: utilizada para gerenciamento e armazenamento de vértices e triângulos.

TriangMesh: utilizada para gerenciamento de malhas de triângulos. Composta de duas listas: lista de vértices e lista de triângulos.

Vis: classe responsável pela visualização de malhas poligonais.

FastVol8: responsável pelo gerenciamento e armazenamento de volumes de dados.

MarchingCubesMR: responsável pela extração multiresolução da geometria de iso-superfícies. Promove métodos tais como: ajustar volume, ajustar tamanho do cubo e ajustar limiar.

As malhas de triângulos geradas pelo algoritmo são exportadas em arquivos com um formato adequado para posterior renderização. Para extração das superfícies utilizou-se o seguinte código C++ :

```
#include <CG/MarchingCubesMR.h>

#define ARG_PROGRAM      0
#define ARG_DENFILE     1
#define ARG_THRESHOLD   2
#define ARG_CUBESIZE    3
#define ARG_TRIANGFILE  4
#define TOTAL_ARGS      5

void main (int argc, char** argv)
{
    MarchingCubesMR mc;
    mc.setDenFile (argv[ARG_DENFILE]);
    mc.setThreshold (atoi(argv[ARG_THRESHOLD]));
    mc.setCubeSize (atoi(argv[ARG_CUBESIZE]));
    mc.makeTriangMesh ();
    mc.save (argv[ARG_TRIANGFILE]);
}
```

5 Resultados

Para verificar os efeitos da multiresolução foi criado um volume artificial contendo um simples cubo. Após a extração de várias superfícies em resoluções diferentes, foram obtidos os resultados mostrados na figura 3.

O conjunto de imagens na figura 4 ilustra as malhas de triângulos obtidas a partir de um volume de tomografia computadorizada. A partir de uma variação do limiar obtem-se o crânio e o rosto do paciente e a partir de uma variação no tamanho do cubo obtem-se diferentes resoluções da malha.

6 Discussão e conclusões

Utilizando o algoritmo MC o número de triângulos extraídos de volumes médicos pode facilmente chegar a 100k. Fazendo uma análise a nível de renderização, suponha-se uma imagem 256x256. Ao admitir

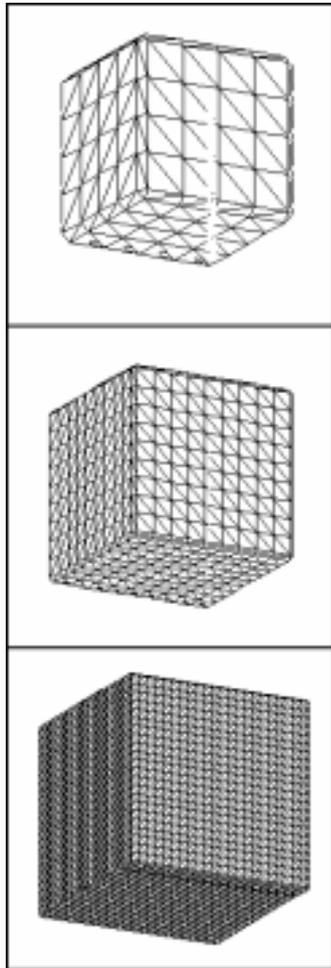


Figura 3: malhas poligonais obtidas com cubos de tamanhos 32, 16 e 8 (de cima para baixo). O número de triângulos gerados foi 296, 1196 e 6344 respectivamente. Utilizando-se o algoritmo MC obtém-se 307196 triângulos.

que cada triângulo ocupe um *pixel* apenas, ter-se-ia um máximo de 65535 triângulos por imagem. Desta forma, tem-se que 100k é um número relativamente alto para a renderização de pequenas imagens pois um *pixel* conteria mais de um triângulo.

Para a solução deste problema, foi proposto o algoritmo *Dividing Cubes* (DC) [2]. Entretanto, o DC não possui a capacidade de extrair malhas poligonais visto que é um algoritmo de renderização. O MCMR é uma outra alternativa para a solução deste problema.

Uma outra aplicação do MCMR é a produção de animações. Como observado, utilizando o MC o tempo de renderização das malhas resultantes im-

pede a construção de animações satisfatórias (mínimo de 30 quadros por segundo). Segundo a técnica conhecida como *refinamento progressivo*, anima-se o objeto de interesse em malhas menos densas e, escolhida a posição para obter mais detalhes, interrompe-se o processo de interação e inicia-se o refinamento, renderizando as malhas mais densas. Evidencia-se assim a necessidade de se ter uma hierarquia de malhas poligonais.

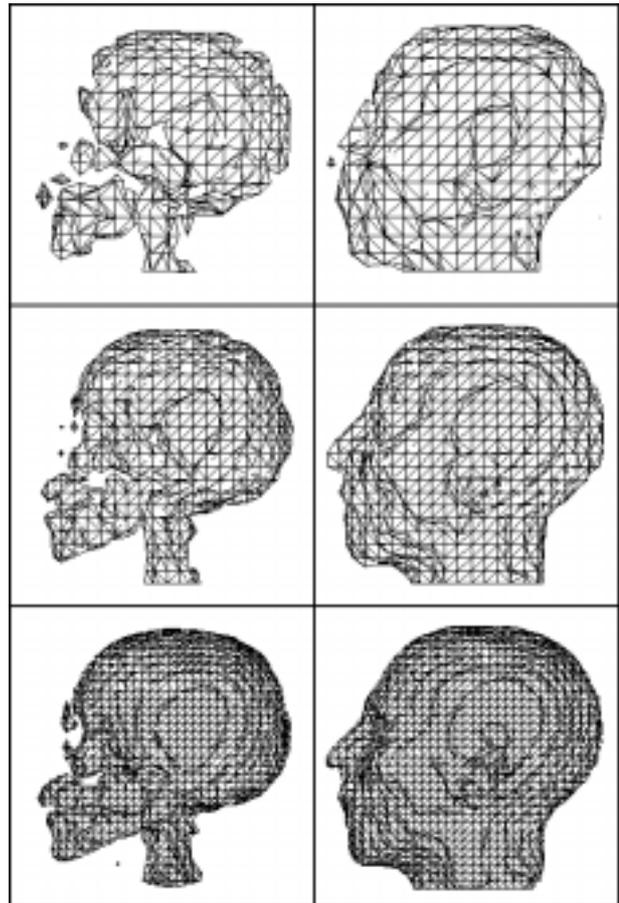


Figura 4: na esquerda estão ilustradas as malhas poligonais obtidas com limiar igual a 100 e cubos de tamanhos 16, 12 e 8 (de cima para baixo). O número de triângulos gerados foi 2288, 4698 e 12798 respectivamente. Na direita estão ilustradas as malhas poligonais obtidas com limiar igual a 50 e cubos de tamanhos 16, 12 e 8 (de cima para baixo). O número de triângulos gerados foi 1754, 3446 e 9286 respectivamente.

Como desvantagem, tem-se que o MCMR continua sendo um algoritmo que pode apresentar extracções ambíguas da malha poligonal. Para o fu-

turo, implementar-se-á o *Marching Tetrahedra Multi-resolução*, eliminando assim o problema da ambigüidade [1].

Como ilustrado na figura 2 nota-se que o MCMR despreza alguns *voxels* do volume de dados durante a análise dos casos. Como solução para este problema propõe-se realizar uma redução da resolução do volume através de processos de filtragem [5] e, posteriormente, aplicar o MCMR com um cubo de tamanho menor para extração da superfície.

Devido à necessidade médica de resultados em tempo real, algoritmos de poligonalização de superfícies implícitas estão sendo cada vez mais utilizados, pois o suporte em *hardware* existente promove visualizações satisfatórias a nível de tempo e qualidade. O alto custo computacional e a grande quantidade de dados gerados pelos exames médicos foram os principais fatores de motivação à elaboração do MCMR.

Nesta contribuição foram discutidos a idéia, implementação e resultados da abordagem proposta. Também foram analisados vantagens, desvantagens e trabalhos futuros, ratificando assim a crescente importância da computação visual na medicina.

Agradecimentos

Os autores gostariam de agradecer ao CNPq e à PRPPG/UFBA pela concessão de bolsas de Recém-Doutor e *Kit-Doutor*, sem as quais este trabalho não teria sido possível e aos demais membros do GIC_{OV} (Grupo de Interesse em Computação Visual): Eduardo Prates de Oliveira, Fábio André Silva Reis, Maurício de Azevedo Gomes e Mitre Costa Dourado.

Bibliografia

- [1] Bernardo Piquet Carneiro, Cláudio T. Silva e Arie E. Kaufman. Tetra-cubes: An algorithm to generate 3D isosurfaces based upon tetrahedra. Em Luiz Velho, Arnaldo de Albuquerque e Roberto A. Lotufo, editores, *SIBGRAPI 96 – IX Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, pages 205 – 210, Caxambú – MG, Outubro 1996. Sociedade Brasileira de Computação (SBC) – Universidade Federal de Minas Gerais (UFMG).
- [2] H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford e B. C. Teeter. Two algorithms for the three-dimensional reconstruction of tomograms. *Med. Phys.*, 15(3):320 – 327, May / June 1988.
- [3] William E. Lorensen e Harvey E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163 – 169, July 1987.
- [4] Heinrich Müller e Michael Stark. Adaptive generation of surfaces in volume data. *The Visual Computer*, 9(4):182 – 199, January 1993.
- [5] Rafael C. Gonzalez e Paul Wintz. *Digital Image Processing*. Addison Wesley, 1987.
- [6] William J. Schroeder, Jonathan A. Zarge e William E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65 – 70, July 1992.