

# Malhas Segmentadas com Resolução Variável

Rui S. V. Rodrigues  
ESTGV-IPV & U. da Beira Interior  
Viseu & Covilhã  
rsrodrigues@estv.ipv.pt

José F. M. Morgado  
ESTGV-IPV  
Viseu  
fmorgado@estv.ipv.pt

Abel J. P. Gomes  
Universidade da Beira Interior  
Covilhã  
agomes@di.ubi.pt

## Resumo

*As malhas com resolução variável (“multiresolution meshes”) têm sido utilizadas como alternativa à técnica dos LODs em ambientes virtuais e jogos de computador, em que o detalhe da cena depende da distância ao observador, bem como na (des)compressão de malhas em tarefas de transferência e transmissão mais célere de malhas através da Internet e da web, o que permite o carregamento progressivo da malha do lado do cliente, que começa com uma malha mais grosseira e termina com uma malha mais refinada. A principal contribuição deste artigo reside na utilização do conceito de multirresolução em malhas segmentadas, em vez de malhas simples (sem segmentação).*

## Palavras-Chave

*Malhas triangulares, multirresolução, segmentação, simplificação de malhas, refinamento de malhas.*

## 1. INTRODUÇÃO

As malhas de resolução variável (também, designadas por malhas multirresolução) surgiram como alternativa às malhas LOD (*level of detail*), por causa da poupança que se obtém de espaço em memória. Na verdade, enquanto um LOD é uma sequência finita de malhas com resoluções diferentes de um mesmo objeto residente em memória, uma malha de resolução variável requer só uma malha do mesmo objeto em memória. No entanto, uma malha multirresolução pode ser refinada e simplificada através de operadores de refinamento e de simplificação, respetivamente. Como dizia Garland [Garland 97], independentemente do nível de detalhe, uma malha de resolução variável deve manter ou, quando muito, aumentar ligeiramente as necessidades de espaço em memória relativamente ao espaço de memória ocupada pelo malha com mais detalhe.

Uma das preocupações sempre presentes para quem pretende desenhar e implementar um esquema de multirresolução prende-se com a preservação da forma do objeto que a malha representa, em particular quando se procede à simplificação da malha. De facto, o processo de simplificação tende a esbater as concavidades da malha. Para obviar este problema, utiliza-se normalmente um critério de manutenção de forma; por exemplo, a coalescência de dois triângulos que incidem numa aresta só tem lugar quando o ângulo diedral é zero ou próximo de zero. Neste artigo, a ideia não é usar um critério genérico de preservação da forma do objeto, mas sim utilizar a segmentação da malha em partes como mecanismo de preservação da forma. Noutras palavras, preserva-se a forma de uma malha preservando a sua segmentação em

partes.

Por isso, pode ser importante tentar perceber como os seres humanos percecionam os objetos e a sua forma. Como foi observado por Biederman [Biederman 87], o ser humano perceciona um dado objeto como um conjunto de partes. Já Hoffman [Hoffman 97] concluiu que o ser humano define as fronteiras entre as diferentes partes dos objetos pelas zonas de curvatura negativa mínima. Falar de um objeto como um agregado de várias partes significa *a priori* a existência de uma malha segmentada em diferentes regiões, tendo-se para isso utilizado um determinado critério de segmentação [Attene 06]. Veja-se o levantamento do estado da arte levado a cabo por Shamir em [Shamir 08] para mais informação sobre critérios e técnicas de segmentação de malhas trianguladas.

Com base na asserção acima referida, como sendo devida a Hoffman [Hoffman 97], Rodrigues et al. [Rodrigues 15] propuseram o algoritmo HPO (*human perception-oriented segmentation*), o qual segmenta a malha que representa um dado objeto em conformidade com a percepção em partes que o ser humano tem do dito objeto, pelo que se se quiser preservar a forma deve-se preservar as partes (ou segmentos); algumas malhas segmentadas através do algoritmo HPO podem ser vistas na Figura. 1.

A técnica de preservação da forma descrita neste artigo para malhas segmentadas com resolução variável tem por objetivo não só preservar os ciclos de arestas que separam as regiões (ou segmentos) da malha entre si, mas também os caminhos vincados (ou vincos), os ápices e os pontos máximos de curvatura (por exemplo as extremidades dos dedos de uma mão) existentes na malha, que são facilmente identificados pelos valores elevados da sua curva-

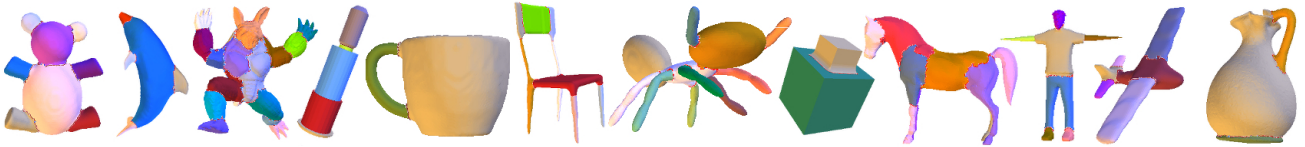


Figura 1. Malhas segmentadas com o algoritmo HPO.

tura, os quais —como refere Mangan et al. [Mangan 99]— são calculados através da norma da matriz de covariância.

Assim sendo, as principais contribuições deste artigo são as seguintes:

- Tanto quanto julgamos saber, propõe-se a primeira estrutura de dados para malhas *segmentadas* com resolução variável.
- Um algoritmo de simplificação que usa a informação da segmentação para manter a forma da malha. É preservada a fronteira entre as diferentes regiões da malha de forma a preservar os segmentos da malha.
- Preserva-se ainda os vincos e os pontos característicos (por exemplo, ápices) da malha.
- O refinamento e a simplificação podem ser aplicados a toda a malha ou somente a um conjunto específico de segmentos da malha.

O restante artigo está organizado da seguinte forma. Na secção 2 faz-se um resumo do trabalho relacionado existente. Na secção 3 é explicada a estrutura de dados proposta para suportar as malhas segmentadas de resolução variável. Nas secções 4 e 5 estão explicados (respetivamente) os algoritmos de simplificação e refinamento. Na secção 6 são apresentados os resultados do trabalho. E, finalmente, na secção 7 são feitas algumas considerações finais sobre o trabalho realizado.

## 2 TRABALHO RELACIONADO

Na literatura, existem duas famílias principais de malhas de resolução variável: subdivisão recursiva e subdivisão não-recursiva. Na subdivisão recursiva dividem-se os polígonos recursivamente todos da mesma maneira. Por exemplo, todo e qualquer triângulo é dividido em três ou quatro triângulos mais pequenos, de maneira que a malha vai ficando visualmente cada vez mais suave *quadtree* [Schröder 00] [Floriani 02]. Na subdivisão não-recursiva utiliza-se *a priori* uma malha com resolução máxima, a qual poderá então ser simplificada, bem como ser refinada se a resolução da malha não estiver no nível máximo [Garland 99]. Além disso, no caso da subdivisão não-recursiva, nem todos os triângulos (ou quadrângulos) serão sujeitos a refinamento/simplificação em cada nível de detalhe ou resolução.

No presente artigo, usar-se-á um esquema de multirresolução não recursiva. Normalmente, um esquema de multirresolução não recursiva utiliza a estrutura de dados para duas finalidades, nomeadamente:

- *Repositório das malhas de resolução variável.* O armazenamento de uma malha de resolução variável em memória serve dois propósitos, isto é, permite fazer operações de edição e operações de visualização.
- *Repositório das alterações das malhas.* Com este repositório guarda-se o historial das alterações, o que permite avançar ou voltar a uma malha específica com maior ou menor resolução.

Segundo Silva e Gomes [Silva 03], a natureza dos elementos que constituem uma malha tem um impacto significativo no *desempenho* das estruturas de dados das malhas de resolução variável. Uma forma de melhorar o desempenho é através de definição de esquemas de incidência entre os diferentes elementos da malha (vértices, arestas e faces). Ni e Bloor [Ni 94] chegaram à conclusão que a estrutura de dados simétrica  $C_4^9$  é ótima para malhas bi-dimensionais. E este tipo de esquema de incidência já o encontramos nas estruturas de dados como a AIF [Silva 03] e GC (*ghost-cell* ou *aresta fantasma*) [Rodrigues 07].

Nos modelos de multirresolução não-recursiva, temos que codificar explicitamente as alterações e as dependências da malha na estrutura de dados. Essa codificação depende do tipo de elemento da malha que é modificado [Floriani 02], o que nos leva à seguinte classificação: multirresolução baseada em operações sobre arestas e multirresolução baseada em operações sobre vértices. No primeiro caso, utiliza-se operações de contração da aresta (*edge collapse*) e de divisão do vértice (*vertex split*) [Hoppe 96] [ES99] [Danovaro 05], ao passo que no segundo caso utiliza-se operações de inserção e remoção de vértices [Schroeder 92] [Ciampalini 96]) [Rossignac 93] [Luebke 97].

O modelo de multirresolução proposto utiliza um algoritmo de simplificação baseado na operação de contração da aresta. Por isso, a aresta a contrair deve ser selecionada segundo um determinado critério, nomeadamente: a chamada função de energia usada em *Mesh Optimization* [Hoppe 93] e otimizada nas *Progressive Meshes* [Hoppe 96]; minimização do erro associado a cada vértice usada em *Surface Simplification Using Quadric Error Metrics* [Garland 97]; já em *Normal-based Simplification Algorithm* (NSA) [Silva 04] escolhe-se a aresta a contrair com base na variação das normais às faces em redor da aresta alvo.

Em cenas 3D, é comum aplicar-se multirresolução às partes visíveis das malhas que representam objetos existentes numa dada cena, em particular no con-

texto de compressão e transmissão de geometria [Kim 01] [Yang 04] [Kim 06] [Maglo 11] [Maglo 13]. Aliás, é comum também usar-se a distância ao observador como critério de multirresolução [Hoppe 97] [ES99] [Pajarola 01] [Choe 09], em particular em modelos de terrenos. Neste sentido, poderá dizer-se que existem partes com mais ou menos detalhe, que poderiam ser designados por segmentos não fosse o facto de que as partes visíveis da malha dependerem da posição do observador. Quando muito, estas partes da malha que não são perceptualmente significativas poderão ser designadas por regiões (*charts*, do inglês).

Nos trabalhos anteriores não é tido em conta qualquer relação entre as regiões e as partes significativas dos objetos (i.e., segmentos). Este tipo de relação encontra-se plasmada pela primeira vez no trabalho de Cheng et. al. [Cheng 07], em que os segmentos são delineados pelos principais contornos de curvatura mínima da malha, embora este trabalho se tenha inspirado no trabalho de Yang et al. [Yang 04], que dividiam a malha em regiões perceptualmente não-significativas. Refira-se, no entanto, que Cheng e seus colegas não tiveram em conta a preservação das fronteiras dos segmentos.

Ao invés, o esquema de multirresolução aqui proposto serve o propósito de malhas segmentadas, em que se preserva a forma através da preservação das fronteiras dos segmentos. Além do mais, é possível ter diferentes níveis de multi-resolução ou de detalhe em diferentes segmentos, mesmo entre segmentos adjacentes ou vizinhos.

### 3 ESTRUTURA DE DADOS BASEADA NA CÉLULA FANTASMA

Neste trabalho, utilizou-se uma estrutura de dados baseada na célula fantasma (*ghost-cell*, do inglês) que foi desenvolvida por Rodrigues et al. [Rodrigues 07]. A estrutura de dados GC (*ghost-cell*) é uma estrutura de dados ótima do tipo  $C_4^9$ , que codifica as seguintes relações topológicas:  $V \prec E$  e  $E \prec F$ , e as suas relações inversas,  $E \succ V$  e  $F \succ E$ .

Esta estrutura de dados foi estendida de molde a suportar malhas segmentadas, pelo que a esta variante da GC foi atribuída a designação de xGC (*extended GC*). As alterações introduzidas na xGC foram as seguintes:

- *Class Mesh*. Nesta classe foi incluída uma lista de segmentos.
- *Class Segment*. Esta é uma nova classe, a qual inclui o identificador do segmento e uma lista de referências para as faces que fazem parte do segmento.
- *Class Face*. Esta classe inclui uma referência para o segmento a que a face pertence.
- *Class Edge*. Nesta classe foi incluída uma *flag* para indicar se a aresta pertence à fronteira de um segmento ou não.
- *Class Vertex*. Nesta classe foi incluída uma *flag* para indicar se o vértice pertence ou não à fronteira do segmento.

Recorde-se que o carácter de resolução variável da estrutura de dados GC advém da existência de duas estruturas genealógicas, uma para os vértices e outra para as arestas. Também estas estruturas genealógicas sofreram as seguintes modificações:

- *Genealogia do vértice*. Nesta estrutura introduziu-se um novo campo para guardar informação sobre o tipo de contração de aresta que esteve na origem do vértice.
- *Genealogia da aresta*. Nesta estrutura foi otimizada a utilização da memória, porque a partir de certos níveis de simplificação os vértices fronteira da aresta sobrepõem-se (resultam da contração da aresta).

### 4 ALGORITMO SIMPLIFICAÇÃO

No algoritmo implementado aplica-se a operação de contração da aresta para conseguir estes objetivos. Para a simplificação é necessário percorrer o array de arestas e seleccionar as arestas a contrair. Estas são seleccionadas segundo critérios topológicos e geométricos. Em termos de critérios topológicos usamos os mesmos que foram aplicados por Rodrigues et. al. [Rodrigues 07], nomeadamente, num nível de simplificação se uma aresta é contraída nesse nível, todas as suas *wing edges* não podem ser contraídas nessa mesma iteração. Este critério garante a aplicação mais uniforme da simplificação, dentro do mesmo nível de simplificação e de nível para nível. Em termos geométricos, apresentamos dois critérios baseados na forma como o ser humano percepcionam os objetos, e que são:

1. Preservar as partes significativas dos objetos. Nesse sentido, a malha é segmentada uma única vez (isto é, antes de qualquer simplificação), usando o algoritmo HPO [Rodrigues 15]. Este de forma automática segmenta a malha e identifica as partes significativas dos objetos, sendo a informação dos segmentos guardada na estrutura de dados xGC. Durante a simplificação da malha, a fronteira dos segmentos não é alterada de maneira a preservar as diferentes partes do objeto. Isto vai implicar que todas as arestas que pertencem à fronteira não sejam contraídas. Nas Figuras 4, 5 e 6, pode-se verificar que a linha de fronteira entre segmentos não sofreu qualquer alteração durante as sucessivas simplificações da malha. Além disso, todas as *wing edges* dos vértices que ficam na fronteira dos segmentos (excepto as que fazem parte da fronteira), quando contraídas, terão que o ser de modo a não haver deslocação das arestas da fronteira.
2. Preservar vincos e pontos característicos. Os objetos também têm normalmente um conjunto de vincos e pontos característicos que são percecionados pelos seres humanos, o que fazemos é tentar preservá-los. Estes caracterizam-se na maior parte das vezes pelos valores elevados de curvatura [Mangan 99], sendo que uma forma de preservar esses pontos é impedir a

contração de arestas e de *wing edges* de vértices com elevada curvatura.

No seu conjunto, a segmentação da malha, os critérios de preservação de forma referidos anteriormente e a estrutura de dados xGC permitem também simplificar um conjunto específico de segmentos (simplificação parcial). Este segmentos podem ser seleccionados e simplificados de diferentes formas, como por exemplo:

1. Ponto de vista do observador. Os segmentos mais próximos do observador com simplificações menores que os que estão mais longe.
2. Seleccionado pelo utilizador. Os segmentos são seleccionados por ação direta dos utilizadores, sendo que só estes são simplificados.

No nosso algoritmo, e de forma a permitir preservar a fronteira dos segmentos, tem então que se aplicar diferentes tipos de contração de aresta (Figura 2), nomeadamente:

1. Contração para ponto médio. Quando a aresta a contrair  $(V_i, V_j)$  não pertence às *wing edges* dos vértices da fronteira, a contração desta é feita segundo a forma definida por Hoppe [Hoppe 98], ou seja, a aresta  $(V_i, V_j)$  é contraída para o ponto médio  $P = (V_i + V_j)/2$  da mesma, sendo posteriormente calculado o chamado vetor de Hoppe  $\vec{v}_H = V_i - P$  que permite depois reconstruir a aresta. O vértice criado é classificado como sendo do tipo 1.
2. Contração para extremidade. Quando é uma aresta que pertence às *wing edges* dos vértices da fronteira (mas não é fronteira do segmento), a aresta  $(V_i, V_j)$  é contraída para um dos pontos extremos da mesma [Kobbelt 98], isto é,  $P = V_i$  se  $V_i$  é o vértice da fronteira, sendo que o vetor de Hoppe é dado por  $\vec{v}_H = V_j - V_i$ , e o vértice criado é classificado como sendo do tipo 2; do mesmo modo,  $P = V_j$  se  $V_j$  é o vértice da fronteira, sendo que o vetor de Hoppe é dado por  $\vec{v}_H = V_i - V_j$ , e o vértice criado é classificado como sendo do tipo 3.

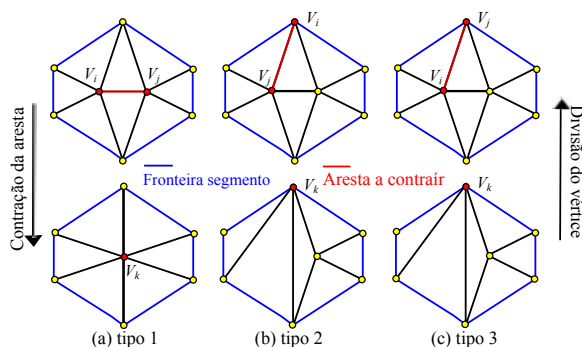


Figura 2. Tipos de contração da aresta.

O algoritmo de simplificação opera sobre o *array* de arestas de forma sequencial. Se uma aresta  $(V_i, V_j)$  é passível de ser contraída, então a contração é efectuada do seguinte modo:

1. Determina o tipo de aresta.
2. Calcula/define o novo vértice  $P$  resultante do colapso da aresta.
3. Calcula o vetor de Hoppe  $\vec{v}_H$ .
4. Testa se haverá inversão de normais das faces incidentes em  $V_i$  e  $V_j$ , se estes passarem a ter as coordenadas de  $P$ . Em caso afirmativo, é anulada a contração da aresta (situação de erro pode ser visualizada na Figura 3); caso contrário, segue-se para passos seguintes para efectivar a contração da aresta.
5.  $V_i$  é redefinido como o novo vértice  $V_{N+1}$  a partir de  $P$ , onde  $N$  indica o maior índice dos vértices.
6. Define  $\vec{v}_H$  para  $V_{N+1}$ .
7. Criar a sub-árvore genealógica de  $V_{N+1}$ , i.e.  $V_{N+1}(\text{id}(V_i), \text{id}(V_j))$ , sendo os identificadores  $\text{id}(V_i)$  e  $\text{id}(V_j)$  armazenados como nó esquerdo e direito, respetivamente. Além disso é guardada informação acerca do tipo de aresta envolvida na contração.
8. Actualiza a lista de arestas da malha.
9. Actualiza lista de arestas de  $V_{N+1}$  com as de  $V_j$ .
10.  $V_j$  guardado na lista de vértices fantasma.

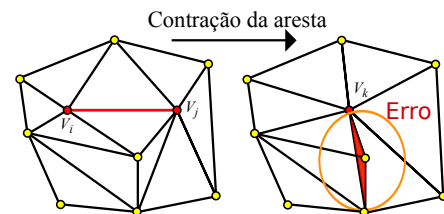


Figura 3. Erro da contração da aresta.

## 5 ALGORITMO DE REFINAMENTO

Este algoritmo permite restaurar a malha (ou os segmentos da malha) para o nível seguinte de maior detalhe. Basicamente o que se faz nesta iteração é aplicar o algoritmo de refinamento proposto por Rodrigues et. al. [Rodrigues 07], mas com as alterações que a seguir se descrevem.

A informação que permite restaurar a malha para um nível de detalhe superior está toda ela armazenada nas árvores genealógicas associadas aos vértices. Aquando da operação de divisão de um vértice  $V_k$ , há que primeiro verificar o tipo de contração de aresta que esteve na origem desse vértice. A partir dessa informação é possível então restaurar a geometria dos vértices que faziam parte da aresta contraída, o que é feito do seguinte modo:

1. Tipo 1. Neste caso (contração da aresta para o ponto médio), os novos vértices  $V_i$  e  $V_j$  serão iguais a  $V_i = V_k - \vec{v}_H$  e  $V_j = V_k + \vec{v}_H$ .
2. Tipo 2. Neste caso (contração da aresta para o  $V_i$  da aresta), os novos vértices  $V_i$  e  $V_j$  serão iguais a  $V_i = V_k$  e  $V_j = V_k + \vec{v}_H$ .
3. Tipo 3. Neste caso (contração da aresta para o  $V_j$  da aresta), os novos vértices  $V_i$  e  $V_j$  serão iguais a  $V_i = V_k + \vec{v}_H$  e  $V_j = V_k$ .

## 6 RESULTADOS EXPERIMENTAIS

No final interessa conhecer quais os resultados da aplicação do esquema de multirresolução nomeadamente no que diz respeito:

1. Preservação de forma. Verificar se a forma geral do objeto e dos segmentos se mantêm com a aplicação sucessiva de simplificações. De referir que a simplificação pode ser aplicada a todo o objeto ou somente a um conjunto de segmentos da malha.
2. Taxa de simplificação. Verificar qual a percentagem de redução de polígonos da malha durante as sucessivas etapas de simplificação da malha.

Para os testes foram utilizados os modelos existentes no Princeton Benchmarking [Chen 09], que disponibiliza uma grande diversidade de objetos divididos por 19 famílias de objetos.

### 6.1 Preservação da forma

A fim de avaliar como a forma dos objetos é preservada, fizemos dois testes. No primeiro simplificou-se as malhas sem alterar a distância ao observador, com o intuito de demonstrar que a qualidade visual da malha se mantém. No segundo teste, a simplificação só incidiu num segmento da malha, e não na malha no seu todo. Os resultados do primeiro teste estão ilustrados nas Figuras 4 e 5, onde para dois objetos bastante diferentes se aplicaram sucessivas simplificações, nomeadamente 2 e 4 iterações. Como se pode observar, os objetos apesar de terem reduções significativas do número de faces, mantêm a forma dos mesmos e preservam a fronteira entre as diferentes regiões. Além disso, os vincos e pontos característicos dos objetos dentro de cada segmento permanecem sem alterações significativas. A junção do critério topológico com os critérios geométricos permite uma distribuição mais uniforme da simplificação, preservando a forma da malha, tanto a nível geral como em termos de segmentos e pontos característicos do objeto.

Na Figura 6 ilustra-se os resultados do segundo teste. Neste caso, pode-se observar a aplicação parcial da simplificação à asa direita da malha de um avião que foi selecionado pelo utilizador. Este segmento foi sendo sucessivamente simplificado, embora a figura só mostre a asa com 2 e 4 iterações de simplificação. Pode-se observar que a asa preserva a forma, ainda que o número de faces tenha

sido bastante reduzido, tendo passado de um número inicial de 2915 faces para 471 faces. Além disso, a fronteira do segmento da asa foi completamente preservado durante este processo.

### 6.2 Taxa de Simplificação

Na Figura 7 estão representados os resultados do algoritmo de simplificação em termos de percentagem de redução do número de faces, aplicados a 19 tipos de objetos diferentes. Como se pode observar, numa primeira fase existe uma redução de aproximadamente 50% no número de faces. Esta redução no número de faces dos objetos verifica-se em média até ao quinto e sexto níveis de simplificação. A partir destes níveis, a simplificação deixa de ocorrer devido aos critérios geométricos que são usados na preservação da forma dos objetos. Pode-se também constatar que as superfícies com um maior número de segmentos, com formas cilíndricas e em que o diâmetro é mais pequeno, acabam por ter no final têm uma taxa de redução de polígonos menor, pois têm uma maior curvatura.

## 7 CONCLUSÕES

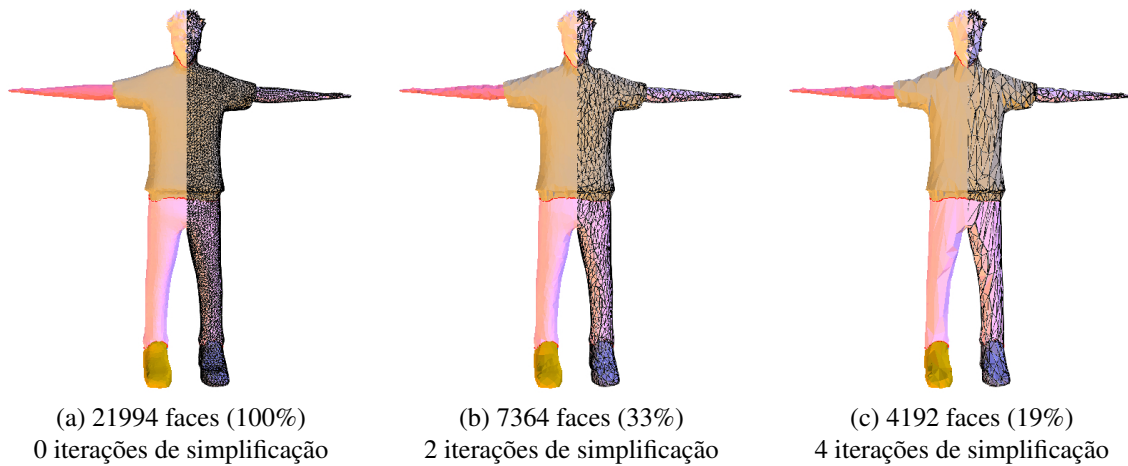
Neste artigo foi apresentado um esquema completo para malhas segmentadas com resolução variável. Este esquema incorpora uma estrutura de dados, designada xGC, que permite armazenar malhas segmentadas e que dá suporte ao funcionamento dos algoritmos de simplificação e refinamento da malha de resolução variável que preservam a forma. A preservação da forma faz -se à custa da preservação das fronteiras entre segmentos e dos principais pontos característicos da malha. A utilização de malhas segmentadas permite variar a resolução de toda a malha ou então de só uma parte específica desta malha, isto, é de um segmento ou conjunto de segmentos.

Todo este processo pode ser feito de forma automática desde o algoritmo de segmentação ao algoritmo de simplificação, pois o método de simplificação pára automaticamente na quinta ou na sexta iteração de simplificação, em resultado dos critérios geométricos impostos no processo de simplificação.

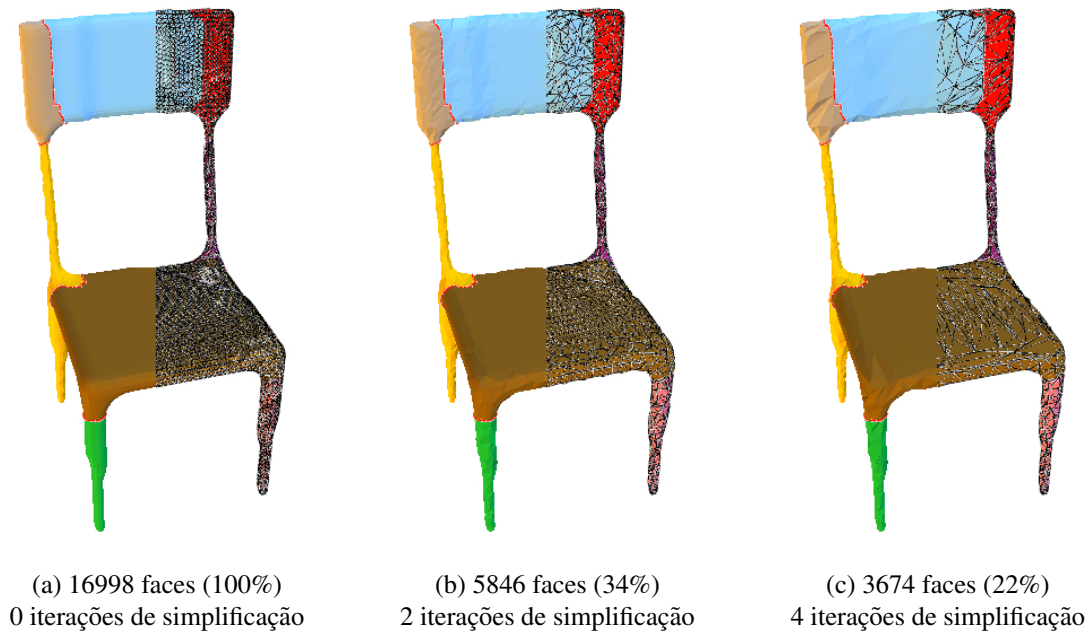
Como trabalho futuro, pretende-se fazer a introdução de uma suavização da malha após cada simplificação, mas esta operação implica alterações a nível do armazenamento da informação na genealogia dos vértices.

### Referências

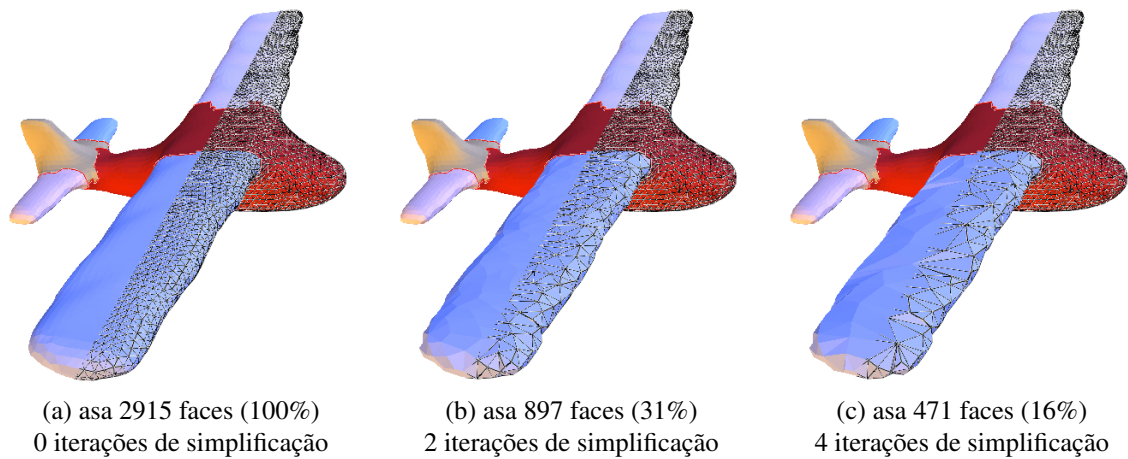
- [Attene 06] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, e A. Tal. Mesh segmentation — a comparative study. Em *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI'2006)*, páginas 7–19. IEEE Computer Society, 2006.
- [Biederman 87] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.



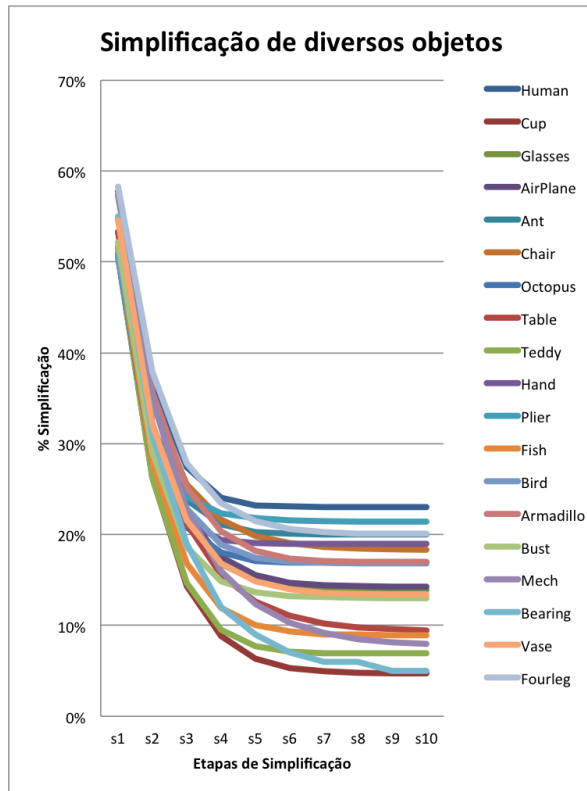
**Figura 4. Simplificação de malha de ser humano .**



**Figura 5. Simplificação de malha de uma cadeira .**



**Figura 6. Simplificação da asa direita da malha de um avião .**



**Figura 7. Evolução da simplificação da malha.**

- [Chen 09] Xiaobai Chen, Aleksey Golovinskiy, e Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics*, 28(3):73:1–73:12, August 2009.
- [Cheng 07] Zhi-Quan Cheng, Hua-Feng Liu, e Shi-Yao Jin. The progressive mesh compression based on meaningful segmentation. *The Visual Computer*, 23(9-11):651–660, 2007.
- [Choe 09] Sungyul Choe, Junho Kim, Haeyoung Lee, e Seungyong Lee. Random accessible mesh compression using mesh chartification. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):160–173, Janeiro 2009.
- [Ciampalini 96] A. Ciampalini, P. Cignoni, C. Montani, e R. Scopigno. Multiresolution decimation based on global error. Relatório técnico, Paris, France, France, 1996.
- [Danovaro 05] E. Danovaro, L. Floriani, P. Magillo, E. Puppo, D. Sobrero, e N. Sokolovsky. The half-edge tree: A compact data structure for level-of-detail tetrahedral meshes. Em *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005 (SMI'05)*, páginas 334–339, Washington, DC, USA, 2005. IEEE Computer Society.
- [ES99] Jihad El-Sana e Amitabh Varshney. Generalized view-dependent simplification. *Computer Graphics Forum*, 18(3):83–94, 1999.
- [Floriani 02] Leila Floriani e Paola Magillo. Multi-resolution mesh representation: Models and data structures. Em *Tutorials on Multiresolution in Geometric Modelling*, páginas 363–417. Springer, 2002.
- [Garland 97] Michael Garland e Paul S Heckbert. Surface simplification using quadric error metrics. Em *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, páginas 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.
- [Garland 99] Michael Garland. Multiresolution modeling: Survey & future opportunities. *State of the Art Report*, páginas 111–131, 1999.
- [Hoffman 97] Donald D. Hoffman e Manish Singh. Saliency of visual parts. *Cognition*, 63(1):29 – 78, 1997.
- [Hoppe 93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, e W. Stuetzle. Mesh optimization. Em *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, páginas 19–26, New York, USA, 1993. ACM Press.
- [Hoppe 96] Hugues Hoppe. Progressive meshes. Em *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, páginas 99–108. ACM, 1996.
- [Hoppe 97] Hugues Hoppe. View-dependent refinement of progressive meshes. view-dependent refinement of progressive meshes. Em *Proceedings of the 24rd annual conference on Computer graphics and interactive techniques*, volume 31, páginas 189–198. ACM, 1997.
- [Hoppe 98] H. Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. Em *VIS '98: Proceedings of the Conference on Visualization '98*, páginas 35–42, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

- [Kim 01] Junho Kim e Seungyong Lee. Truly selective refinement of progressive meshes. Em *Graphics Interface*, volume 1, páginas 101–110, 2001.
- [Kim 06] Junho Kim, Sungyul Choe, e Seungyong Lee. Multiresolution random accessible mesh compression. Em *Computer Graphics Forum*, volume 25, páginas 323–331. Wiley Online Library, 2006.
- [Kobbelt 98] Leif Kobbelt, Swen Campagna, e Hans-Peter Seidel. A general framework for mesh decimation. Em *Graphics interface*, volume 98, páginas 43–50, 1998.
- [Luebke 97] D. Luebke e C. Erikson. View-dependent simplification of arbitrary polygonal environments. Em *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, páginas 199–208, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [Maglo 11] Adrien Maglo, Ian Grimstead, e Céline Hudelot. Cluster-based random accessible and progressive lossless compression of colored triangular meshes for interactive visualization. Em *Computer Graphics International*, volume 11, 2011.
- [Maglo 13] Adrien Maglo, Ian Grimstead, e Céline Hudelot. Pomar: Compression of progressive oriented meshes accessible randomly. *Computers and Graphics*, 37(6):743 – 752, 2013. Shape Modeling International (SMI) Conference 2013.
- [Mangan 99] Alan Mangan e Ross Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, October 1999.
- [Ni 94] Xiujun Ni e M Susan Bloor. Performance evaluation of boundary data structures. *Computer Graphics and Applications, IEEE*, 14(6):66–77, 1994.
- [Pajarola 01] R. Pajarola. Fastmesh: Efficient view-dependent meshing. Em *PG '01: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, página 22, Washington, DC, USA, 2001. IEEE Computer Society.
- [Rodrigues 07] Rui Rodrigues, José Morgado, Frutuoso Silva, e Abel Gomes. A ghost cell-based data structure for multiresolution meshes. Em *Computational Science and Its Applications–ICCSA 2007*, páginas 666–679. Springer, 2007.
- [Rodrigues 15] Rui S.V. Rodrigues, José F.M. Morgado, e Abel J.P. Gomes. A contour-based segmentation algorithm for triangle meshes in 3d space. *Computers & Graphics*, 49(0):24 – 35, 2015.
- [Rossignac 93] J. Rossignac e P. Borrel. Multi-resolution 3D approximations for rendering complex scenes. Em B. Falcidieno e T. Kunii, editores, *Modeling in Computer Graphics: Methods and Applications*, Berlin, 1993. Springer-Verlag.
- [Schröder 00] Peter Schröder, Denis Zorin, T DeRose, DR Forsey, L Kobbelt, M Lounsbery, e J Peters. Subdivision for modeling and animation. *SIGGRAPH 2000 Course Notes*, 2000.
- [Schroeder 92] W. J. Schroeder, J. A. Zarge, e W. E. Lorenson. Decimation of triangle meshes. Em *SIGGRAPH '92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, páginas 65–70, New York, NY, USA, 1992. ACM Press.
- [Shamir 08] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, September 2008.
- [Silva 03] Frutuoso GM Silva e Abel JP Gomes. Adjacency and incidence framework: a data structure for efficient and fast management of multiresolution meshes. Em *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, páginas 159–166. ACM, 2003.
- [Silva 04] Frutuoso GM Silva e Abel JP Gomes. Normal-based simplification algorithm for meshes. Em *Theory and Practice of Computer Graphics, 2004. Proceedings*, páginas 211–218. IEEE, 2004.
- [Yang 04] Sheng Yang, Chang-Su Kim, e CC Jay Kuo. A progressive view-dependent technique for interactive 3-d mesh transmission. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(11):1249–1264, 2004.
- [Rodrigues 07] Rui Rodrigues, José Morgado, Frutuoso Silva, e Abel Gomes. A ghost cell-based data structure for multiresolution meshes. Em *Computational Science and Its Applications–ICCSA 2007*, páginas 666–679. Springer, 2007.