

# Reconstrução de Superfícies Trianguladas a partir de Nuvens de Pontos sem Restrições Angulares

Gonçalo Leitão  
ESTG/IPG  
Guarda  
gleitao@ipg.pt

Abel Gomes  
Instituto de Telecomunicações  
Universidade da Beira Interior  
agomes@di.ubi.pt

---

## Sumário

A reconstrução de uma superfície triangulada a partir de uma nuvem de pontos é um problema difícil de resolver dado depender da forma do objeto original e da densidade dos pontos obtidos pelo dispositivo de aquisição 3D utilizado (scanner). Em grande medida, os algoritmos baseados em expansão da malha que existem na literatura não são bem-sucedidos porque os triângulos são apenas à malha em expansão com base em intervalos angulares que não podem ser ultrapassados. Por contraposição, propõe-se neste artigo um algoritmo que resolve este problema através da conjunção de funções que quantificam três propriedades geométricas fundamentais: coplanaridade, proximidade e regularidade.

## Palavras-chave

Surface reconstruction, 3D point cloud, mesh growing.

---

## 1. INTRODUÇÃO

A reconstrução de superfícies a partir de uma nuvem de pontos é um importante tópico de investigação no domínio da computação gráfica. A sua aplicação em ciências e engenharias é variada; por exemplo, na modelação geométrica, realidade virtual, animação por computador, medicina ou engenharia reversa.

Na literatura encontramos vários métodos para a reconstrução de superfícies, os quais podem ser agrupados em três classes principais: *simpliciais* (ou trianguladas), *paramétricas* e *implícitas*. Os métodos simpliciais podem ainda ser subdivididos em dois grupos: os que são baseados em técnicas de Voronoi/Delaunay e os métodos de proliferação da malha (*mesh growing*). Neste artigo, só estamos interessados em métodos de reconstrução de superfícies trianguladas que seguem uma estratégia de proliferação da malha. Os métodos baseados na proliferação da malha geram uma superfície triangulada a partir de um triângulo inicial, a qual se vai propagando através do apensamento de novos triângulos, os quais terão de satisfazer um ou mais critérios geométricos.

Um desses algoritmos, que é baseado na proliferação da malha, é designado por *ball pivoting algorithm* (BPA) e foi introduzido por Bernardini et al. [Bernardini99]. Este algoritmo utiliza uma esfera assente em cada aresta fronteira da malha, e procura o próximo ponto da nuvem que se situa sobre a esfera de modo que mais nenhum outro ponto se situe dentro da esfera. Os três pontos sobre a esfera formam um possível triângulo, mas para isso é preciso que o ângulo diedral entre o triângulo da fronteira e o novo triângulo seja menor que  $90^\circ$ . Este algoritmo não é robusto nos casos em que a nuvem de pontos tem

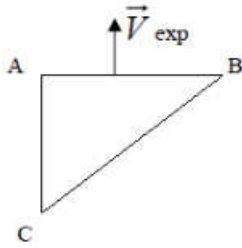
densidade variável, tem ruído ou quando apresenta vincos (*ridges*); estes vincos correspondem a arestas no objeto original.

Li et al. [Li09] propõem um algoritmo de proliferação da malha que utiliza uma função de prioridade, que é dada pela soma conjunta de três grandezas. A primeira é o valor do cosseno do maior ângulo interno do novo triângulo, a segunda grandeza é o cosseno do ângulo diedral entre o triângulo da fronteira e o novo triângulo, e a terceira corresponde à distância da aresta fronteira ao ponto em avaliação. Nesta última grandeza, notar que a distância é normalizada relativamente ao raio de pesquisa usado para encontrar o conjunto dos pontos próximos. A função de prioridade aplica-se aos pontos candidatos para definir uma fila de prioridades de acordo com os respetivos valores. O ponto selecionado será o que tiver a maior prioridade. O algoritmo mostra insuficiências com modelos de formas complexas e impõe restrições às amplitudes dos ângulos internos de qualquer novo triângulo.

Angelo et al. [Angelo11] apresentam um algoritmo para a rápida reconstrução da superfície que se baseia no critério G2S, que é uma aproximação ao critério de Delaunay em 2D. No entanto, o algoritmo não apresenta bons resultados nos casos em que as nuvens de pontos não são uniformes.

O algoritmo proposto por Wongwaen et al. [Wongwaen12] subdivide o cubo que envolve a nuvem de pontos em pequenos cubos do mesmo tamanho, ou seja, de forma regular. De seguida, a triangulação dos pontos efetua-se no interior de cada cubo, sendo depois estas sub-triangulações ou sub-malhas posteriormente interligadas por triangulação intersticial. A progressão

dos triângulos é orientada pelo vetor de proliferação de cada aresta fronteira. Este vetor é perpendicular à aresta fronteira da malha em expansão e é coplanar com o triângulo fronteiro delimitado pela referida aresta, tal como se ilustra na Fig. 1.



**Figura 1: Vetor de proliferação da aresta AB.**

Contudo, a proliferação da malha é condicionada pela imposição de limites às amplitudes dos ângulos internos do novo triângulo que irá ser anexado à malha. Além disso, a divisão uniforme do espaço circundante da nuvem de pontos pode originar triangulação intersticial errada no interior de um cubo. Por vezes, são unidos segmentos não adjacentes da superfície, pela razão de pertencerem ao interior do mesmo cubo em resultado do objecto original apresentar uma variação rápida de curvatura. A partição uniforme do cubo que envolve a nuvem de pontos pode ainda ser inadequada quando a densidade dos pontos não é uniforme, podendo criar furos inexistentes no objeto original.

De modo semelhante, em [Wang13], e de modo a evitar transições drásticas na expansão dos triângulos, é limitada a variação angular relativamente à direção de proliferação. Assim, são colocadas restrições ao ângulo entre a normal do triângulo na fronteira da malha e a normal do novo triângulo. Além disso, e para evitar triângulos demasiado estreitos, são impostos limites às amplitudes dos ângulos internos dos novos triângulos.

Em [Xumin14], com o objetivo de reduzir o tempo e a complexidade, é proposto um algoritmo de reconstrução também baseado na expansão da malha e que utiliza a subdivisão *octree*, sem necessidade de processar o vetor normal em cada ponto. Contudo, o método de reconstrução da superfície também determina restrições à amplitude do ângulo diedral entre o novo triângulo e o triângulo de fronteira da malha, o mesmo acontecendo com os ângulos internos do novo triângulo.

Portanto, em geral, os problemas decorrentes da reconstrução de uma malha triangular relacionam-se com a dificuldade em encontrar critérios adequados que sustentem a construção de triângulos, em conformidade com a forma geométrica da superfície do objeto original. Por exemplo, regiões com curvatura acentuada ou vincada, bem como regiões de forma oscilatória, geram dificuldades na sua triangulação. A triangulação direta de uma nuvem de pontos também pode ser dificultada pela densidade variável (ou não-uniforme) dos seus pontos, em particular quando o objeto original possui furos. Além disso, não existe na literatura qualquer algoritmo baseado na proliferação da malha que construa uma

malha de triângulos tendencialmente regular, ou seja, triângulos tendencialmente equiláteros, a não ser que se use um passo de regularização dos triângulos após a construção da malha.

O algoritmo aqui apresentado, para reconstruir uma superfície triangulada a partir de uma nuvem de pontos, enquadra-se nos métodos de proliferação da malha. De modo diferente, este algoritmo utiliza funções que quantificam propriedades geométricas, tais como a *coplanaridade*, a *proximidade* e a *regularidade*, que são a base para a escolha do próximo triângulo a ser anexado à malha em crescimento. Com a aplicação de tais funções não existe a necessidade de estimar os vetores normais em cada ponto, e evita-se ainda a imposição de restrições angulares, quer à amplitude dos ângulos internos dos triângulos da malha, quer ao ângulo diedral entre triângulos adjacentes. A proliferação da malha vai avançando sempre pelas zonas onde se verifica menor variação de forma, ou seja, onde se verificam condições de maior coplanaridade. Deste modo, as regiões de maior curvatura ou afiadas são reconstruídas seguindo um caminho ou direção mais suave ou mais planar.

## 2. O ALGORITMO

O algoritmo consiste nos seguintes passos principais: (i) cálculo dos pontos vizinhos de cada ponto; (ii) construção do triângulo inicial; (iii) proliferação incremental da malha até à reconstrução completa da superfície. A descrição pormenorizada de cada um dos destes passos faz-se nas subsecções que se seguem. Mas antes disso, vejamos os conceitos fundamentais que sustentam o algoritmo aqui descrito.

### 2.1 Critérios geométricos

Como se referiu acima, há três critérios essenciais que são sintetizados no nosso algoritmo: proximidade, coplanaridade e regularidade.

A *proximidade* tem que ver com o cálculo dos pontos mais próximos e em redor de cada ponto. No algoritmo, iremos também calcular o ponto mais próximo de uma dada aresta. Assim, considerando os dois pontos extremos de uma aresta na fronteira da frente de expansão da malha, diz-se que um terceiro ponto é o ponto mais próximo a essa aresta, se for aquele que tem a menor soma das distâncias aos dois pontos extremos da aresta.

A *coplanaridade* é a propriedade dominante utilizada aquando da escolha do próximo triângulo a ser acrescentado à malha. Para isso, utiliza-se o conceito de vetor de proliferação associado a cada aresta da frente de expansão. Trata-se de um vetor coplanar com o respetivo triângulo da frente de expansão, que é perpendicular à própria aresta e é orientado para o seu exterior (Fig. 1). Este vetor é calculado tal como foi descrito em [Wongwaen12]. Note-se que o vetor de proliferação gera um semiespaço positivo que tem como fronteira o plano perpendicular ao triângulo e contém a aresta considerada na frente de expansão.

A *regularidade* da malha tem que ver com a produção de uma malha de triângulos que são tendencial e desejavelmente equiláteros. A quantificação da regularidade dos triângulos de uma malha faz-se do seguinte modo. Ao observarmos dois triângulos, aquele que geometricamente se aproxima mais de um triângulo equilátero é aquele onde a soma dos cossenos dos três ângulos internos tem maior valor. Este cálculo é efetuado pela aqui designada função de regularização ( $r$ ).

## 2.2 Pontos Vizinhos

O primeiro passo do algoritmo tem como objetivo encontrar para cada ponto uma vizinhança de pontos próximos. Este conjunto de pontos vizinhos resulta da reunião de dois subconjuntos. Assim, dado um ponto A, o primeiro subconjunto de pontos vizinhos é formado pelos  $k$  ( $=6$ ) pontos mais próximos de A. Note-se que o valor de 6 tem que ver com o facto de que numa malha regular cada ponto tem uma vizinhança de 6 triângulos (1 hexágono). Por sua vez, o segundo subconjunto de pontos vizinhos de A é constituído pelos pontos que incluem o ponto A como um dos seus  $k$  pontos mais próximos, mas que não pertencem ao primeiro subconjunto de pontos vizinhos de A. Nem sempre a relação de um ponto com os seus mais próximos é recíproca.

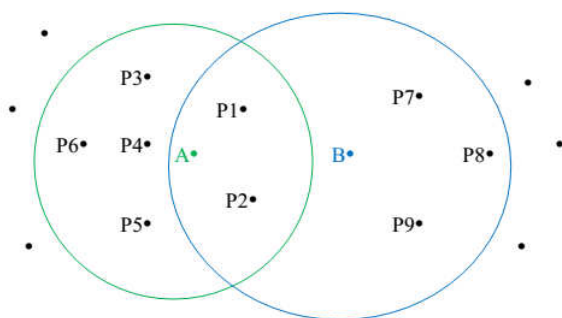


Figura 2: Pontos mais próximos de A e de B.

Por exemplo, na Fig. 2, o conjunto  $\{P1, P2, P3, P4, P5, P6\}$  é o primeiro subconjunto da vizinhança de A, isto é, são os 6 pontos mais próximos de A. Quanto ao ponto B, o primeiro subconjunto da sua vizinhança é  $\{P1, P2, P7, P8, P9, A\}$ , que são os 6 pontos mais próximos de B. Embora o ponto B não seja um dos pontos mais próximos de A, acontece que o ponto A é um dos pontos mais próximos de B. Assim sendo, o ponto B é um dos elementos do segundo subconjunto da vizinhança do ponto A, ou seja,  $V_A = \{P1, P2, P3, P4, P5, P6\} \cup \{\dots, B, \dots\}$ . Notar que os pontos não irão ter as respetivas vizinhanças com igual dimensão. Deste modo, reduz-se o tempo de processamento em relação aos casos em que, *a priori*, se fixa o número de pontos vizinhos a considerar. Além disso, permite que possamos triangular zonas menos densas com outras mais densas.

Para diminuir o nível de complexidade e o tempo de procura para encontrar os pontos vizinhos, utiliza-se a

subdivisão *octree* da *bounding box* que contém a nuvem de pontos. A subdivisão é executada até que os nós folhas contenham um número de pontos inferior a um valor pré-definido, que no nosso caso é de 48 pontos. Note-se que este valor de 48 pontos foi escolhido tendo em conta que numa malha tendencialmente regular cada ponto tem uma vizinhança de 6 pontos, os quais por sua vez têm também uma vizinhança de 6 pontos, o que perfaz um total de 36 pontos, aos quais adicionámos mais 12 pontos por forma a não limitar demasiado o espaço de procura dos pontos. Deste modo, a procura dos pontos vizinhos fica circunscrita aos pontos que se encontram no respetivo nó folha e aos seus nós adjacentes.

Note-se ainda que a escolha deste valor de 48 pontos a utilizar como critério de paragem da subdivisão não tem qualquer influência sobre o resultado da reconstrução da superfície, embora deva ser um mínimo de 36 pontos pela razão referida acima. O objetivo é reduzir, para cada ponto, o domínio de procura dos pontos mais próximos, em vez de ser a totalidade dos pontos da nuvem. Refira-se também que quando consideramos o conjunto de pontos vizinhos de cada ponto, tal como foi descrito anteriormente, e unimos cada ponto a todos os seus vizinhos, gera-se um esqueleto unidimensional que se aproxima bastante do esqueleto final da malha que representa a superfície, como se ilustra na Fig.3.

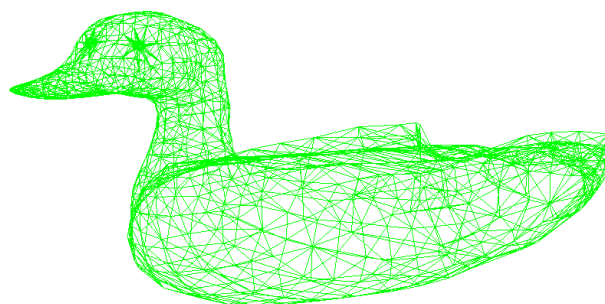
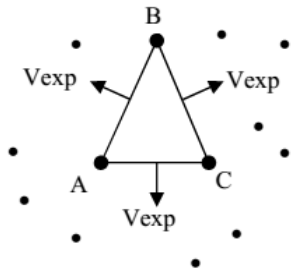


Figura 3: Esqueleto unidimensional gerado por proximidade.

## 2.3 Triângulo Inicial

A proliferação da malha tem início a partir das arestas de fronteira do triângulo inicial, como se ilustra na Fig. 4. Este triângulo deve encontrar-se numa região com relativa suavidade. Além disso, o triângulo inicial não pode ser degenerado (i.e., um triângulo alongado que quase se parece com um segmento de reta). Deve também ter-se em conta que o triângulo deve ser tangencial à superfície curva imaginária que interpola os pontos, pois, caso contrário, a reconstrução da superfície ficará logo comprometida à partida.



**Figura 4: Triângulo inicial.**

O triângulo é construído da seguinte forma. Dado um ponto e o seu vizinho mais próximo forma-se um segmento de reta. No conjunto dos pontos vizinhos de ambos os extremos desse segmento, encontrar o ponto mais próximo a esse segmento de reta. Estes três pontos formam um triângulo candidato a triângulo inicial. Se todos os restantes pontos vizinhos desses três pontos estiverem na união dos semiespaços definidos pelos vetores proliferação de cada aresta, então está encontrado o triângulo inicial (Fig. 4). Isto significa que, o triângulo inicial forma a secção de um prisma triangular onde todos os restantes pontos são exteriores a esse prisma.

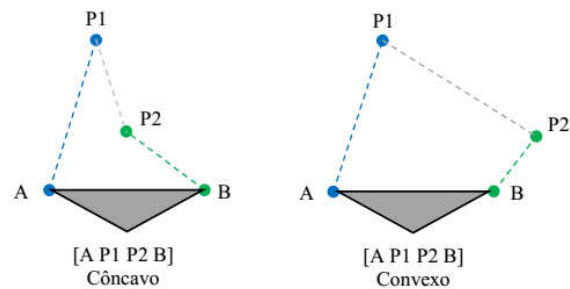
## 2.4 Proliferação da Malha

Depois de construído o triângulo inicial, onde todas as arestas são fronteiras, tem início o processo de crescimento da malha. O princípio orientador do crescimento ou proliferação da malha é que esta avança sempre no sentido para onde se verificar maior coplanaridade, entre o plano do triângulo que suporta a aresta fronteira na frente de expansão e o plano do novo triângulo. Como consequência, a proliferação da malha vai progredindo preferencialmente pelas regiões de menor curvatura. Para cumprir este objetivo, em cada iteração do algoritmo são estudadas todas as arestas fronteiras, aplicando a cada uma delas um conjunto de critérios que identifica o triângulo candidato à anexação nessa aresta. Por fim, de entre os triângulos candidatos, o novo triângulo a ser construído será aquele que apresentar maior grau de coplanaridade com um dos triângulos da frente de expansão.

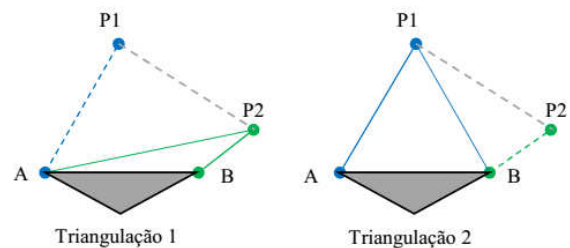
Note-se que, em [Wongwaen12] ou [Xumin14], a lista de arestas fronteiras é percorrida apenas até ser encontrada uma aresta onde o triângulo candidato verifica os critérios pré-definidos. Assim, inibe-se o conhecimento global da forma ao longo de toda a fronteira. Por outro lado, os critérios aplicados impõem limites aos valores dos ângulos internos do novo triângulo adjacente à aresta fronteira em estudo. No nosso caso, os critérios usados são funções que quantificam propriedades geométricas dos triângulos. A conjunção dos valores dessas funções é determinante na escolha do novo triângulo a ser construído em cada iteração.

O estudo aplicado a cada aresta fronteira para encontrar o triângulo candidato à anexação segue as seguintes condições:

- No conjunto dos pontos vizinhos do primeiro ponto extremo da aresta fronteira, encontrar o ponto mais próximo a essa aresta e que se localiza no semiespaço positivo definido pelo vetor de proliferação.
- Repetir o passo anterior para o segundo ponto extremo da aresta fronteira. Um destes dois pontos irá fechar o triângulo procurado e que será adjacente à aresta fronteira.
- Se os dois pontos obtidos coincidirem, então o triângulo candidato está encontrado.
- Caso os dois pontos encontrados não coincidam, então forma-se um quadrilátero, que poderá ser côncavo ou convexo, tal como se ilustra na Fig. 5.



**Figura 5: Classificação dos quadriláteros.**



**Figura 6: Triangulações possíveis do quadrilátero convexo.**

- Caso o quadrilátero seja côncavo, o ponto mais próximo da aresta será o escolhido. Na Fig. 5, à esquerda, o ponto escolhido será o ponto P2.
- Caso o quadrilátero seja convexo são possíveis duas triangulações (Fig. 6). Neste caso, o ponto escolhido pode não ser o mais próximo da aresta, mas aquele que provocar uma subdivisão do quadrilátero de forma mais equilibrada.

Por exemplo, na Fig. 6, a triangulação que será escolhida será a segunda. Para selecionar a triangulação, usa-se a função soma dos cossenos dos três ângulos internos de um triângulo, já acima designada por função de regularização ( $r$ ), e aplica-se a cada um dos triângulos resultantes da subdivisão do quadrilátero. O princípio seguido é o de maximizar a soma dos dois resultados e minimizar a sua diferença. Este princípio pode traduzir-se pela expressão a seguir:

$$r(A,P1,B) + r(B,P1,P2) - |r(A,P1,B) - r(B,P1,P2)|$$

No final de cada iteração, quando todas as arestas fronteiras da frente de expansão tiverem sido analisadas, o novo triângulo a construir será aquele que entre todos os triângulos candidatos resulte no maior grau de coplanaridade com o triângulo adjacente à aresta fronteira.

O novo triângulo dá origem a três casos quanto ao número de arestas novas a construir, como se ilustra na Fig. 7. Estes casos geram, respetivamente, 2, 1, ou 0 novas arestas. O primeiro caso, no qual se acrescenta duas arestas, acontece quando os vértices terminais da aresta da frente de expansão são ambos convexos. O segundo caso, em que se acrescenta uma aresta, surge quando um dos vértices da aresta da frente de expansão é côncavo e o outro é convexo. O terceiro caso dá-se quando ambos os vértices terminais são côncavos.

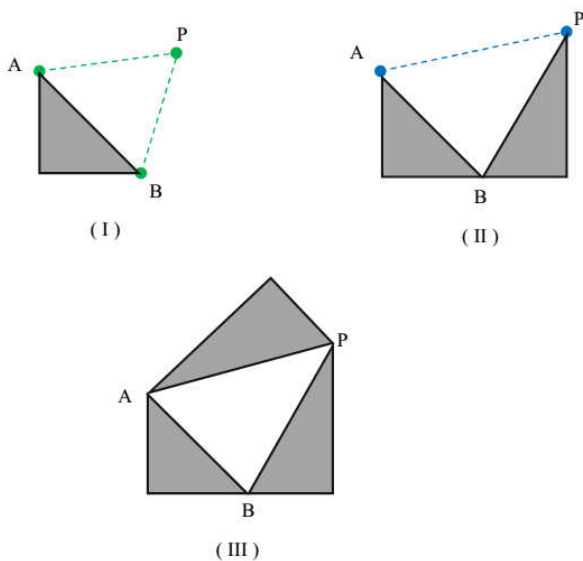


Figura 7: Topologias da reconstrução.

Obviamente, a reconstrução da malha termina quando já não houver arestas fronteiras.

### 3. RESULTADOS

#### 3.1 Caracterização do ambiente de trabalho

O algoritmo proposto foi implementado em C++ com *OpenGL*, sobre *Microsoft Visual Studio 2010* em ambiente *Windows 8*. Os testes foram realizados em computador com processador *Pentium M* de 1.73 Ghz e 2 GB de *RAM*.

#### 3.2 Modelos geométricos

Os modelos utilizados têm formato *obj* ou *pwn* e foram obtidos nos endereços, <http://tf3dm.com/3d-models/all> e [http://home.eps.hw.ac.uk/~ab226/software/mpu\\_implicit/mpu\\_input\\_data.zip](http://home.eps.hw.ac.uk/~ab226/software/mpu_implicit/mpu_input_data.zip), e ainda no conjunto de modelos incluídos na aplicação *MeshLab v1.3.1*.

#### 3.3 Avaliação qualitativa

A avaliação qualitativa do algoritmo pode ser efetuada com base nos problemas que foram sendo constatados na literatura, nomeadamente: características topológicas (e.g., furos), densidade não uniforme dos pontos, zonas vincadas, e zonas de curvatura acentuada ou com rápida variação de forma.

O critério da proximidade na computação localizada dos pontos mais próximos de cada ponto faz com que a reconstrução seja bem-sucedida mesmo em casos em que o objeto original tem características topológicas difíceis de reconstruir, como são o caso dos furos (Fig. 9(b) e Fig. 9(d)).

Note-se que a conjugação dos três critérios (i.e., proximidade, coplanaridade e regularidade) também resolve a questão da densidade não uniforme dos pontos, como é o caso da região do olho do pato representado na Fig. 8. Na verdade, pode observar-se que a região do olho tem uma densidade de pontos muito superior ao restante. Este caso mostra também que o algoritmo é bem sucedido em zonas vincadas. Repare-se que existe um vinco à volta do olho, algo que grande parte dos algoritmos não consegue reconstruir corretamente.

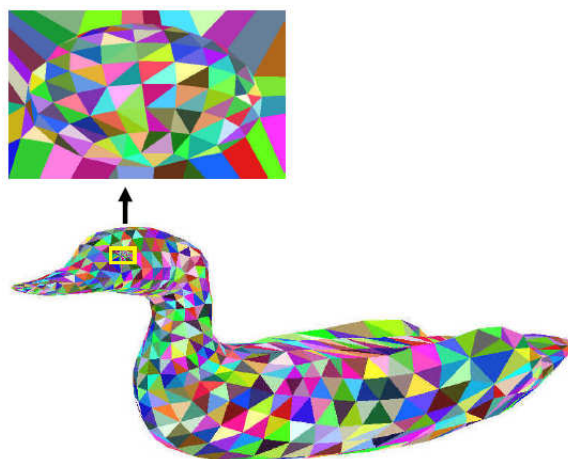


Figura 8: Malha com detalhe do olho.

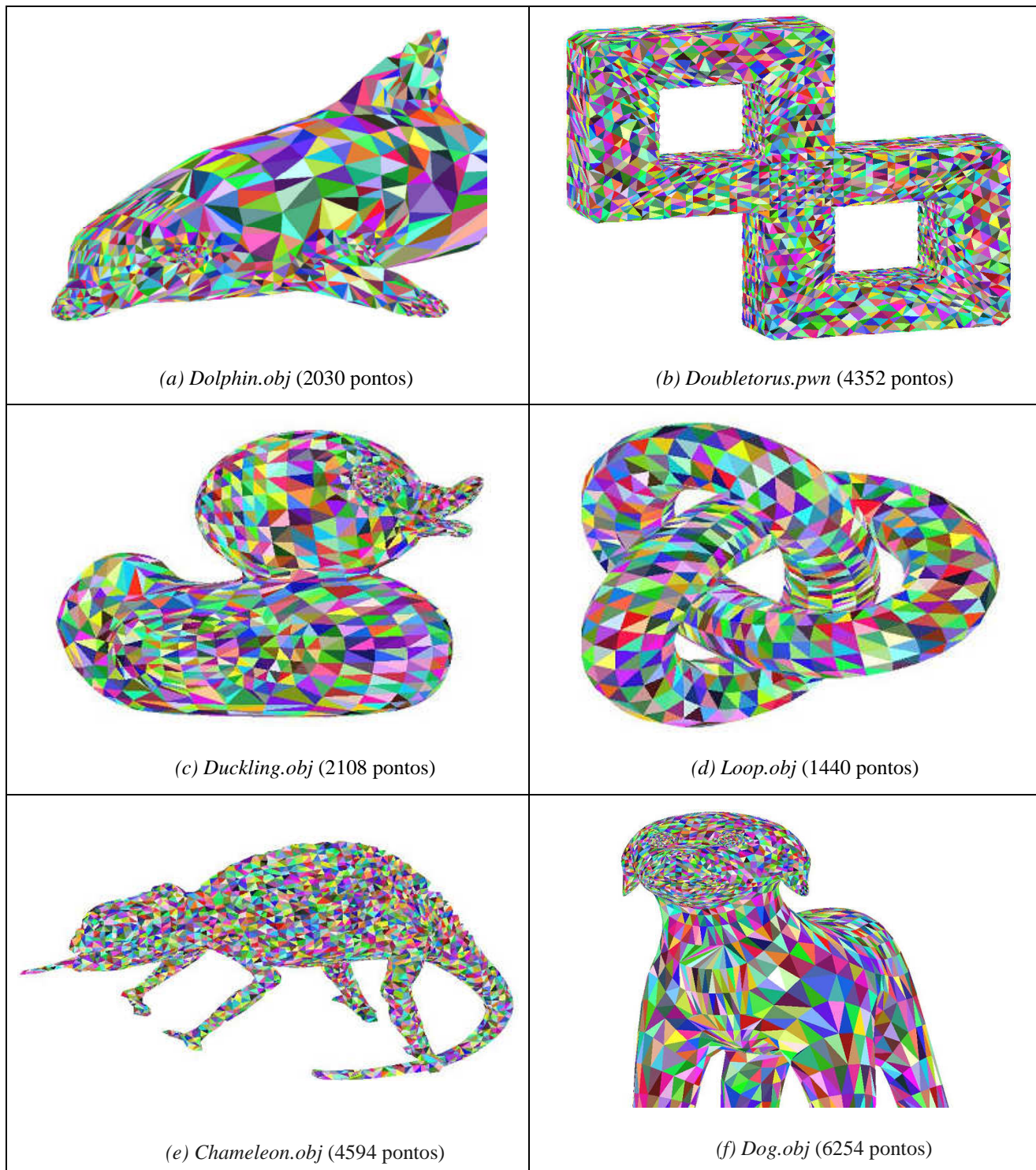
As zonas vincadas são bem reconstruídas por causa da satisfação do critério da coplanaridade. Da mesma maneira, a prioridade dada ao critério da coplanaridade resolve o problema das zonas com curvatura acentuada ou com rápida variação da forma, como são os casos das barbatanas do golfinho na Fig. 9(a) ou o bico do pato na Fig. 9(c). No caso do golfinho, durante a reconstrução, a expansão da malha vai progredindo de ambos os lados da barbatana até às zonas de maior curvatura da barbatana. Só após as zonas mais complanares da barbatana estarem reconstruídas, é que o algoritmo avança para as zonas menos complanares (i.e., com mais curvatura). O mesmo se observa no caso do bico do pato.

Portanto, a coplanaridade é o critério chave que torna o algoritmo simples, eficiente, correto e robusto. Repare-se que, o algoritmo impõe que cada aresta só poderá ter no máximo dois triângulos incidentes, pelo que, logo à

partida está garantida a correção topológica da malha reconstruída.

Refira-se também que a regularidade dos triângulos faz com que a malha final tenha à partida um número menor de triângulos relativamente a outros algoritmos existentes na literatura. Por exemplo, o modelo da Fig. 9(d) foi

gerado com 2880 triângulos, o que acontece também com a aplicação *MeshLab*, mas o algoritmo proposto em [Xumin14] gera uma malha com 3000 triângulos para a mesma nuvem de pontos.



**Figura 9: Malhas reconstruídas a partir de nuvens de pontos.**

#### 4. CONCLUSÃO

Foi proposto um algoritmo para reconstrução de malhas triangulares a partir de nuvens de pontos que se enquadra nos métodos de proliferação da malha (*mesh growing*). O método proposto resolve o problema das restrições angulares, quer dos ângulos internos dos triângulos quer da direção de expansão. O problema foi resolvido pela aplicação de funções que quantificam propriedades geométricas como a coplanaridade, regularidade e proximidade. Mais ainda, a reconstrução avança após o conhecimento global ao longo da fronteira e não apenas localmente. Os resultados mostram o bom funcionamento do algoritmo, mesmo em casos onde usualmente existem dificuldades devido à geometria ou topologia.

#### 5. REFERÊNCIAS

- [Angelo11] L. D. Angelo, P. H. Stefano, L. Giaccari. A new mesh-growing algorithm for fast surface reconstruction. *Computer Aided Design*, Vol. 43, no. 6, 639-650, 2011.
- [Bernardini99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin. The ball pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, no. 4, 349-359, 1999.
- [Li09] X. K. Li, C. Y. Han, W. G. Wee. On surface reconstruction: A priority driven approach. *Computer Aided Design*, Vol. 41, no. 9, 626-640, 2009.
- [Wang13] N.Wang, Q. Zhang, D. Zhou, X. Wei. Local optimum triangulation for unorganized point cloud. *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 6, no.10, 1862-1867, 2013.
- [Wongwaen12] N.Wongwaen, S. Tiendee, C. Sinthanayothin. Method of 3D mesh reconstruction from point cloud using elementary vector and geometry analysis. *International Conference on Information Science and Digital Content Technology (ICIDT 2012)*, Vol. 1, 156-159, June 26-28, Jeju, South Korea, 2012.
- [Xumin14] Liu Wongwaen, Yang Lixin, Li Cailing. A robust mesh growing surface reconstruction algorithm based on octree. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 7, No. 3, 135-146, 2014.

Este trabalho encontra-se ainda em desenvolvimento, motivo pelo qual não são apresentadas comparações quantitativas com outros métodos segundo alguma métrica. Este será um dos objetivos do trabalho futuro, assim como a otimização computacional de alguns procedimentos para tornar o algoritmo ainda mais eficiente e robusto, em particular em modelos de maior tamanho em termos do número de triângulos, bem como avaliar a sua aplicação com outros tipos de geometrias.