

# MODELAGEM DE TERRENOS NATURAIS ATRAVÉS DE MALHAS DE TRIÂNGULOS

ANDERSON SANGREMAN\*, GUSTAVO FREITAS\*, RAMON R. COSTA\*

\*Programa de Engenharia Elétrica, COPPE/UFRJ

Emails: sangreman@poli.ufrj.br, gfreitas@coep.ufrj.br, ramon@coep.ufrj.br

**Abstract**— In this paper we present a method for modeling natural terrain covered by vegetation. The model construction is made from triangulation of a 3D point cloud, in this case obtained by a laser, producing a  $2\frac{1}{2}$  D surface. Tests performed in natural terrain illustrate the performance of the method. The obtained maps represent the terrain profile in detail according to the defined resolution, which should be chosen taking into account a tradeoff between level of detail and computational processing power required.

**Keywords**— Digital Terrain Modeling (DTM), Delaunay Triangulation, Field Robotics

**Resumo**— Nesse trabalho é apresentado um método de modelagem de terrenos naturais cobertos por vegetação. A construção do modelo é feita a partir da triangulação de uma nuvem de pontos 3D, neste caso obtidos por um laser, gerando uma superfície  $2\frac{1}{2}$  D. Testes realizados em um ambiente natural ilustram o funcionamento do método proposto através dos resultados obtidos. Os mapas gerados representam o terreno com detalhes, de acordo com a resolução definida, que deve ser escolhida levando em conta o compromisso entre nível de detalhamento e processamento computacional exigido.

**Palavras-chave**— Modelagem Digital de Terreno (MDT), Triangulação de Delaunay, Robótica de Campo

## 1 Introdução

A modelagem de um terreno é uma forma de representar a distribuição da elevação da sua superfície (Plagemann et al., 2009), representando infinitos pontos por um modelo finito.

Essa modelagem é crucial em áreas como engenharias cartográfica, civil, ambiental e militar, fornecendo informações indispensáveis para, por exemplo, viabilizar o planejamento urbano (projetos de construção de estradas, túneis, barragens e dimensionamento dos respectivos deslocamentos de terras), extração de curvas de níveis e redes de drenagem, estudos de impacto ambiental como o cálculo de áreas alagadas na construção de hidroelétricas (M. A. Piteri, 2007), pesquisas geológicas, treinamentos em simulações de voo e robótica móvel. Para mais aplicações, ver (Barbosa et al., 2003).

Terrenos naturais (em inglês também denominados *outdoor*) podem apresentar parte da superfície coberta por vegetação, pedras e outros obstáculos. Estes terrenos tendem a compor superfícies irregulares, cuja representação através de figuras geométricas básicas não é trivial. Nestes casos a modelagem se torna complexa e outros problemas surgem à tona, como capacidade de memória e processamento computacional.

A representação de terrenos e a construção de mapas são tarefas centrais na robótica móvel, e podem ser utilizadas para planejamento e simulação de trajetória, identificação de pontos de contato, análise de estabilidade, entre outras. Estas tarefas requerem especial atenção no caso de operações em ambientes naturais (Hamner et al., 2008), lidando com riscos de tombamento causado pela estrutura irregular do terreno e colisão com objetos oclusos por vegetação.

Para operar de maneira autônoma em terrenos naturais, o robô necessita de uma representação do meio de operação. Um exemplo é apresentado em (Batavia et al., 2002), onde uma estratégia para navegação autônoma em ambientes *outdoor* é proposta

com base no modelo do ambiente previamente construído. Além disso, quanto maior for a velocidade do veículo, mais precisa deve ser a representação do terreno (Brotten and Collier, 2006).

Um sensor usualmente utilizado para perceber o ambiente é o *laser range finder*, que apresenta uma boa razão entre eficiência e custo com respeito a outros sensores. Além do laser, merecem destaque outros dispositivos de instrumentação: sensores de ultrassom e câmeras. O sensor de ultrassom possui baixo custo; porém seu alcance, tempo de resposta e precisão são consideravelmente inferiores ao laser. Já a câmera é um sensor passivo, e suas medidas são altamente sensíveis a variações na iluminação do ambiente. Além disso, o processamento de imagens pode exigir alta capacidade computacional.

O laser pode ser empregado para varrer um determinado terreno e assim gerar uma nuvem de pontos no espaço tridimensional. A representação do terreno por pontos desconexos dificulta a visualização da superfície percorrida, e não oferece nenhuma informação adicional de natureza topológica (incidência, adjacência, conectividade), propriedades de natureza geométrica (declividade e orientação do terreno) e outras características que auxiliam nas diferentes aplicações de um modelo. Dessa forma, deve ser realizada algum tipo de aproximação permitindo a criação de uma superfície contínua no espaço 3D.

Uma solução comum para a representação de terrenos naturais é o  $2\frac{1}{2}$  D *grid map* (Brotten and Collier, 2006; Hebert and Krotkov, 1993). Um mapa  $2\frac{1}{2}$  D pode ser gerado pela regra de formação  $z = f(x, y)$ , onde  $z$  é a elevação do terreno. Representações de terrenos através de mapas  $2\frac{1}{2}$  D são denominadas de Modelo Digital de Terreno (MDT), Mapa de Elevação Digital ou simplesmente Mapa de Terreno. Uma análise mais completa dos métodos usados para a modelagem de terrenos é apresentada em (Hugentobler, 2004).

No caso da robótica móvel, a formação de uma superfície contínua a partir da nuvem de pontos 3D

apresenta diversas funcionalidades. A superfície pode ser empregada na identificação de pontos de contato entre as rodas do robô e o terreno, permitindo a análise de estabilidade do mesmo. O modelo também é útil em tomadas de decisão relativas à navegação, evitando regiões com presença de vegetação densa, ou ainda identificando objetos que se destacam no ambiente, por exemplo uma árvore, utilizados como referência para localização (Siegwart and Nourbakhsh, 2004).

Uma superfície pode ser aproximada por um conjunto de polígonos adjacentes e, como geometricamente um plano pode ser definido por três pontos, o polígono mais usual é o triângulo. Assim, um dos métodos mais utilizados na modelagem de terrenos a partir da amostragem de pontos é a triangulação (Figura 1).

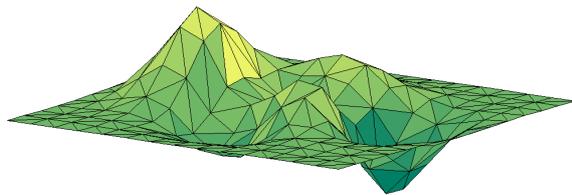


Figura 1: Exemplo de uma superfície triangulada.

A criação de uma malha de triângulos não é um processo único, e permite soluções distintas. Uma das formas de triangulação mais utilizadas é a de Delaunay, devido à sua propriedade de maximizar o menor ângulo de cada triângulo; assim o método garante uma malha triangular o mais equiangular possível (Sibson, 1978).

O objetivo desse trabalho é apresentar uma metodologia para modelagem de terrenos naturais com base apenas em informações geométricas do ambiente. Para tal, são utilizados dados coletados por um veículo autônomo equipado com *laser range finder* e sensor de posicionamento de alta precisão, percorrendo terrenos naturais de forma a gerar uma nuvem de pontos tridimensional. Após o devido tratamento dos dados, é realizada uma filtragem para excluir do modelo arbustos e grama alta. A modelagem é realizada através da triangulação de Delaunay, gerando uma superfície  $2 \frac{1}{2}$  D. Essa modelagem é atualmente implementada em um processo *offline* utilizando o software *Matlab*.

Esse artigo é organizado da seguinte maneira: na seção 2 é apresentada uma revisão bibliográfica sobre modelagem de terrenos por triangulação de pontos, com ênfase na robótica móvel. Na seção 3 são apresentados detalhes sobre método de triangulação adotado. Na seção 4 é formulado o método da modelagem de terrenos utilizado e na seção 5 são apresentados resultados experimentais. O artigo é concluído na seção 6, onde são propostas sugestões de trabalhos futuros.

## 2 Revisão Bibliográfica

A reconstrução de um terreno por triangulação de pontos não é uma ideia recente. Malhas triangulares têm sido aplicadas na modelagem de terrenos naturais para reduzir a informação geométrica disponível à sua forma mais básica, como na aerofotogrametria.

Uma vantagem em utilizar malhas triangulares é a capacidade de alterar a resolução do modelo de acordo

com a complexidade do terreno, conforme descrito em (Bakambu et al., 2006).

A triangulação deve ser realizada de forma a obter uma representação mais próxima possível do terreno original. Foi observado que triangulações que levam a boas interpolações tendem a evitar triângulos longos e finos (Barnhill, 1977), e por isso a técnica de Delaunay corresponde a uma eficiente forma de representação, dada a propriedade de maximizar o menor ângulo interno dos triângulos.

Uma aplicação do método é apresentada em (Wettergreen et al., 2010), onde um robô móvel com laser computa uma malha de triângulos para criar um modelo 3D do terreno a ser percorrido (Figura 2). A superfície é obtida através da triangulação de Delaunay 2D e o modelo é utilizado para duas tarefas: planejamento de trajetória e ajuste de configuração do robô.

O planejamento de trajetória é obtido empregando um algoritmo de minimização de custo; essa trajetória é posteriormente suavizada visando a uma maior eficiência energética e menor tempo de percurso. No ajuste de configuração do robô, é feita uma estimativa da interação entre robô e terreno, ajustando a configuração do robô para evitar o capotamento devido a irregularidades do terreno percorrido.



Figura 2: Robô SCARAB modelando o terreno através da triangulação de Delaunay (Wettergreen et al., 2010). O robô possui um laser range finder e constrói um modelo em tempo real.

No trabalho de (Gerbaud et al., 2004) é descrita uma plataforma móvel capaz de realizar simultaneamente a exploração e a construção do mapa ao seu redor. O ambiente é percebido por lasers e o conjunto de pontos medidos é projetado em um plano para a realização da triangulação de Delaunay 2D.

## 3 Triangulação

A triangulação de um conjunto de pontos estabelece uma conexão entre objetos de dimensão 0 (os pontos) formando objetos de dimensão 1 e 2 (as arestas e as faces triangulares). Dessa forma, é criada uma estrutura topológica onde é possível obter informações úteis sobre o terreno, representado aqui por uma superfície  $2 \frac{1}{2}$  D. Uma forma de garantir que a triangulação de pontos gere uma superfície desse tipo é realizar uma triangulação de Delaunay 2D, utilizando as projeções dos pontos originais em uma plano horizontal. Após a triangulação, as projeções dos pontos são elevados às suas alturas originais (Figura 3).

A triangulação de Delaunay 2D não garante que o triângulo obtido, quando reprojeto no espaço tridimensional, seja necessariamente o mais equiangular possível. Ainda assim, esta variação do método possui consideráveis vantagens em comparação à triangulação tridimensional.

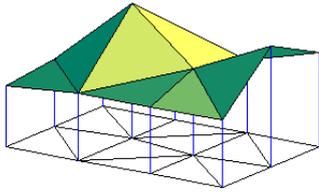


Figura 3: Triangulação gerando uma superfície  $2\frac{1}{2}$  D.

Segundo (Rippa, 1990), a triangulação 2D minimiza a rugosidade do terreno resultante, independentemente da altura dos dados. Esta pode ser uma característica desejada durante a interpolação de pontos. Outras vantagens são a simplicidade de implementação, menor processamento computacional associado e maior literatura disponível. De fato, os trabalhos destacados na Seção 2 empregaram a triangulação de Delaunay 2D.

Um método alternativo é apresentado em (Dyn et al., 1990), combinando triangulação 2D com otimização de uma função de custo calculada levando em conta a altura de cada ponto. Uma abordagem semelhante é utilizada em (Quak and Schumaker, 1990), podendo resultar em representações mais realistas com respeito à triangulação de Delaunay no caso de superfícies suaves.

### 3.1 Propriedades da Triangulação de Delaunay

Além de maximizar o menor ângulo interno, a triangulação de Delaunay apresenta outras características fundamentais (Lee and Schachter, 1980):

- Se três pontos são vértices de um mesmo triângulo, o círculo que passa por esses três pontos não contém nenhum outro ponto em seu interior.
- Se dois pontos formam uma aresta, existe um círculo que passa por esses dois pontos e não contém mais nenhum ponto.

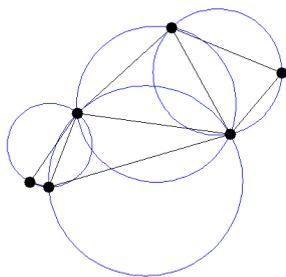


Figura 4: Triangulação de Delaunay de 6 pontos no plano. Observe que o círculo circunscrito a cada triângulo não contém nenhum outro ponto.

Essas propriedades são utilizadas na construção da malha triangular (Figura 4). Triângulos e arestas que não obedeçam às características fundamentais são chamadas de ilegais. Caso uma triangulação possua somente triângulos e arestas legais, trata-se de uma triangulação de Delaunay.

A triangulação de Delaunay não é necessariamente única. Caso no conjunto de pontos existam

quatro ou mais pontos pertencentes a um mesmo círculo, mais de uma triangulação pode ser obtida. Caso contrário, a solução é única.

### 3.2 Construção da Triangulação de Delaunay

Um quadrilátero convexo pode ser triangulado de duas maneiras invertendo-se a aresta interior conforme ilustrado na Figura 5.

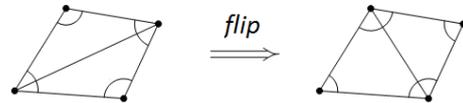


Figura 5: Flip de arestas em um quadrilátero.

A triangulação apresentada à direita é obtida utilizando a técnica de Delaunay. É possível perceber que ela maximiza o menor ângulo dos triângulos. A inversão de arestas é chamada em inglês de *flip*. Uma implementação simples da técnica parte de uma triangulação qualquer dos pontos. Em seguida realiza-se o *flip* de arestas ilegais até que só existam arestas legais. Contudo, esta abordagem é pouco eficiente.

Métodos para a construção da triangulação de Delaunay já foram amplamente divulgados na literatura e uma grande quantidade de algoritmos para sua obtenção estão disponíveis. O método utilizado nesse trabalho para a triangulação foi o algoritmo incremental aleatório (*randomized incremental algorithm*) conforme proposto em (Guibas et al., 1992). Esse é um algoritmo eficiente e sua implementação é simples, com convergência na ordem de  $n \log(n)$ , sendo  $n$  o número de pontos.

A modelagem começa pela criação de um primeiro triângulo auxiliar que irá englobar todos os pontos do modelo. Os pontos são ordenados de forma aleatória e adicionados ao triângulo auxiliar um de cada vez, gerando novos triângulos. Quando um ponto é adicionado, primeiramente é identificado a qual triângulo ele pertence. O novo ponto é ligado aos vértices do triângulo em que está contido, criando três novos triângulos. Em seguida, é verificado se as arestas dos novos triângulos são legais. Em caso negativo, é realizado um *flip* e as arestas envolvidas são novamente verificadas. Quando restarem apenas arestas legais, o próximo ponto é adicionado. O processo é repetido até que não existam mais pontos a se adicionar, quando então é concluída a triangulação.

O algoritmo implementado nesse trabalho utiliza uma estrutura de armazenamento de dados do tipo árvore, conforme sugerido em (De Berg et al., 2008), composta de um elemento principal chamado *raiz* ligado a outros elementos chamados de *ramos*. Cada ramo pode levar a outros elementos, gerando novos ramos, até que atinja um elemento que não se ramifica denominado de *folha*. Dessa forma, é armazenado o histórico de criação dos triângulos e dos *flips* realizados, de maneira que o elemento da árvore que armazena cada triângulo também armazena um indicador para os triângulos que o substituíram. Para identificar o triângulo que contém um determinado ponto, parte-se da raiz da estrutura proposta, indo para os níveis seguintes até chegar ao último nível (folha) correspondente.

#### 4 Metodologia

Este artigo propõe uma metodologia de modelagem de terreno através de uma nuvem de pontos tridimensional. Nesta aplicação, a nuvem de pontos foi obtida por um veículo utilizando um sistema de posicionamento de alta precisão e um *laser range finder* fornecendo medições transversais do terreno diretamente à frente do veículo.

A modelagem é feita em quatro etapas (Figura 6): tratamento inicial dos dados, transformação de coordenadas dos pontos do laser para um referencial inercial, filtragem da vegetação e triangulação dos pontos.

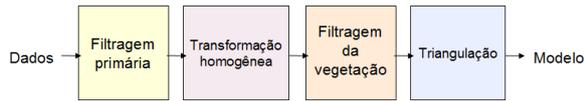


Figura 6: Diagrama da metodologia proposta.

##### 4.1 Tratamento Inicial dos Dados

O primeiro passo consiste em filtrar as medições obtidas pela instrumentação embarcada, no caso um sensor de posição e um *laser range finder*. No presente caso, foi utilizado um sensor Applanix POS 220 LV INS/GPS que, devido à sua acurácia na medida de pose, induz pouco erro de modelagem e permite dispensar o uso de filtros.

O principal ruído do sensor laser é conhecido em inglês como *salt and pepper* (Lingemann et al., 2005), e consiste em medidas que não acompanham a geometria da superfície local (Sotoodeh, 2006). Ruídos desse tipo são causados por limites de oclusão, refletância de superfícies e múltiplos caminhos de reflexão. Para eliminar tais ruídos, foi adotada uma técnica de identificação e rejeição de *outliers* análoga à aplicada em (Underwood et al., 2007).

##### 4.2 Transformação Homogênea

O laser percebe pontos do terreno em frente ao veículo, fornecendo as distâncias e seus respectivos ângulos de medição no plano de visão do sensor. Contudo, os pontos devem ser registrados com respeito a um sistema de coordenadas inercial, tornando necessária transformações de coordenadas dos dados. Para isso, são definidos três sistemas de coordenadas (Figura 7): um sistema de coordenadas coincidente com o plano de medição do laser com origem  $O_L$ , um sistema de coordenadas do veículo cuja origem  $O_V$  está localizada entre as rodas traseiras, e um sistema de coordenadas inercial com origem  $O_I$  localizado no ponto de partida.

O processo de registro consiste em 2 transformações homogêneas. A primeira transformada leva os pontos do sistema de coordenadas do laser para o sistema de coordenadas do veículo e a segunda leva do sistema do veículo para o sistema de coordenadas inercial. A primeira matriz de transformação é definida pela posição e orientação do laser com respeito à origem do sistema de coordenadas do veículo, obtida através do processo de calibração descrito em (Freitas et al., 2012). A segunda matriz de transformação é

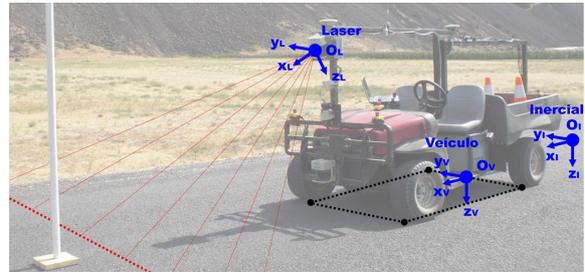


Figura 7: Sistemas de coordenadas utilizados na transformação homogênea.

dada pela translação e rotação do veículo com respeito à origem do sistema de coordenadas inercial, obtida diretamente através do sensor Applanix.

##### 4.3 Filtragem da Vegetação

O objetivo da filtragem de vegetação consiste em eliminar arbustos e grama alta, equivalentes a volumes porosos representados na nuvem de pontos 3D. O método de filtragem adotado é o proposto em (Vandapel et al., 2006). Para tal, um cone auxiliar é criado a partir de cada ponto da nuvem em direção ao terreno, conforme ilustrado na Figura 8. Caso exista algum outro ponto no interior do tronco de cone definido pelos parâmetros  $\theta$  e  $\rho$ , o ponto no topo é classificado como vegetação.

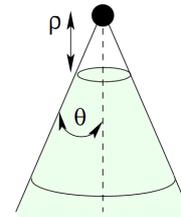


Figura 8: Parâmetros do algoritmo de classificação de vegetação.

##### 4.4 Triangulação

O conjunto de pontos filtrados é usado como entrada no processo de triangulação. Para isso são consideradas apenas as coordenadas  $x$  e  $y$  das medições, realizando uma triangulação planar. O pseudocódigo 4.1 descreve a implementação da triangulação de Delaunay através do algoritmo incremental aleatório.

Após concluída a triangulação no plano, a componente  $z$  é novamente agregada a cada ponto, preservando as conexões estabelecidas entre eles. Durante o processo, os triângulos são armazenados numa estrutura de dados do tipo árvore. Quando um triângulo final (folha) dá origem a novos triângulos, este não é apagado da memória, e sim reclassificado como ramo. Esta estrutura é útil para a aproximação de um ponto não amostrado a partir da triangulação, onde é necessário obter o triângulo folha que contém o ponto estimado. Uma forma eficiente de busca parte do triângulo raiz e segue nos ramos correspondentes na árvore, até que se chegue a uma folha equivalente ao triângulo final desejado.

**Algorithm 4.1:** TRIANGULAÇÃO DE DELAUNAY(*Pontos*)

```

procedure LEGALIZARARESTA( $p_r, p_i, p_j, T$ )
  if aresta( $p_i, p_j$ ) é ilegal
    then  $\begin{cases} Find(p_k) \\ Replace(aresta(p_i, p_j), aresta(p_r, p_k)) \\ LegalizarAresta(p_r, p_i, p_k, T) \\ LegalizarAresta(p_r, p_k, p_j, T) \end{cases}$ 
main
  CreateBigTriangle(Pontos)
  RandomSort(Pontos)
  for  $r \leftarrow 1$  to  $n$ 
     $T(p_i, p_j, p_l) \leftarrow FindTriangle(p_r)$ 
    do  $\begin{cases} Insert(p_r, T) \\ LegalizarAresta(p_r, p_i, p_j, T) \\ LegalizarAresta(p_r, p_i, p_l, T) \\ LegalizarAresta(p_r, p_j, p_l, T) \end{cases}$ 

  DeleteBigTriangle
  return (MalhaDeTriangulos)

```

Vale comentar algumas funções presentes no algoritmo:

- *LegalizarAresta*( $p_r, p_i, p_j, T$ ): O ponto sendo inserido é  $p_r$ , e a aresta( $p_i, p_j$ ) pertence ao triângulo  $T(p_r, p_i, p_j)$  que pode precisar da realização de um *flip* para tornar-se uma aresta legal. A aresta( $p_i, p_j$ ) é considerada ilegal caso o círculo circunscrito ao triângulo  $T(p_r, p_i, p_j)$  contenha algum outro ponto;
- *Find*( $p_k$ ): encontra  $p_k$  (caso exista) que forma o triângulo  $T(p_i, p_j, p_k)$ , que é adjacente ao triângulo  $T(p_r, p_i, p_j)$  e compartilha com esse a aresta( $p_i, p_j$ );
- *Replace*(aresta( $p_i, p_j$ ), aresta( $p_r, p_k$ )): Substitui a primeira aresta pela segunda (*flip*);
- *CreateBigTriangle*(*Pontos*): Cria um triângulo auxiliar que envolve todos os pontos;
- *FindTriangle*( $p_r$ ): Encontra o triângulo  $T(p_i, p_j, p_l)$  que contém  $p_r$ ;
- *Insert*( $p_r, T$ ): Insere  $p_r$  em  $T$ , criando 3 triângulos ao conectar  $p_r$  a  $p_i, p_j$  e  $p_l$ ;
- *DeleteBigTriangle*: Apaga o triângulo auxiliar, seus 3 pontos e as arestas ligadas a eles.

### 5 Validação Experimental

A metodologia proposta é validada utilizando dados de campo obtidos com um veículo agrícola autônomo (Figura 7) desenvolvido pela *Carnegie Mellon University* durante o projeto *Comprehensive Automation for Specialty Crops - CASC*.

A plataforma móvel é equipada com um sistema de localização de precisão Applanix POS 220 LV INS/GPS com precisão de 3 cm para posição e  $0.05^\circ$  para orientação, e também um *laser range finder* Sick LMS 291. O laser possui um ângulo de visão de  $180^\circ$  com resolução de  $1^\circ$  e alcance máximo de leitura de 80 m, com precisão de 3.5 cm. Este sensor está instalado na parte superior frontal do veículo, inclinado em aproximadamente  $-20^\circ$  com respeito a horizontal de forma a medir o terreno 4 m a frente.

A coleta de dados foi realizada no pomar Soergel (PA-EUA) em 2011, quando o veículo percorreu aproximadamente 100 m de uma estrada de terra irregular parcialmente coberta por grama (Figura 9), numa velocidade média de 1 m/s.

Seguindo a metodologia proposta, as medições do laser passaram por remoção de *outliers*, registro com

respeito ao sistema de coordenadas inercial, filtragem de vegetação e por fim triangulação, obtendo assim o modelo do terreno. A Figura 10 ilustra diferentes etapas do processo, incluindo (1) a nuvem de pontos 3D com respeito a  $O_I$ , (2) a classificação dos pontos como vegetação ou terreno, e (3) a superfície gerada pela triangulação.

É possível realizar uma comparação visual do modelo com o terreno real. Na Figura 11 são exibidas imagens obtidas de três trechos do terreno, e podem ser comparadas com as seções (a), (b) e (c) destacadas na Figura 10. O filtro de vegetação permite identificar as regiões do terreno cobertas por grama alta, conforme ilustrado na Figura 12. A mesma figura apresenta declividades e buracos do modelo que correspondem a adversidades reais do terreno.



Figura 11: Imagens do terreno percorrido que podem ser comparadas com o modelo ilustrado nas Figuras 10 e 12.

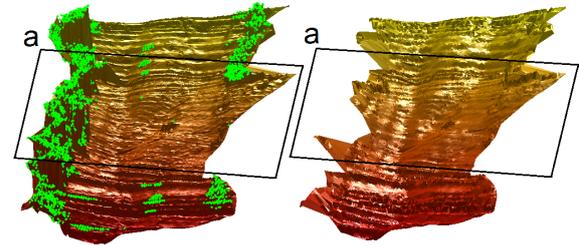


Figura 12: Trecho do terreno modelado destacado na Figura 11(a). À esquerda o modelo com vegetação e à direita o modelo após a filtragem da vegetação.

### 6 Conclusões e Trabalhos Futuros

Considerando a importância da modelagem de terrenos naturais para diversas aplicações, em especial na robótica móvel, este trabalho apresenta uma solução baseada apenas em informações geométricas do ambiente. A metodologia proposta é composta por quatro etapas: remoção de *outliers*, registro com respeito ao sistema de coordenadas inercial, filtragem de vegetação e triangulação de Delaunay.

A modelagem foi realizada *offline* utilizando dados experimentais. O passo seguinte consiste em implementar a metodologia proposta em um robô móvel e realizar experimentos em tempo real. Neste caso, a resolução do modelo deve ser ajustada conforme as limitações de processamento da *hardware* utilizado. A resolução pode ser reduzida através da exclusão de medições do ambiente, o que reduz os requisitos compu-



Figura 9: Imagem do terreno percorrido obtida através da interpolação de imagens obtidas por uma câmera.

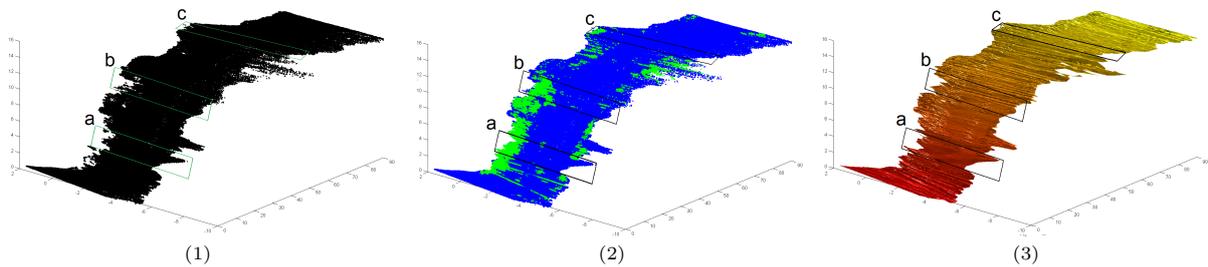


Figura 10: Etapas do processo de modelagem de terreno: (1) Nuvem de pontos 3D; (2) Classificação dos pontos como vegetação ou terreno (verde ou azul respectivamente); (3) Superfície criada pela triangulação dos pontos.

tacionais exigidos, porém aumenta a incerteza do modelo obtido. Outra técnica mais sofisticada de ajuste de resolução consiste em aglutinar triângulos adjacentes semelhantes.

Uma dificuldade associada consiste em estimar os erros do modelo obtido com respeito ao terreno natural percorrido. Futuramente o método será testado em ambientes conhecidos, permitindo a comparação destes. Pensando em sistemas com maior capacidade de processamento, uma linha futura de trabalho consiste em integrar imagens em cores ao sistema para a classificação de vegetação.

#### Agradecimentos

Esse trabalho foi financiado pelo CNPq e FAPERJ. Os autores agradecem a todos os membros do projeto CASC, em especial o Prof. Marcel Bergerman e Prof. Sanjiv Singh pela orientação e suporte prestados.

#### Referências

- Bakambu, J. N., Allard, P. and Dupuis, E. (2006). 3d terrain modeling for rover localization and navigation, *Computer and Robot Vision, 2006. The 3rd Canadian Conf. on*, IEEE, pp. 61–61.
- Barbosa, R. L., SILVA, J., MENEGUETTE, J. and Gallis, R. (2003). Geração de modelo digital de terreno utilizando triangulação de delaunay e thin plate spline, *Anais do III Colóquio Brasileiro de Ciências Geodésicas*.
- Barnhill, R. E. (1977). Representation and approximation of surfaces, *Mathematical Software* **3**: 69–120.
- Batavia, P. H., Roth, S. A. and Singh, S. (2002). Autonomous coverage operations in semi-structured outdoor environments, *Intelligent Robots and Systems, 2002. IEEE/RSJ Int. Conf. on*, Vol. 1, IEEE, pp. 743–749.
- Brotten, G. and Collier, J. (2006). Continuous motion, outdoor, 2 1/2d grid map generation using an inexpensive nodding 2-d laser rangefinder, *Robotics and Automation, 2006. ICRA 2006. Proc. 2006 IEEE Int. Conf. on*, IEEE, pp. 4240–4245.
- De Berg, M., Cheong, O. and Van Kreveld, M. (2008). *Computational geometry: algorithms and applications*, Springer. DOI: [10.1007/978-3-540-77974-2](https://doi.org/10.1007/978-3-540-77974-2)
- Dyn, N., Levin, D. and Rippa, S. (1990). Data dependent triangulations for piecewise linear interpolation, *IMA J. of numerical analysis* **10**(1): 137–154. DOI: [10.1093/imanum/10.1.137](https://doi.org/10.1093/imanum/10.1.137)
- Freitas, G., Hamner, B., Bergerman, M. and Singh, S. (2012). A practical obstacle detection system for autonomous orchard vehicles, *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf. on*, IEEE, pp. 3391–3398.
- Gerbaud, T., Polotski, V. and Cohen, P. (2004). Simultaneous exploration and 3d mapping of unstructured environments, *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, Vol. 6, IEEE, pp. 5333–5337.
- Guibas, L. J., Knuth, D. E. and Sharir, M. (1992). Randomized incremental construction of delaunay and voronoi diagrams, *Algorithmica* **7**(1-6): 381–413.
- Hamner, B., Singh, S., Roth, S. and Takahashi, T. (2008). An efficient system for combined route traversal and collision avoidance, *Autonomous Robots* **24**(4): 365–385. DOI: [10.1007/s10514-007-9082-3](https://doi.org/10.1007/s10514-007-9082-3)
- Hebert, M. and Krotkov, E. (1993). Local perception for mobile robot navigation in natural terrain: Two approaches.
- Hugentobler, M. (2004). *Terrain modelling with triangle based free-form surfaces*, PhD thesis, Citeseer.
- Lee, D.-T. and Schachter, B. J. (1980). Two algorithms for constructing a delaunay triangulation, *Int. J. of Computer & Information Sciences* **9**(3): 219–242. DOI: [10.1007/BF00977785](https://doi.org/10.1007/BF00977785)
- Lingemann, K., Nchter, A., Hertzberg, J. and Surmann, H. (2005). High-speed laser localization for mobile robots, *Robotics and Autonomous Systems* **51**(4): 275–296. DOI: [10.1016/j.robot.2005.02.004](https://doi.org/10.1016/j.robot.2005.02.004)
- M. A. Piteri, M. Meneguete, A. D. S. F. D. O. (2007). Triangulação de delaunay e o princípio de inserção randomizado, *II Simpósio Brasileiro de Geomática* pp. 655–663.
- Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N. and Burgard, W. (2009). A bayesian regression approach to terrain mapping and an application to legged robot locomotion, *J. of Field Robotics* **26**(10): 789–811. DOI: [10.1002/rob.20308](https://doi.org/10.1002/rob.20308)
- Quak, E. and Schumaker, L. L. (1990). Cubic spline fitting using data dependent triangulations, *Computer Aided Geometric Design* **7**(1): 293–301. DOI: [10.1016/0167-8396\(90\)90037-R](https://doi.org/10.1016/0167-8396(90)90037-R)
- Rippa, S. (1990). Minimal roughness property of the delaunay triangulation, *Computer Aided Geometric Design* **7**(6): 489–497. DOI: [10.1016/0167-8396\(90\)90011-F](https://doi.org/10.1016/0167-8396(90)90011-F)
- Sibson, R. (1978). Locally equiangular triangulations, *The Computer J.* **21**(3): 243–245. DOI: [10.1093/comjnl/21.3.243](https://doi.org/10.1093/comjnl/21.3.243)
- Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*, MIT press.
- Sotoodeh, S. (2006). Outlier detection in laser scanner point clouds, *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **XXXVI**(5): 297–302.
- Underwood, J., Hill, A. and Scheduling, S. (2007). Calibration of range sensor pose on mobile platforms, *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ Int. Conf. on*, IEEE, pp. 3866–3871.
- Vandapel, N., Donamukkala, R. and Hebert, M. (2006). Experimental results in using aerial lidar data for mobile robot navigation, *Field and Service Robotics*, Springer, pp. 103–112.
- Wettergreen, D., Moreland, S., Skonieczny, K., Jonak, D., Kohanbash, D. and Teza, J. (2010). Design and field experimentation of a prototype lunar prospector, *The Int. J. of Robotics Res.* **29**(12): 1550–1564. DOI: [10.1177/0278364910370217](https://doi.org/10.1177/0278364910370217)