A Simple Local Path Planning Algorithm for Autonomous Mobile Robots

Fábio Demo da Rosa

Universidade Federal de Santa Maria
Pós-Graduação em Ciência da Computação
Disciplina de Robótica Móvel

faberdemo@gmail.com

23 de novembro de 2023

Visão Geral

Introdução I

- Planejamento de caminho: elemento crucial para robôs móveis;
- Objetivo: determinar rotas para passar por pontos específicos do ambiente;
- As abordagens são de acordo com o ambiente, o tipo de sensor, as capacidades do robô, entre outros;
  - e essas abordagens estão gradualmente buscando um melhor desempenho em termos de tempo, distância, custo e complexidade (BUNIYAMIN et al., 2011).

Abordagens de Planejamento de Caminho I

- Deseja-se encontrar caminhos adequados para um robô com geometria específica;
- Objetiva-se alcançar uma posição e orientação final a partir de uma inicial;
- O problema de navegação do robô móvel pode ser dividido em três subtarefas, de acordo com (BUNIYAMIN et al., 2011):
  - **Mapeamento e Modelagem do Ambiente:** o robô deve ser capaz de construir um mapa do ambiente e modelar o ambiente;
  - **Planejamento de Caminho:** o robô deve ser capaz de planejar um caminho para alcançar o objetivo;
  - **Travessia de Caminho:** o robô deve ser capaz de seguir o caminho planejado e evitar colisões com obstáculos.
- Tipos de ambiente: estático (sem objetos móveis) e dinâmico (com objetos móveis);
- Abordagens: planejamento local e global;

Evolução da Modelagem de Ambiente do Robô I

- **Modelagem de Ambiente:**
  - Década de 1980: Introdução de Cones Generalizados (Generalized Cones) por Rodney Brooks. Funciona em ambientes simples, mas limitado em complexidade.
    - Abordagem de Mapa Rodoviário (Roadmap), incluindo Grafos de Visibilidade (Visibility Graph) e Diagramas de Voronoi. Efetivos em ambientes simples, mas complexos e demorados na criação.
  - Década de 1990: Abordagem de Decomposição Celular (Cell Decomposition) torna-se popular, adequada para ambientes estáticos e dinâmicos, fácil implementação e atualização, mas inicialmente lenta em computadores antigos.
  - Século XXI: Introdução de Quadtree e Framed Quadtree, e Grafos MAKLINK, para aumentar a precisão de caminhos encontrados.

Planejamento de Caminho (Local e Global) I

- **Planejamento de Trajetória Global:**
  - Exige informação prévia do ambiente.
  - Planejamento completo do trajeto antes do movimento do robô.
  - Abordagens incluem Grafos de Visibilidade, Diagramas de Voronoi, Decomposição Celular, e métodos modernos como Algoritmo Genético, Redes Neurais e Otimização de Colônia de Formigas.
- **Planejamento de Trajetória Local:**
  - **Definição:** Essencial para robôs em ambientes dinâmicos, foca na evasão de obstáculos usando sensores.
  - **Funcionamento:** Robôs seguem a rota mais direta, alterando-a ao encontrar obstáculos.
  - **Método do Campo Potencial:** Robôs operam como partículas em campos de potenciais atrativos e repulsivos.
  - **Algoritmo PointBug:** Evita o perímetro dos obstáculos, diferenciando-se dos algoritmos Bug tradicionais.
  - **Implementação do PointBug:** Utiliza sensores de longo alcance e sistemas de navegação para decisões de trajetória.

Algoritmos Bug I

- **Introdução:** Algoritmos Bug são métodos de navegação para robôs móveis em planejamento de trajetória local, com sensores mínimos e algoritmos simples.
- **Variações Comuns:** Incluem Bug1, Bug2, DistBug, VisBug e TangentBug. Outras variações são Alg1, Alg2, Class1, Rev/Rev2, OneBug e LeaveBug.
- **Evolução e Melhorias:** Focados em caminhos mais curtos, menor tempo, algoritmos mais simples e melhor desempenho.
- **Bug1 e Bug2:** Bug1 é cauteloso e cobre mais do que o perímetro total do obstáculo, enquanto Bug2 é menos eficiente em alguns casos, mas tem cobertura menor.
- **Desenvolvimentos Posteriores:** VisBug, Alg1 e Alg2 melhoram a eficiência, com Rev1 e Rev2 resolvendo problemas de procedimentos reversos. DistBug e TangentBug incorporam sensores de alcance.
- **Aplicação em Planejamento de Trajetória Local:** Utilizados para navegação em ambientes desconhecidos ou dinâmicos, adaptando-se a mudanças e obstáculos.

Algoritmos PointBug I

- **Contexto:** O PointBug é um algoritmo de navegação recentemente desenvolvido para robôs em ambientes planares desconhecidos com obstáculos estacionários.
- **Detecção de Pontos:** Utiliza um sensor de alcance para detectar mudanças súbitas na distância até o obstáculo mais próximo. Mudanças súbitas são identificadas quando a distância varia significativamente em um curto intervalo de tempo.
- **Determinação do Próximo Ponto:** O próximo ponto de movimento é determinado pela saída do sensor de alcance, baseando-se na variação (Δd) da distância detectada.
- **Funcionamento do Algoritmo:** O robô inicialmente se direciona ao ponto-alvo, rotacionando para localizar um ponto súbito e mover-se em sua direção. Esse processo se repete até o robô atingir o ponto-alvo.
- **Resolução de Problemas de Mínimos Locais:** O algoritmo é eficaz para resolver o problema de mínimos locais em ambientes desconhecidos, identificando pontos súbitos de maneira confiável.

Algoritmos PointBug II

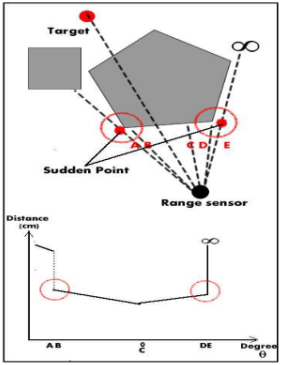
- **Exemplos e Demonstrações:** O artigo (BUNIYAMIN et al., 2011) analisa obstáculos de diferentes formas e como o algoritmo identifica os pontos súbitos.

```
While Not Target
  If robot rotation <= 360
    Robot rotates right of left according to position of dmin
  If sudden point
      If 180 degree rotation
        Ignore reading /* to avoid robot return to previous point */
      Else
        Get distance from current sudden point to next sudden point
        Get angle of robot rotation
        Move to new point according to distance and rotation angle
        Record New dmin value
        Reset rotation
      End if
    End if
  Else
```

Algoritmos PointBug III

```
16      Robot Stop /* No sudden point and exit loop */
17    End if
18  While end
19  Robot Stop /* Robot successfully reaches target */
```

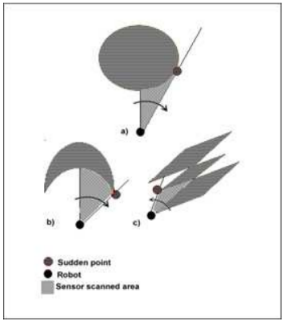
Algoritmos PointBug IV



Fonte: (BUNIYAMIN et al., 2011)

Figura 1: O sensor de alcance está detectando um obstáculo da esquerda para a direita e da direita para a esquerda.

Algoritmos PointBug V



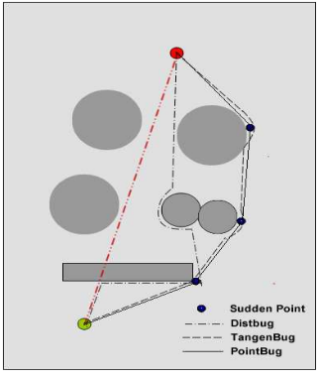
Fonte: (BUNIYAMIN et al., 2011)

Figura 2: Trajetória gerada pelo PointBug para resolver o problema de mínimos locais.

Simulação I

- A simulação do algoritmo de navegação ponto a ponto (PointBug) foi realizada utilizando ActionScript 2.0 no Adobe Flash CS3.
- Foram simulados três tipos de ambientes: um ambiente livre, um ambiente baseado em labirinto e um ambiente semelhante a um escritório.

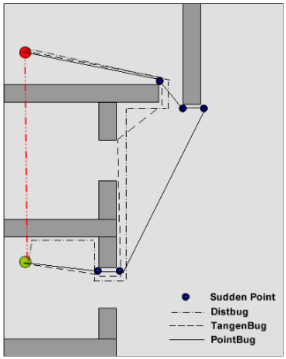
Simulação II



Fonte: (BUNIYAMIN et al., 2011)

Figura 3: Trajetória gerada usando os algoritmos Distbug, TangenBug e PointBug em ambiente livre.

Simulação III



Fonte: (BUNIYAMIN et al., 2011)

Figura 4: Trajetória gerada usando o algoritmo Distbug, TangenBug e PointBug em um ambiente simples como o do Office.

Conclusões I

- **Simulação do Algoritmo PointBug:** A simulação do algoritmo de navegação PointBug foi realizada utilizando ActionScript 2.0 no Adobe Flash CS3.
- **Ambientes de Teste:** Foram simulados três tipos de ambientes: um ambiente livre, um ambiente baseado em labirinto e um ambiente semelhante a um escritório.

BUNIYAMIN, Norlida et al. A simple local path planning algorithm for autonomous mobile robots. **International journal of systems applications, Engineering & development**, v. 5, n. 2, p. 151–159, 2011.

A Simple Local Path Planning Algorithm for Autonomous Mobile Robots

Fábio Demo da Rosa

Universidade Federal de Santa Maria
Pós-Graduação em Ciência da Computação
Disciplina de Robótica Móvel

faberdemo@gmail.com

23 de novembro de 2023