

## Algoritmos de planejamento de caminhos 3D\*

Cleiton Almeida dos Santos<sup>1</sup>, André Luiz Pires Guedes<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)  
Centro politécnico – Curitiba – PR – Brazil

{casantos, andre}@inf.ufpr.br

**Abstract.** *This extended abstract presents a research in progress about 3D path planning algorithms. The computational problem of interest is to find a path of optimal or near-optimal length from an origin point  $s$  to a destination point  $d$  in a 3D environment with obstacles. Is part of this research to identify a computational representation, a data structure, that better describe the environment, specially space partition in voxels and visibility graphs. A comparison of this two and a proposed mixed representation is to be done. Other computational geometry methods, as Voronoi diagrams, octrees, space subdivision data structures, etc., are used in the project. The obtained algorithms may be included in autonomous vehicle software, digital games, crowd simulations, and robotics.*

### 1. Introdução

Existe uma recente demanda por aplicações computacionais relacionadas a ambientes tridimensionais, sejam elas reais ou virtuais. Tecnologias como veículos autônomos, drones, jogos digitais, simulação de multidões, robótica, impressoras 3D, entre outros, necessitam de algoritmos eficientes. A pesquisa em andamento, em planejamento de caminhos, busca criar e refinar algoritmos para suporte à movimentação de agentes em um ambiente 3D. Entenda-se por agente um personagem, uma ferramenta, um veículo ou algum objeto, que precisa se movimentar com segurança no ambiente escolhido.

Seja um ambiente  $W$  composto por regiões ocupadas por obstáculos  $W_o$  e regiões livres de obstáculos  $W_f$  onde  $W = W_o \cup W_f$  e  $W_o \cap W_f = \emptyset$ . Escolhidos um ponto de origem  $s$  e um ponto de destino  $d$  na região livre de obstáculos  $s, d \in W_f$ . O problema de planejamento de caminhos consiste em encontrar uma curva que começa em  $s$  e termina em  $d$  e está totalmente contida em  $W_f$ . O planejamento de caminhos no  $\mathbb{R}^3$  é bem mais difícil de ser implementado que o planejamento de caminhos no plano ( $\mathbb{R}^2$ ) [Berg et al. 2008]. Por exemplo, para desviar de um obstáculo no  $\mathbb{R}^2$  é feita a escolha entre 2 alternativas, mas no  $\mathbb{R}^3$  a escolha passa a ser entre infinitas alternativas.

Segundo [Yang et al. 2016] os algoritmos de planejamento de caminhos em ambiente 3D podem ser classificados em: baseados em amostragem, baseados em modelos matemáticos, bioinspirados, baseados em nodos e multi-fusão. Os algoritmos baseados em amostragem falham ou ficam lentos para gerar caminhos sinuosos. Os algoritmos baseados em modelos matemáticos apesar dos ótimos resultados são lentos e especializados. Os bioinspirados não possuem garantias de qualidade dos resultados nem do tempo de execução. Os algoritmos baseados em nodos permitem ajustar entre qualidade do caminho e tempo de execução. A presente pesquisa se concentra nos algoritmos baseados em amostragem, baseados em nodos e na fusão destes. Na seção 2, será detalhada a solução proposta, e na seção 3 as considerações finais.

---

\*Com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

## 2. Solução proposta

Dentre algoritmos de planejamento de caminhos foram escolhidos inicialmente para implementação: grafo de visibilidade e grade 3D. Esses algoritmos possuem complexidade de tempo diferentes e são passíveis de ajustes em diversos aspectos para melhorar desempenho. Cada obstáculo do ambiente é um volume convexo. Um obstáculo não convexo é previamente decomposto em um conjunto de volumes convexos. Cada volume é organizado em *doubly connected edge list* (DCEL) [Berg et al. 2008]. A partir dos obstáculos e de um valor de folga é obtido um *boundingbox* alinhado com os eixos cartesianos, que contém no seu interior todos os obstáculos e espaço livre envolvendo os obstáculos com espessura mínima do valor de folga informado.

O grafo de visibilidade  $G$  é gerado criando, para cada vértice dos obstáculos,  $P$ , um vértice em  $G$ ,  $v_P$ . Um par de vértices de  $G$ ,  $v_P$  e  $v_Q$  é adjacente se e somente se  $P$  e  $Q$  são visíveis entre si, ou seja, o segmento de reta  $\overline{PQ}$  não atravessa algum obstáculo [Berg et al. 2008]).

A grade 3D é gerada pela subdivisão de todo o ambiente em pequenos cubos no tamanho definido. Apenas os cubos que estão inteiramente no espaço livre são representados como vértices em um grafo. São geradas arestas em  $G$  entre os vértices geometricamente vizinhos [Yang et al. 2016]. Cada vértice do grafo possui grau máximo 26.

No grafo gerado por um dos métodos anteriores, será aplicado um dos algoritmos de busca de caminho mínimo. O planejamento de caminhos proposto consistirá em incluir 2 novos vértices ao grafo  $v_s$  e  $v_d$ , respectivamente origem e destino, conectá-los ao grafo e encontrar o caminho mínimo entre  $v_s$  e  $v_d$ .

## 3. Considerações finais

Uma vez implementados os algoritmos citados, serão realizados testes e análises de desempenho em variados cenários, por exemplo, urbano, floresta, rural, subaquático, edificações, cavernas, etc. A partir dos resultados obtidos novas hipóteses serão levantadas. Algumas já até podem ser vislumbradas:

- Aplicar o algoritmo de visibilidade na grade gerada, criando saltos no grafo.
- Acelerar a geração do grafo de visibilidade pela avaliação dos vetores normais das faces dos obstáculos criando uma lista reduzida de candidatos a visibilidade.
- Redução do grafo gerado pela grade através de aglutinação de cubos vizinhos.
- No grafo de visibilidade, incluir vértices sobre arestas e/ou faces dos obstáculos para obter caminhos menores e/ou mais suaves.

No avançar dessa pesquisa quando resultados forem obtidos publicações serão feitas. Haverá também disponibilização dos códigos fonte em um repositório *git*. O desafio é como manter o grafo pequeno preservando os vértices e arestas mais significativos para obter caminhos mínimos próximos de ótimo.

## Referências

- Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition.
- Yang, L., Qi, J., Song, D., Xiao, J., Han, J., and Xia, Y. (2016). Survey of robot 3d path planning algorithms. *Journal of Control Science and Engineering*, 2016:5.