

Planejamento de caminhos 3D: Comparação dos algoritmos por grade 3D e grafo de visibilidade *

Cleiton A. Santos ¹, André L. P. Guedes ¹

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

{casantos, andre}@inf.ufpr.br

Resumo. Este artigo descreve o andamento de uma pesquisa focada em planejamento de caminhos 3D. O objetivo da pesquisa é encontrar algoritmos de baixa complexidade computacional, capazes de gerar um caminho próximo do ótimo em um ambiente controlado. Os algoritmos em estudo possuem como entrada a descrição do ambiente, e como saída o melhor caminho entre os pontos de origem e de destino. As abordagens por visibilidade e por grade são comparadas em ambientes simples. Também são propostas melhorias.

Abstract. This paper describes the progress of a research focused on 3D path planning. The goal of the research is to find algorithms of low computational complexity capable of generating a near-optimal path in a controlled environment. The algorithms under study have as input the description of the environment, and as output, the best path between the source point and the target point. We compare the approaches by visibility and by grid in simple environments. We also propose improvements.

Palavras-chave: Planejamento de caminhos 3D; Grafo de visibilidade 3D; Grade 3D.

1. Introdução

Para atender o recente aumento da demanda por aplicações computacionais relacionadas a ambientes tridimensionais, tais como: veículos autônomos, drones, jogos digitais, simulação de multidões, robótica, impressoras 3D, entre outros são necessários 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.

Em um ambiente W com obstáculos convexos W_o e espaço livre $W_f = W \setminus W_o$, dados os pontos origem s e destino t , contidos no espaço livre ($s, t \in W_f$), uma curva \tilde{st} , também contida no espaço livre ($\tilde{st} \in W_f$) é chamada de *caminho*. Em W podem existir vários caminhos de s a t . O melhor caminho, aquele de menor custo, é chamado de *caminho ótimo* e denotado por \tilde{st}^* . *Planejamento de caminhos* é o processo para encontrar os caminhos e identificar o caminho ótimo. Em um ambiente 3D o conjunto de caminhos

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

é infinito e não computável. Achar o caminho ótimo é provado por [Canny 1988] ser NP-Difícil. Portanto, é adotada uma variação do planejamento de caminhos que encontra alguns caminhos e identifica o *melhor caminho discretizado* que é o mais próximo do caminho ótimo que a representação do ambiente permite alcançar em tempo polinomial.

A pesquisa em andamento tem por objetivo encontrar algoritmos para planejamento de caminhos em um ambiente 3D controlado, que gerem caminhos próximos do ótimo e com baixa complexidade computacional. Em [Yang et al. 2016] são apresentadas abordagens de algoritmos para planejamento de caminho. Os algoritmos de interesse precisam atender aos requisitos de obter os melhores resultados em tempo de execução previsível e o algoritmo precisa ser aplicável a diferentes cenários. Por não atender estes requisitos, as abordagens bioinspiradas (ACO, GA, PSO, etc.), de exploração aleatória (PRM, RRT, etc.), baseadas em modelos matemáticos (MILP, BIP) e outras como campos potenciais artificiais (APF) foram descartadas.

A fase inicial da pesquisa descrita em [Santos e Guedes 2019] consistiu em identificar as abordagens que poderiam ser empregadas, escolher algumas consideradas promissoras, implementar e avaliar os resultados. O critério de comparação é a qualidade dos caminhos gerados, ou seja, caminhos de menor custo. Foi objetivo na implementação inicial avaliar a qualidade das saídas dos algoritmos. Embora o potencial de reduzir a complexidade computacional dos métodos tenha sido relevante na escolha. As abordagens trabalhadas inicialmente foram algoritmos baseados em grade e visibilidade. Para eficiente busca de caminho mínimo em grafos, tanto densos como esparsos, foram implementados os métodos *Dijkstra* [Dijkstra, 1959] e *Floyd-Warshall* [Ingerman, 1962].

O presente artigo tem por objetivo apresentar as análises e resultados preliminares, bem como, descrever o que será objeto de estudo a seguir. Na Seção 2 são detalhados os algoritmos implementados. A Seção 3 é apresenta uma comparação dos resultados gerados pelos algoritmos. Na Seção 4 são descritas possíveis melhorias nos algoritmos. Finalmente na Seção 5 estão as considerações finais.

2. Os algoritmos implementados

Os algoritmos de planejamento de caminhos baseados em grade e visibilidade, são conhecidos e aplicados em ambientes 2D [Andayesh e Sadeghpour 2014, Berg et al. 2008]. Na presente pesquisa, é considerada a expansão para ambientes 3D. Os algoritmos implementados são detalhados a seguir.

2.1. Algoritmos baseados em grade 3D

A ideia base dos algoritmos por grade 3D é particionar o ambiente em uma grade uniforme de pequenos volumes cúbicos, também chamados de *voxels*. A primeira etapa do algoritmo é marcar cada elemento da grade como livre ou ocupado. São considerados *voxels* ocupados aqueles que pelo menos um obstáculo ocupa pelo menos uma parte do seu volume. Apenas os cubos livres interessam. A partir dos livres é criado um grafo onde cada *voxel* é representado por um vértice no grafo. As arestas do grafo são baseadas na vizinhança de cada cubo. Os critérios de vizinhança podem ser faces, arestas ou vértices em comum entre dois *voxels*. Ao grafo gerado, a partir da grade, são adicionados

dois vértices e duas arestas. Os vértices adicionados representam os pontos de origem s e destino t , e as arestas adicionadas representam a ligação dos novos vértices com os representantes dos *voxels* que contém s e t . Com o grafo gerado, são utilizados algoritmos de busca em grafos para encontrar o melhor caminho. Nas Figuras 1 e 2 podem ser visualizadas as etapas do algoritmo baseado em grade para obstáculos com formatos de cubo e esfera¹ respectivamente, na Figura 1 o espaço foi subdividido em 27 *voxels* e na Figura 2 o espaço foi subdividido em 729 *voxels*. É intencional o cubo possa ser inscrito na esfera, estejam na mesma posição e ocupem os mesmos *voxels*. Se trocar o cubo pela esfera na Figura 1 ou a esfera pelo cubo na Figura 2 os caminhos gerados respectivamente pelo algoritmo de grade serão o mesmo.

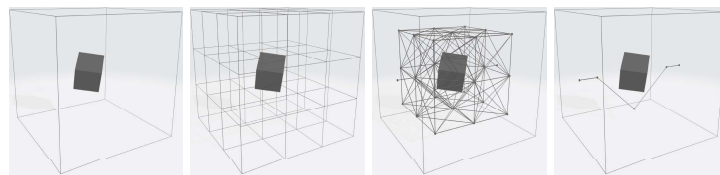


Figura 1. Grade 3D - Cubo - 27 *voxels*

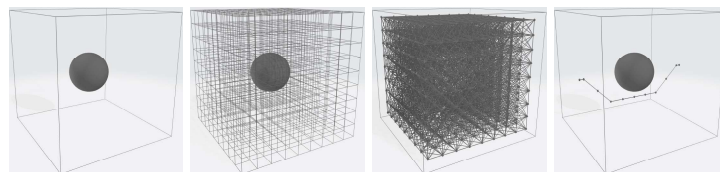


Figura 2. Grade 3d - Esfera - 729 *voxels*

2.2. Algoritmos baseados em visibilidade

A abordagem dos algoritmos por grafo de visibilidade consiste em construir um grafo a partir dos vértices dos obstáculos, sendo cada um representado por um vértice no grafo. São acrescentados ao grafo, também, 2 vértices para representar os pontos de origem s e destino t . Se dois vértices do grafo são visíveis entre si é criada uma aresta entre eles no grafo. Ser visível significa que um segmento de reta entre o par de vértices não atravessa obstáculos. Uma vez gerado o grafo, são aplicados algoritmos de busca de caminhos em grafos para encontrar o melhor caminho. Nas Figuras 3 e 4 podem ser visualizadas as etapas do algoritmo baseado em visibilidade para cubo e esfera respectivamente.

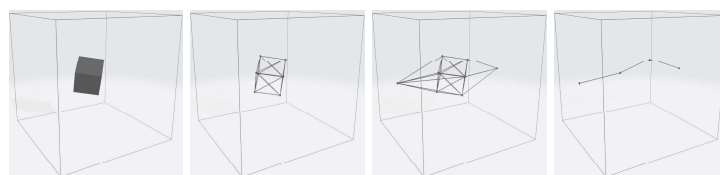


Figura 3. Grafo de visibilidade - Cubo

¹O termo *esfera* se refere a um poliedro regular com muitas faces e aparência de uma esfera.

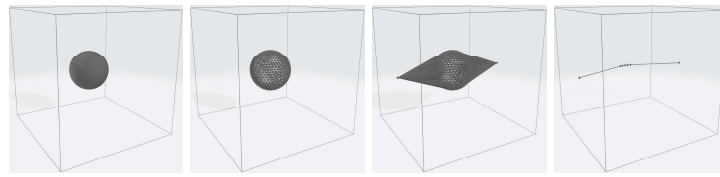


Figura 4. Grafo de visibilidade - Esfera

3. Comparação dos métodos

A abordagem por grade 3D, sofre de problema semelhante a “maldição de dimensionalidade”. Quanto menores os *voxels* melhor será caminho obtido. Mas reduzir os *voxels* provoca aumento no volume de dados. O caminho gerado por grade possui muitas arestas e aparência serrilhada. Por outro lado, se um obstáculo for substituído por outro mas ambos ocupam os mesmos *voxels*, não importando a complexidade deles, o caminhos gerado, pelo algoritmo por grade 3D, será o mesmo. Já a abordagem por grafo de visibilidade se baseia nos vértices dos obstáculos e sofre degradação de desempenho quando os obstáculos são poliedros complexos. O caminho gerado por visibilidade costuma ser menor em custo, com arestas mais longas e em menor quantidade se comparado com o caminho gerado a partir da Grade 3D.

Tabela 1. Comparação entre as abordagens. Grade 3 x 3 x 3 (27 *voxels*, com 26 livres) e 9 x 9 x 9 (729 *voxels*, com 702 livres); Visibilidade Cubo (8 vértices, 18 arestas, 12 faces) e Esfera (642 vértices, 1.920 arestas, 1.280 faces)

		Grade		Visibilidade Cubo		Visibilidade poliedro	
		3x3x3	9x9x9	Vértice	Aresta	Vértice	Aresta
Grafo	Vértices	28	704	10	20	644	1.922
	Arestas	268	13.812	68	148	4.768	10.264
Caminho	Vértices	5	11	4	3	7	12
	Arestas	8	20	6	4	12	22
	Tamanho	1.108,5	1.025,7	906,3	895,4	912,8	912,6

A Tabela 1 compara os resultados obtidos. Os obstáculos são um cubo e uma esfera. As colunas a esquerda representam experimentos com grade, e a direita por visibilidade. Por grade o ambiente de dimensão 900 x 900 x 900, foi dividido nos experimentos em 27 *voxels* de 300 x 300 x 300, e em 729 *voxels* de 100 x 100 x 100. Os obstáculos estão posicionados no centro do ambiente e o diâmetro de ambos é 300, ocupando um *voxel* e 27 *voxels* respectivamente. Por visibilidade as colunas da tabela, foram subdivididas em visibilidade por vértices (ver. Subseção 2.2) e por arestas (ver Subseção 4.2). Na tabela as linhas superiores apresentam as dimensões do grafo que representa o ambiente, e as linhas inferiores apresentam o melhor caminho obtido entre *s* e *t*. Por grade, os vértices do grafo representam os *voxels* livres e as arestas do grafo representam a vizinhança dos *voxels*. Por visibilidade, os vértices do grafo representam pontos do ambiente, que são, os vértices ou as arestas dos obstáculos e as arestas do grafo indicam a visibilidade entre os pares de vértices. O grafo é orientado, portanto, existem duas arestas para cada par de vértices conectado. Ao grafo são acrescentados dois vértices extra para representar *s* e *t*.

Por grade são acrescentadas quatro arestas extra para a ligação de s e t com os *voxels* que os contém. Por visibilidade são adicionadas as arestas extra para a ligação dos vértices extra aos demais vértices visíveis. Na grade a relação de vizinhança adotada é 26 (faces, arestas e vértices) e o grau máximo é 26. O grau máximo por visibilidade é, no pior caso, o número de vértices menos um. O melhor caminho é indicado pelos os vértices, arestas e o custo entre s e t .

4. Melhorias nos algoritmos

Em ambas as abordagens, existem formas de melhorar tanto os resultados como o desempenho dos algoritmos. Nas subseções a seguir são abordadas algumas alternativas.

4.1. Otimização de algoritmo baseado em grade 3D

Para otimizar a abordagem por grade é necessário buscar formas de reduzir o número de vértices do grafo mantendo os mais significativos. Uma maneira de obter este efeito seria utilizar *voxels* de tamanho variável. Um critério de redução seria adotar perto de obstáculos *voxels* pequenos e longe de obstáculos *voxels* grandes. Para se obter esta variação de tamanho é possível utilizar uma *octree*, que é uma estrutura de dados em forma de árvore, em que cada nó-pai possui 8 nós-filhos, que representam a subdivisão de um volume maior em 8 volumes menores [Meagher 1982]. Para reduzir o efeito serrilhado, seria necessário fazer um pós-processamento para suavizar e simplificar o caminho, eliminando vértices e unindo arestas adjacentes. Para 3 vértices consecutivos no caminho gerado, verificar se é possível ligar diretamente, por uma aresta, o primeiro vértice ao último vértice, eliminando o vértice do meio.

4.2. Otimização de algoritmo baseado em visibilidade

Para otimizar os grafos de visibilidade existe um número maior de hipóteses. É importante observar que em ambientes 2D o melhor caminho passa pelos vértices dos obstáculos porém em ambientes 3D o melhor caminho passa por algum ponto desconhecido das arestas dos obstáculos [Berg et al. 2008]. Construir o grafo a partir do ponto médio das arestas deve melhorar o caminho. Porém para obter melhora significativa seria necessário adicionar um algoritmo de pós-processamento iterativo no caminho gerado fazendo uma espécie de busca binária sobre as arestas dos obstáculos até que o ganho obtido comparado com a iteração anterior seja pequeno. Essa hipótese otimiza uma solução inicial, encontrando o melhor ponto de passagem nas arestas de obstáculos previamente escolhidas. Pois o caminho ótimo pode passar por arestas de obstáculos não escolhidas inicialmente. Como este pós-processamento é sobre o caminho gerado e não sobre o grafo inteiro o custo computacional não é tão elevado.

Outra otimização a ser considerada é que um par de vértices, quanto a visibilidade, pode ser classificado em 4 grupos:

- **Ocluso** - Há um obstáculo entre os vértices do par bloqueando a visibilidade.
- **Invisível** - Pelo menos um dos vértices do par é adjacente a somente faces ocultas.
- **Central** - Pelo menos um dos vértices do par é adjacente a somente faces visíveis.
- **Tangente** ambos os vértices estão na fronteira entre as faces visíveis e ocultas.

Somente os pares **tangentes** são candidatos a participar do melhor caminho. Os pares **centrais** geram caminhos mais longos e incluí-los infla o grafo de arestas irrelevantes tornando a busca de caminhos mais lenta. Os pares **invisíveis** sobrecarregam o teste de visibilidade que é o trecho de maior complexidade computacional do algoritmo. A classificação dos pares como **invisível**, **central** ou **tangente** pode ser feita pelas normais das faces adjacentes. É necessário verificar a visibilidade para descartar os pares **occlusos**. Se a verificação de visibilidade for ordenada em uma direção do espaço é possível detectar a oclusão com menor custo computacional. Os vértices tangentes de um obstáculo próximo formam um fecho convexo e todos os pares de vértices que intersectam este fecho convexo são pares oclusos.

5. Considerações finais

Pelos aspectos citados, avaliando os resultados apresentados na Tabela 1 e pelo potencial de otimização, a abordagem por grafo de visibilidade tem demonstrado ser mais promissora que a abordagem por grade. Porém, como já comentado, em cenários complexos a abordagem por grade leva vantagem. Combinar as abordagens permitiria aproveitar o melhor de cada uma. Existe uma ordem topológica no ambiente que ainda precisa ser melhor compreendida e aproveitada. Encontrar uma estrutura de dados adequada é uma meta e existem outras abordagens, citadas por [Yang et al. 2016] como alternativas, que precisam ser verificadas, por exemplo: particionamento binário do espaço (BSP), tetraedralização de Delaunay e/ou diagrama de Voronoi e fecho convexo. Outra perspectiva que existe é que sendo encontrada uma solução satisfatória na terceira dimensão, avançar para dimensões mais altas será menos custoso. Os problemas do mundo real, que são os mais interessantes, necessitam de muitos graus de liberdade, recaindo nas dimensões mais elevadas.

Referências

- Andayesh, M. e Sadeghpour, F. (2014). A comparative study of different approaches for finding the shortest path on construction sites. *Procedia Engineering*, 85:33–41.
- Berg, M. d., Cheong, O., Kreveld, M. v., e Overmars, M. (2008). *Computational geometry : algorithms and applications*. Springer, Germany, 3rd ed edition.
- Canny, J. (1988). *The complexity of robot motion planning*. MIT press.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Ingerman, P. Z. (1962). Algorithm 141: path matrix. *Communications of the ACM*, 5(11):556.
- Meagher, D. (1982). Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147.
- Santos, C. A. e Guedes, A. L. P. (2019). Algoritmos de planejamento de caminhos 3D. *Anais do Computer on the Beach*, pages 774–775.
- Yang, L., Qi, J., Song, D., Xiao, J., Han, J., e Xia, Y. (2016). Survey of robot 3D path planning algorithms. *Journal of Control Science and Engineering*, 2016:5.