

PLANEJAMENTO AUTOMÁTICO DE ROTAS DE NAVEGAÇÃO PARA VANTS ATRAVÉS DE ÁRVORES ALEATÓRIAS DE RÁPIDA EXPLORAÇÃO E MODELOS DIGITAIS DE ELEVAÇÃO

Lucas Masiero Silva

Universidade Federal de São Paulo
Avenida Cesare Monsueto Giulio Lattes, 1211 - Jardim Santa Inês I, São José dos Campos - SP,
12231-280
lucasmasier@gmail.com

Monica Maria De Marchi

Instituto de Estudos Avancados
Trevo Coronel Aviador José Alberto Albano do Amarante - 01 - Putim, São José dos Campos -
SP, 12228-001
monica@ieav.cta.br

Felipe Leonardo Lobo Medeiros

Instituto de Estudos Avancados
Trevo Coronel Aviador José Alberto Albano do Amarante - 01 - Putim, São José dos Campos -
SP, 12228-001
felipe@ieav.cta.br

RESUMO

Este trabalho aborda o planejamento automático de rotas de navegação para Veículos Aéreos Não Tripulados (VANTS), com um ambiente de navegação definido por um modelo digital de elevação. Uma rota de navegação é a estrutura através da qual uma trajetória segura e dinamicamente viável é criada por meio das restrições cinemáticas e dinâmicas do VANT. Rotas de navegação podem ser planejadas através da aplicação de métodos de busca em grafos. O objetivo deste trabalho é a aplicação de árvores aleatórias de rápida exploração, tradução de *Rapidly-exploring Random Trees* (RRTs), no problema de planejamento de rotas mencionado. Rotas planejadas através de RRTs são apresentadas neste trabalho. Alguns destes resultados são comparados com uma rota planejada com a aplicação do algoritmo Dijkstra, considerando amostragens do mesmo ambiente de navegação representadas por um grafo de visibilidade.

PALAVRAS CHAVE. Árvores aleatórias de rápida exploração, Planejamento automático de rotas, Modelos digitais de elevação.

ÁREA PRINCIPAL: MH – Metaheurísticas

ABSTRACT

This work approaches the automatic planning of paths for Unmanned Aerial Vehicles (UAVs), with a navigation environment defined by a digital elevation model. A navigation path is the structure through which a safe and dynamically feasible trajectory is planned by the kinematic and dynamic constraints of the UAV. Navigation paths can be planned by applying search methods in graphs. The objective of this work is the application of Rapidly-exploring Random Trees (RRTs) in the mentioned path planning problem. Paths planned through RRTs are presented in this work. Some of these results are compared with a path planned by the application of the Dijkstra's algorithm, considering samples of the same navigation environment represented by a visibility graph.

KEYWORDS. Rapidly-exploring random trees, Automatic path planning, Digital elevation models.

MAIN AREA: MH – Metaheuristics

1. Introdução

Um dos principais focos das pesquisas relacionadas a Veículos Aéreos Não Tripulados (VANTs) é o aumento de autonomia destes veículos, que consiste na diminuição do grau de dependência do VANT de operadores, transferindo parte do processo de tomada de decisão do operador para o próprio veículo. O cálculo ou planejamento automático de rotas de navegação é essencial para a implementação da maioria das novas capacidades pretendidas com o aumento da autonomia de VANTs.

Uma rota de navegação para um VANT é a estrutura através da qual uma trajetória segura e dinamicamente viável é criada por meio das características cinemáticas e dinâmicas do veículo. Uma rota é transformada em uma trajetória de navegação através da aplicação de métodos de suavização como, por exemplo, as curvas de Dubins [2] e as curvas de Bézier [7].

Rotas de navegação podem ser planejadas através da aplicação de métodos de busca em grafos como, por exemplo, o algoritmo Dijkstra [1] e o algoritmo A* [3]. Estes grafos são constituídos por nós que são posições de navegação obtidas por amostragens das regiões navegáveis de um ambiente de navegação. Grafos de visibilidade, roadmaps e diagramas de Voronoi generalizados são exemplos destes grafos [6]. Um exemplo de uma rota e um exemplo de uma trajetória planejada através desta rota são apresentados na Figura 1. Nesta figura, os polígonos pretos representam obstáculos à navegação do veículo. A rota de navegação é apresentada na cor azul e a trajetória na cor vermelha.

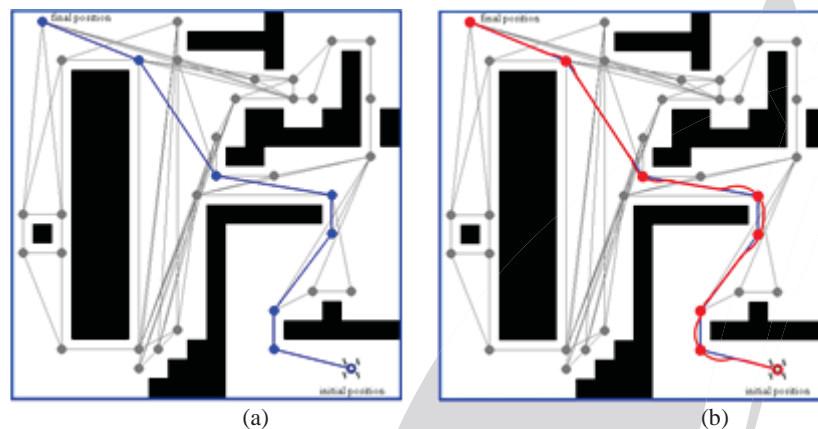


Figura 1. Exemplos de uma (a) rota de navegação e de uma (b) trajetória segura e dinamicamente viável, planejadas através de um grafo de visibilidade reduzido.

Árvores aleatórias de rápida exploração [4][5], tradução de Rapidly-exploring Random Trees (RRTs), são métodos que também podem ser usados no planejamento automático de rotas.

Este trabalho tem como objetivo a aplicação de RRTs ao planejamento de trajetórias para VANTs, considerando o ambiente de navegação definidos por modelos digitais de elevação.

A Seção 2 deste relatório descreve o método RRT. Na Seção 3 são apresentados alguns resultados obtidos com a aplicação do método RRT. Na Seção 4 são apresentadas algumas observações sobre o projeto em desenvolvimento.

2. Método RRT

Uma RRT é uma combinação de um grafo na forma de uma árvore, com um método simplificado de busca. O nó raiz da árvore é a posição inicial da rota a ser planejada. O funcionamento de uma RRT clássica ou padrão consiste em expandir a árvore de modo aleatório a partir de seu nó raiz até que um determinado número de iterações seja alcançado. A expansão da árvore corresponde a uma amostragem das regiões navegáveis do ambiente de navegação. Um exemplo de RRT é apresentado na Figura 2.

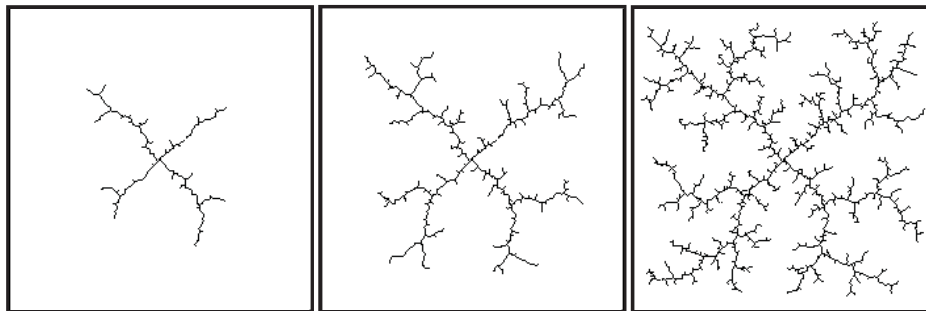


Figura 2. Exemplos de árvores geradas usando o método RRT, variando o número de interações (K).
 Fonte: LaValle [5]

Algumas características promissoras das RRTs são [5]: a expansão de uma RRT é fortemente direcionada para espaços menos explorados do ambiente de navegação; pode ser usada em ambientes de navegação com quaisquer dimensões; pode ser utilizada em ambientes com obstáculos não convexos; uma RRT é probabilisticamente completa; uma RRT pode ser adaptada como um método de planejamento de rotas.

O algoritmo de construção de uma RRT é apresentado na Tabela 1. Neste trabalho, este algoritmo foi implementado na linguagem de programação C. Os gráficos foram implementados através da biblioteca gráfica e computacional denominada OpenGL.

Tabela 1. Algoritmo para a construção de uma RRT

Etapas	Descrição
1	inserir a raiz q_{init} na árvore G
2	para $k=1$ até K faça
3	$q_{rand} \leftarrow \text{RAND_CONFIG}(C)$
4	$q_{near} \leftarrow \text{NEAREST_VERTEX}(q_{rand}, G)$
5	$q_{new} \leftarrow \text{NEW_CONF}(q_{near}, \Delta q)$
6	inserir o nó q_{new} na árvore G
7	inserir a aresta que une q_{near} a q_{new} na árvore G
8	retornar a árvore G

No algoritmo de construção de uma RRT: G é o grafo que representa a árvore RRT; K é o número máximo de iterações do algoritmo; q_{init} é uma posição que corresponde à raiz da árvore; q_{rand} é uma posição gerada aleatoriamente no espaço C ; q_{near} é o nó da árvore mais próximo da posição q_{rand} ; q_{new} é um novo nó da árvore gerado no segmento de reta que une q_{near} a q_{rand} , e cuja distância em relação a q_{near} é igual a Δq ; RAND_CONF é uma função que gera aleatoriamente uma posição contida no espaço C , tal que o segmento de reta q_{near}, q_{new} não deve interceptar qualquer obstáculo contido em C ; NEAREST_VERTEX é uma função que retorna o nó q_{near} mais próximo de q_{rand} ; e NEW_CONFIG é uma função que gera q_{new} seguindo as restrições descritas anteriormente.

Neste trabalho, cada espaço C é a representação computacional bidimensional de um ambiente de navegação, que é delimitado por duas posições geográficas: a primeira posição inferior à esquerda (posição delimitadora inicial); e a última posição superior à direita (posição delimitadora final). Foi utilizado um ambiente definido pelo modelo digital de elevação apresentado na Figura 3, que usa o datum WGS84 como modelo matemático teórico da representação da superfície da Terra. Todos os ambientes de navegação são matrizes binárias, em

que as células com valor 1 indicam obstáculos à navegação do veículo.

No ambiente de navegação da Figura 4b, os obstáculos são definidos considerando uma altitude de corte de 1200 metros (m). Isto significa que toda célula do modelo digital de elevação com valor superior ou igual a 1200 m corresponde a uma célula obstáculo do ambiente de navegação. Deste modo, na utilização deste ambiente para o planejamento de rotas para um VANT, a altitude de navegação do VANT pode ser especificada como sendo igual a soma da altitude de corte com uma altura de segurança. Este ambiente de navegação é uma matriz quadrada de ordem 1201. Cada célula corresponde a uma região real com 90 m de largura por 90 m de comprimento. O ambiente de navegação é delimitado pelas posições geográficas $(lat_{inf}, long_{inf}) = (-22.994583, -45.99875)$ e $(lat_{sup}, long_{sup}) = (-22.000417, -45.002917)$.

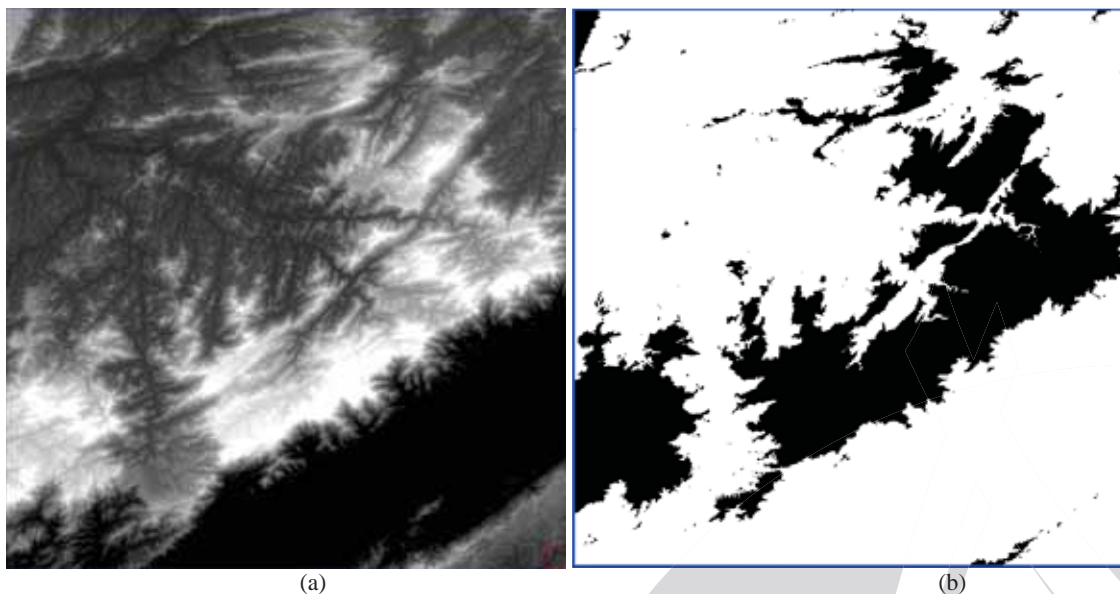


Figura 3. (a) Modelo digital de elevação e (b) ambiente de navegação criado através do modelo de elevação, considerando uma altitude de 1200 metros.

2.1 – Método RRT para o Planejamento de Rotas

Em problemas de planejamento de rotas, as RRTs são adaptadas de modo que a expansão ocorra até que um de seus nós-folha seja criado a uma distância da posição final menor ou igual a um limiar l_d . Quando isto ocorre, a posição final é incluída na árvore através de uma conexão com o nó-folha mais próximo, isto é, o nó-folha que alcançou a distância. Cada nó possui informação de seu nó antecessor. A rota é então traçada da posição final até a posição de origem e depois invertida.

Assim, a adaptação do método RRT para o problema de planejamento de rotas consiste em quatro principais alterações: verificar se a distância euclidiana entre a posição do ambiente de navegação correspondente a q_{new} e a posição de destino q_{dest} é menor ou igual a um limiar l_d ; inserir q_{dest} como um novo nó da RRT se a distância mencionada $d(q_{new}, q_{dest}) \leq l_d$; utilizar esta distância como condição de parada do algoritmo, substituindo a condição de parada do algoritmo baseada no número máximo de iterações K ; e gerar a rota de navegação.

O procedimento de criação da rota de navegação consiste na utilização de uma estrutura de dados pilha para o armazenamento das posições da rota. Como apresentado na Figura 5, cada nó possui informação sobre seu nó antecessor. Assim, começando-se o empilhamento pelo nó destino q_{dest} , é possível criar uma rota no formato de pilha, sendo que o topo da pilha é a posição de origem da navegação, isto é, q_{init} . O algoritmo de uma RRT para o planejamento de rotas é apresentado na Tabela 2.

Tabela 2. Algoritmo para o planejamento de rotas através de uma RRT

Etapas	Descrição
1	inserir a raiz q_{init} na árvore G
2	$s \leftarrow 0$
3	enquanto $s = 0$ faça
4	$q_{rand} \leftarrow \text{RAND_CONFIG}(C)$
5	$q_{near} \leftarrow \text{NEAREST_VERTEX}(q_{rand}, G)$
6	$q_{new} \leftarrow \text{NEW_CONF}(q_{near}, \Delta q)$
7	inserir o nó q_{new} na árvore G
8	inserir a aresta que une q_{near} a q_{new} na árvore G
9	se $d(q_{new}, q_{dest}) \leq l_d$ faça
10	inserir o nó q_{dest} na árvore G
11	inserir a aresta que une q_{new} a q_{dest} na árvore G
12	$s \leftarrow 1$
13	empilhar a rota R, começando de q_{dest} até o nó raiz q_{init}
14	retornar a rota R

No algoritmo para planejamento de rota: s é a variável booleana correspondente a condição de parada da RRT modificada; d é a função que mede a distância euclidiana entre o ponto q_{new} e q_{dest} ; R é uma estrutura do tipo pilha, sendo que o topo da pilha é a posição de origem da navegação, isto é, q_{init} .

3 – Resultados e Discussão

Um conjunto de experimentos de planejamento de rotas foi realizado com a aplicação do método RRT implementado neste trabalho. Os experimentos foram feitos considerando-se Δq definido por

$$\Delta q = p(long_{final} - long_{inicial}) \quad (1)$$

Em que: $p \in (0,1]$ é uma porcentagem; $long_{inicial}$ e $long_{final}$ são, respectivamente, a menor e a maior longitude do ambiente de navegação.

Alguns resultados da aplicação de RRT ao planejamento automático de rotas são apresentados na Figura 4. Nesta figura, os obstáculos do ambiente de navegação são representados pelos polígonos na cor preta. A RRT é apresentada na cor cinza e as rotas na cor vermelho

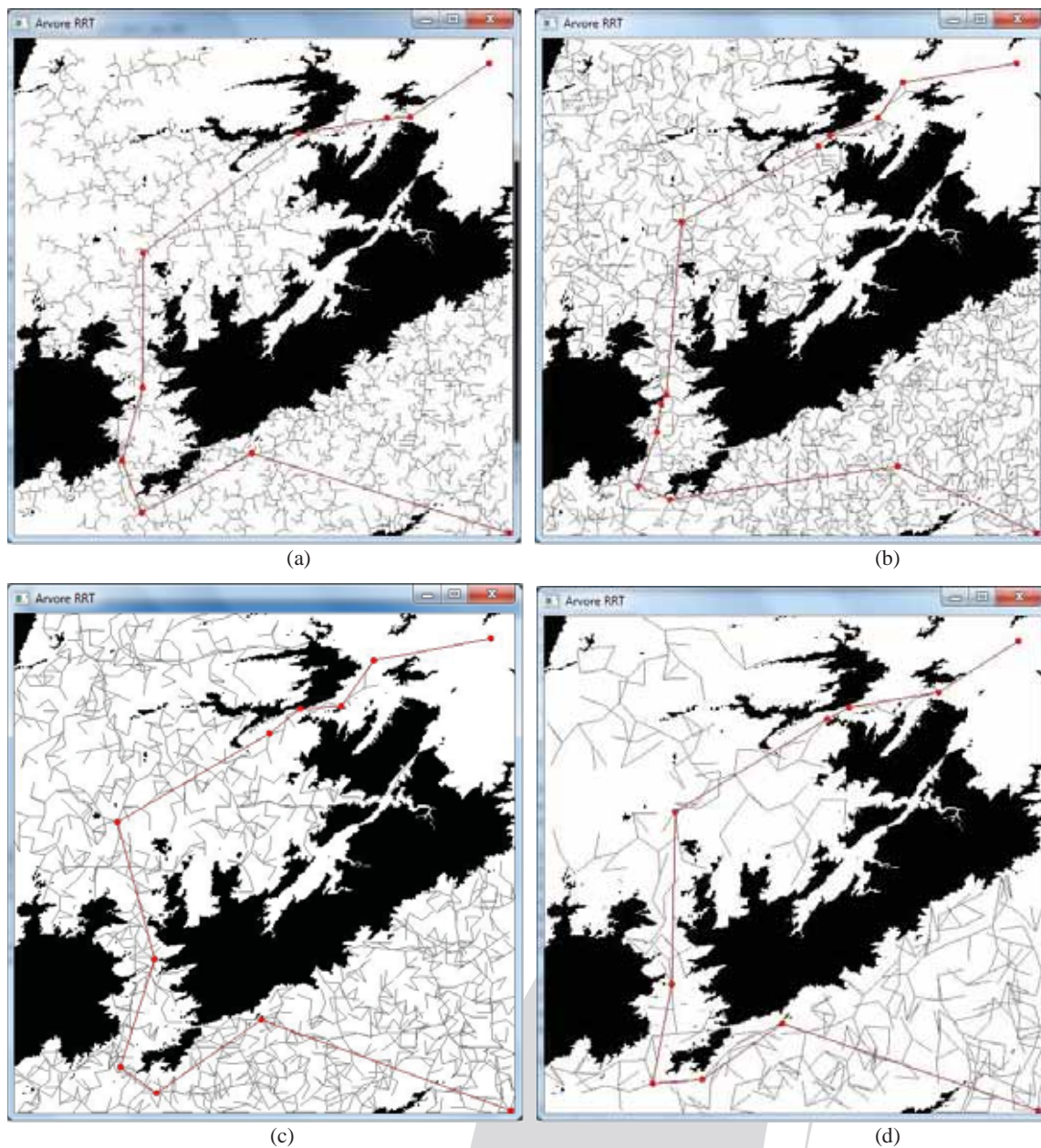


Figura 4. Rotas planejadas com RRTs, considerando: (a) $\Delta q=0.01$ (b) $\Delta q=0.02$ (c) $\Delta q=0.03$ (d) $\Delta q=0.05$

A complexidade computacional do algoritmo RRT implementado neste trabalho é $O(n_n^2 p n_c)$, em que n_n é o número de nós da RRT, que tende a aumentar a cada iteração, e n_c é o número de colunas do modelo digital de elevação. É importante ser mencionado que a complexidade computacional pode ser reduzida com a utilização de um método mais eficiente para calcular o nó q_{near} .

Após a execução do algoritmo de uma RRT para o planejamento de rotas, cada rota é simplificada com uma técnica de verificação de visibilidade, eliminando nós desnecessários da pilha R .

As rotas planejadas e apresentadas na figura 4 possuem as seguintes extensões/comprimentos, número de nós e tempo gasto: rota da Figura 4a com extensão de 236724,7 metros, 3138 nós e 5.66 segundos; rota da Figura 4b com extensão de 238239.7 metros, 2828 nós e 5.85 segundos; rota da Figura 4c com extensão de 249052.9 metros, 1511 nós e 3.32 segundos; e rota da Figura 4d com extensão de 240203.9 metros, 418 nós e 1.19 segundo.

Analisando estes resultados, pode-se verificar que há uma tendência da redução de Δq causar a redução da extensão da rota. Entretanto, a redução do Δq tende a reduzir a área explorada em cada iteração, o que pode causar o aumento do tempo de execução da implementação do algoritmo, pois há a necessidade da criação de um número maior de nós-folha para explorar o ambiente de navegação.

Uma maneira de analisar o comprimento de uma rota planejada pela RRT entre duas posições de um ambiente de navegação é compará-la com o comprimento da rota de menor extensão possível planejada entre as mesmas posições. Neste trabalho, a rota de menor comprimento possível é denominada solução ótima do problema de planejamento de uma rota entre duas posições de um ambiente de navegação. Um meio de calcular a solução ótima deste problema é a combinação do algoritmo de Dijkstra [1] com grafos de visibilidade [6]. O algoritmo de Dijkstra é um método que permite a solução ótima do problema de menor caminho ou rota entre dois nós de um grafo. Um grafo de visibilidade consiste em uma amostragem das regiões navegáveis de um ambiente de navegação. Cada nó corresponde a uma posição definida pelo vértice convexo de um obstáculo do ambiente de navegação. Os nós são conectados por arestas que não interceptam qualquer obstáculo. Um grafo de visibilidade possui a característica de sempre permitir o planejamento da rota de menor comprimento entre duas posições quaisquer de um ambiente de navegação, desde que tais posições não estejam completamente isoladas pelos obstáculos. A Figura 5 apresenta um exemplo de rota de menor extensão planejada pelo algoritmo de Dijkstra com o ambiente de navegação representado por um grafo de visibilidade e pelo algoritmo RRT modificado respectivamente. Estas rotas foram planejadas entre as posições geográficas $(lat_{init}, long_{init}) = (-22.99, -45.01)$ e $(lat_{dest}, long_{dest}) = (-22.05, -45.05)$.

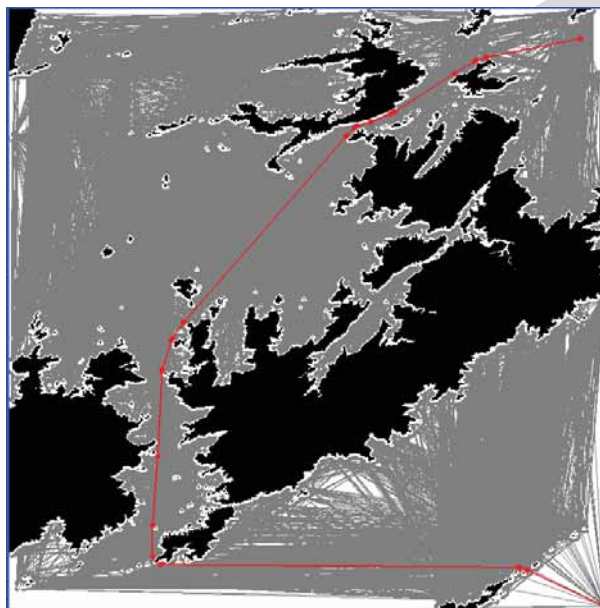


Figura 5. Exemplo de uma rota de menor extensão planejada através da aplicação do algoritmo Dijkstra a um grafo de visibilidade.

A rota na figura 5 possui a extensão de 222818.53 metros. Neste exemplo, os nós e arestas do grafo de visibilidade estão na cor cinza. A fina camada na cor branca em torno dos obstáculos é uma envoltória de segurança, com uma espessura de 400 metros. Esta envoltória permite uma distância de segurança entre os obstáculos e os nós e arestas do grafo de visibilidade.

A complexidade computacional do método empregado neste trabalho para a construção de um grafo de visibilidade é $O(n_v^2 \max(n_l, n_c))$ [6], em que n_v é o número de vértices

convexos de obstáculos do ambiente de navegação, e n_l e n_c são o número de linhas e o número de colunas do modelo digital de elevação, respectivamente. O grafo da Figura 5 possui 3920 nós e 55694 arestas. Este grafo foi construído em 301.7 segundos. Devido ao elevado custo computacional para construção de um grafo de visibilidade através de modelos digitais de elevação, estes grafos não são eficientes quando há a necessidade de remodelagem do ambiente de navegação e necessidade de replanejamento de uma rota em tempo real.

Comparando as rotas planejadas pelas RRTs com a solução ótima planejada pela combinação Dijkstra/grafos de visibilidade, pode-se perceber que as RRTs são uma interessante alternativa para o problema de replanejamento de rotas quando há alteração do ambiente de navegação, isto é, quando há a necessidade da aquisição de uma nova amostragem das regiões navegáveis do ambiente. Apesar de não garantirem a solução ótima, permitem o planejamento de rotas em tempo expressivamente inferior que o tempo necessário para o planejamento através da fusão Dijkstra/grafos de visibilidade, considerando o custo de construção do grafo de visibilidade.

4. Conclusões

Através dos resultados apresentados na Seção 3, verifica-se que é possível planejar rotas de navegação para VANTs através da aplicação de RRTs com ambientes de navegação definidos por modelos digitais de elevação. Embora uma RRT não assegure o planejamento de rotas com a menor extensão possível, ela pode ser utilizada de modo satisfatório para o replanejamento de trajetórias, quando há remodelagem do ambiente de navegação. Assim, pode-se concluir que o objetivo deste trabalho foi alcançado.

Como trabalhos futuros, serão utilizados outros ambientes de navegação definidos por diferentes modelos digitais de elevação e por diferentes configurações de altitudes de navegação e de alturas de segurança. Também será utilizado o método RRT*, que consiste em uma melhoria de uma RRT, visando ao planejamento de rotas menos extensas.

Referências

- [1] **Dijkstra, E. W.** A note on two problems in connection with graphs. *Numerische Mathematik*, v. 1, p. 269–271, 1959.
- [2] **Dubins, L. E.** On Curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, n. 79, p. 497-516, 1957.
- [3] **Hart, P. E.; Nilsson, N. J.; Raphael B.** A formal basis for the heuristic determination of minimum cost paths. *SIGART Newsletter*, n. 37, p. 28–29, 1968.
- [4] **LaValle, S. M.** *Planning algorithms*. New York: Cambridge University Press, 842 p., 2006.
- [5] **LaValle, S. M.** Rapidly-exploring random trees: A new tool for path planning. *Computer Science Dept.*, Iowa State University, October 1998.
- [6] **Medeiros, F. L. L.** Planejamento de trajetórias para veículos aéreos não tripulados usando modelagem computacional de ambientes de navegação através de grafos de visibilidade e modelos digitais de elevação. Tese de doutorado do curso de Computação Aplicada do Instituto Nacional de Pesquisas Espaciais (INPE), 238 p., 2012.
- [7] **Neto, A. A.; Macharet, D. G.; Campos, M. F. M.** On the generation of trajectories for multiple UAVs in environments with obstacles. *Journal of Intelligent and Robotic Systems*, v. 57, n. 4, p. 123-141, 2010.