

Mapeamento Multi-Robôs e Planejamento de Caminhos

Filipe Rodrigues Batista de Oliveira ¹ e Lucas Roberto Santos Avelar ²

I. INTRODUÇÃO

O trabalho proposto relaciona-se principalmente com a área de mapeamento dentro da Robótica Móvel, mas também aborda conceitos e ideias das áreas de planejamento de caminhos e sistemas multi-robôs. Os principais desafios residem em como mapear um ambiente de forma eficiente, como planejar e executar um bom caminho no ambiente planejado e como realizar uma cooperação entre os robôs que torne o processo mais rápido e eficaz.

Tem-se como objetivo utilizar mais de um robô terrestre em posições iniciais distintas afim de realizar o mapeamento conjunto de um ambiente desconhecido. Posteriormente, deseja-se utilizar o mapa gerado para planejar um caminho válido entre dois pontos escolhidos no ambiente mapeado.

Existem diversas situações nos quais seriam interessantes a cooperação entre diversos robôs afim de mapear um ambiente e, posteriormente, planejar um caminho nele. Ao realizar expedições militares ou científicas em ambientes desconhecidos e potencialmente perigosos, é interessante utilizar drones para realizar um reconhecimento (mapeamento do ambiente), identificar pontos que devem ser evitados e pontos possivelmente seguros e, finalmente, planejar um caminho que possa transportar com segurança os cientistas e militares pelo ambiente. No geral, qualquer situação que exija o transporte de uma carga valiosa por uma região que é pouco conhecida ou mesmo desconhecida pode se aproveitar da ideia de um mapeamento eficiente aliado a um planejamento de caminho.

II. TRABALHOS RELACIONADOS

Não foi encontrado pelos autores trabalhos que descrevem possíveis soluções para o problema proposto, creditamos isso pela natureza introdutória do mesmo, porém acreditamos ser possível a realização usando apenas o conteúdo visto ao longo da disciplina [1].

III. METODOLOGIA

A implementação foi feita através do Jupyter Notebook. Utilizou-se a "Remote API" do CoppeliaSim para a interação com o mesmo, além das bibliotecas: Matplotlib para a visualização dos dados, Numpy para operações vetoriais e armazenamento dos dados e a networkX para operações com grafos.

Inicialmente, gostaríamos de resaltar que como vão ser utilizados (potencialmente) vários robôs, decidiu-se criar uma classe Robo de tal forma que na mesma estivesse centralizado todos os dados necessários para se fazer o controle de cada um, pensando também em facilitar na coordenação, deixando o código mais limpo, além disso, para

uma primeira abordagem, nós decidimos usar um programa com uma única thread de forma que o laço principal das ações de cada robô fosse executado sequencialmente.

Para realizar o mapeamento conjunto de um ambiente usando um sistema multi-robôs, é fundamental que haja coordenação em relação a qual robô mapeia qual região, para se atingir esse objetivo a ideia proposta aqui seria a de subdividir verticalmente o mapa inicial em submapas (representando regiões distintas do ambiente) tal que a quantidade desses submapas seria proporcional a quantidade de robôs presentes no ambiente. Se pelo menos dois robôs caíssem em um mesmo submapa, um deles seria atribuído a responsabilidade de mapear aquela região que não estivesse atribuída a nenhum dos outros, sendo esta ideia repetida diversas vezes até que eventualmente todos os robôs estariam embuídos da tarefa de mapear um local distinto, isso é feito usando o método *escolha_regiao_para_mapear*. Além disso, para verificar se uma região já foi completamente mapeada criou-se o método *ja_mapeou_regiao* que simplesmente verifica se o número de células mapeadas é igual ao número de células do subgrid referente a mesma. Sendo que depois que cada região for completamente mapeada, devemos juntar os mapas obtidos em um único mapa, sendo isso feito na função *juntar_mapa*.

Em seguida, usamos um método simples de locomoção que foi utilizada em uma das aulas iniciais, sendo baseada em condicionais:

Se a distância lida pelo sensor até uma posição em qualquer uma das possíveis direções de movimento (esquerda, direita e frente) for menor que um valor definido a priori (alcance do sensor) o robô ou deve virar 30 graus no sentido horário (se a direção do possível obstáculo está à esquerda, caso contrário se usa o sentido anti-horário) e parar; ou se aplica uma velocidade constante em linha reta até que se encontre um obstáculo nesse sentido.

Os robôs, então, utilizam esse método de locomoção para explorar o ambiente e mapeá-lo, gerando um occupancy grid. Para o occupancy grid, vale ressaltar dois conceitos principais: primeiro, cada célula do ambiente recebe um valor que corresponde à probabilidade dessa célula estar ocupada por um obstáculo ou não. Essa probabilidade é calculada a partir da fórmula de log odds, em que utiliza-se o log de uma divisão de dois valores que somam 1 para verificar a chance de ocupação. Para definir quais células devem ser marcadas como ocupadas ou não, é utilizado o conceito de linha de Bresenham: traça-se uma linha entre o sensor do robô e o ponto final de sua leitura, e todas as células nesse espaço são calculadas com a probabilidade de estarem livres. Os robôs realizam esses processos por um tempo de 4

minutos, suficiente para gerar um occupancy grid satisfatório e bastante preciso com a realidade do ambiente.

Logo em seguida, o occupancy grid é utilizado como base para a geração de um grafo, a fim de ser definido o melhor caminho possível entre dois pontos definidos no ambiente. Para isso, cria-se o grafo inicialmente com todos os vértices do grid disponíveis, e depois inicia-se um loop para verificar quais células do grid provavelmente estão ocupadas e, portanto, devem ser removidas do grafo. Para isso, verifica-se se o valor de provável ocupação atribuído à célula é superior a um limiar (nesse caso, 0.6) e, caso seja, o vértice correspondente é removido do grafo. Após analisar todas as células e gerar o grafo final válido, basta chamar a função "shortest path" para descobrir o caminho mais curto possível entre o ponto inicial e o final, caminhando pelos vértices restantes.

Finalmente, um robô (que não participou da etapa de mapeamento) é acionado, seguindo o caminho dado na saída do algoritmo de grafo de visibilidade, no parágrafo anterior.

IV. RESULTADOS

Serão realizados dois experimentos para cada algoritmo, variando-se o cenário e posições iniciais dos robôs. Sendo utilizados apenas dois robôs nos testes; As cenas utilizadas para os testes consistem de um labirinto, de forma a testar se é possível mapear todo o ambiente e suas divisões, e de um conjunto de pilas de tal forma a analisar o comportamento do algoritmo para formas geométricas menos precisas.

Ao rodar o programa desenvolvido nas cenas vistas acima nós obtivemos os seguintes grafos de visibilidade com seus caminhos escolhidos:

A partir da imagem, percebe-se que os resultados podem ser considerados bastante satisfatórios, já que os grafos gerados possuíam vértices coincidentes com as células provavelmente livres no grid e o caminho entre o ponto inicial e o final seguiu uma configuração válida.

V. CONCLUSÃO

Tendo em vista o tempo apertado e as dificuldades encontradas, não foi possível atingir o objetivo inicial, portanto, a primeira coisa a ser feita, seria corrigir o algoritmo responsável pela coordenação, afim de se chegar ao que foi inicialmente proposto; Logo em seguida mudaríamos o paradigma de comunicação entre os robôs utilizados para troca de mensagens, já que no mundo real ter um servidor central seria mais custoso financeiramente do que realizar a comunicação local entre os robôs; a latência seria alta para o problema, devido as grandes distâncias que podem ser cobertas o que tornaria a abordagem inviável para problemas de tempo real (como este); a solução apresentada não seria escalável: conforme o número de robôs aumenta, haveria sobrecarga do servidor que faria a coordenação; além de ser dependente do mesmo, visto que se este falhar, não será possível realizar a construção do mapa, nem a coordenação entre os robôs. Além do que foi dito, seria interessante experimentar o que foi feito com times de robôs com características distintas (drones com robôs terrestres, por

exemplo), que é onde o potencial do mesmo seria realmente testado.

REFERENCES

- [1] D.G. Macharet, Curso de Robótica Móvel. UFMG. Disponível em: <https://www.youtube.com/channel/UC8rfPiPPgPsmqmt76zpG3FQ>.

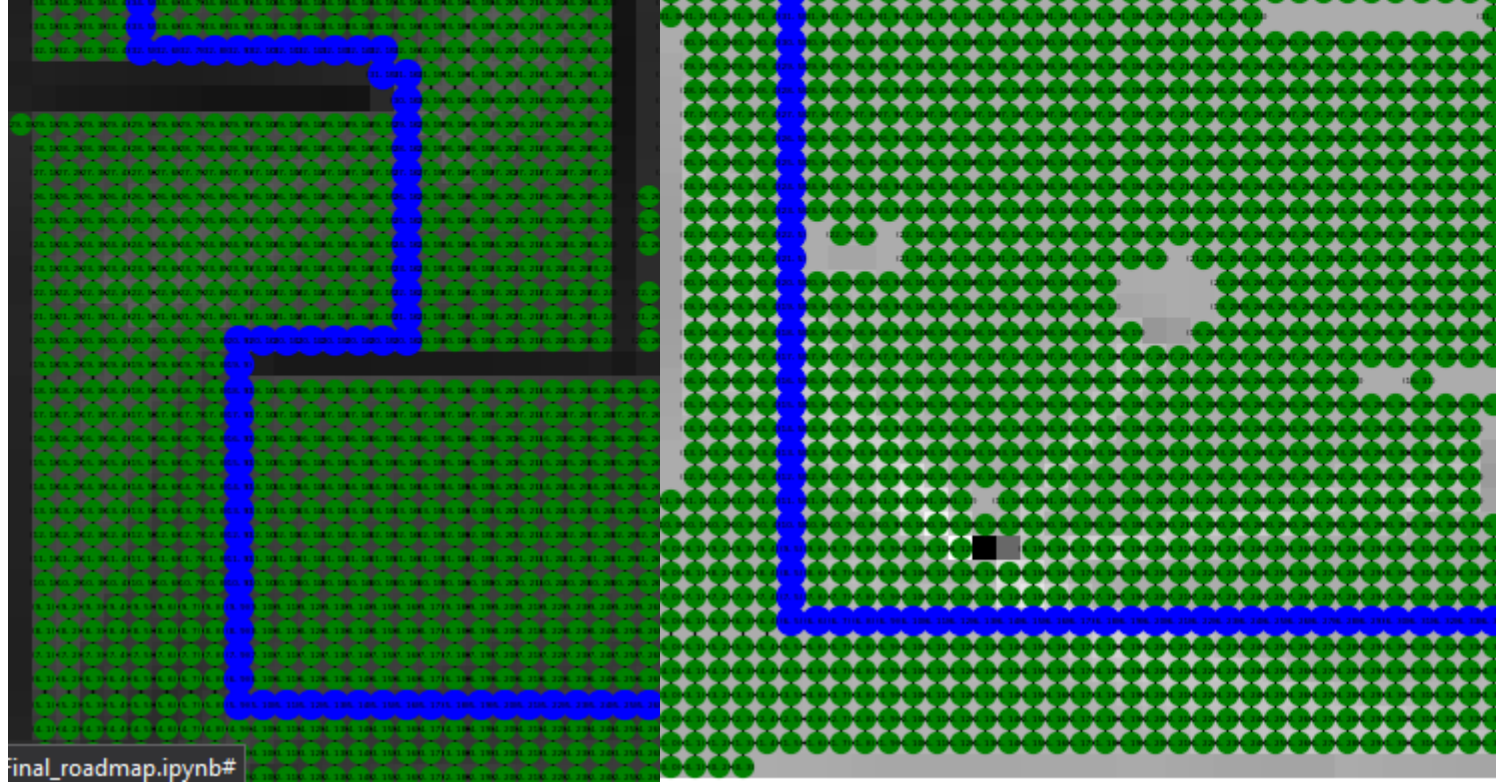


Fig. 2. Grafos de visibilidade das cenas, à esquerda cena 1 e à direita cena 2