Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

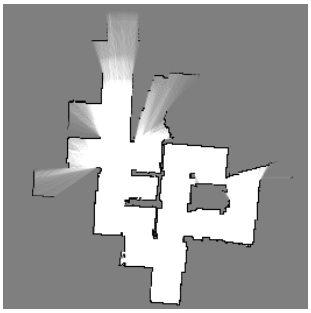
Robótica Móvel
2º Semestre de 2025
Prof. Douglas G. Macharet

Trabalho Prático 3 – Exploração e Mapeamento

Valor: 16 pontos
Data de entrega: 11/11/2025

1. Instruções:

O objetivo deste trabalho prático é familiarizar o aluno com técnicas de exploração e mapeamento, onde deverá ser implementado o algoritmo de *Occupancy Grid*. O aluno também deverá propor uma estratégia de navegação para que o robô explore o ambiente construindo o mapa. Essa estratégia pode ser simples e baseada nos algoritmos vistos em aula. Além disso, é importante lembrar que o algoritmo de *Occupancy Grid* assume que a localização do robô é conhecida (recupere usando a RemoteAPI).



Após a navegação, o programa deve salvar uma imagem com um mapa similar à figura ao lado, onde partes mais escuras representam uma maior probabilidade de ocupação e partes mais claras uma menor probabilidade.

Para melhor visualizar a aplicabilidade da técnica, adicione um pequeno ruído aleatório na leitura do laser, por exemplo, na distância medida e/ou ângulo de leitura. Caso queira, você também pode variar o ruído e verificar qual o impacto da qualidade do sensor no mapa final.

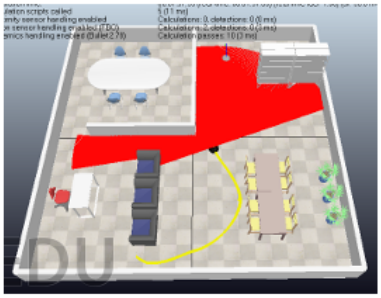
Você deve utilizar o robô diferencial Kobuki equipado com um laser. Esse robô é semelhante aos "robôs aspiradores" tradicionais, e ele já está disponível nas cenas de simulação. Caso necessário, para determinar o modelo cinemático, você pode obter algumas especificações técnicas do robô em: https://yujinrobot.github.io/kobuki/enAppendixKobukiParameters.html
Valores testados em simulação e que tiveram bons resultados (mas que você pode/deve ajustar) são:

$$L = 0.230$$
$$r = 0.035$$

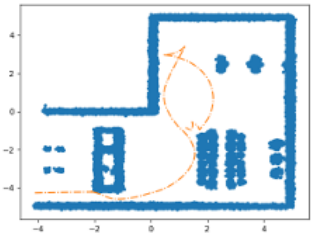
Dois cenários foram disponibilizados para realização dos experimentos, um estático e outro dinâmico. Inicialmente, utilizando o cenário estático, avalie pelo menos 3 diferentes valores de tamanho de célula (totalizando pelo menos 3 testes), por exemplo, 0,01, 0,1 e 0,5. A partir dessa avaliação inicial, decida qual valor foi melhor para os testes seguintes. **ATENÇÃO**: Os três mapas gerados devem constar na documentação. **Verifique e discuta** o impacto do tamanho das células na qualidade final do mapa.

Em seguida, com o melhor valor do tamanho da célula definido, realize pelo menos **dois** experimentos para **cada** cenário disponibilizado (estático e dinâmico). Os experimentos no mesmo cenário devem ter **diferentes posições iniciais**. Além disso, deixe o robô navegando por uma quantidade de tempo

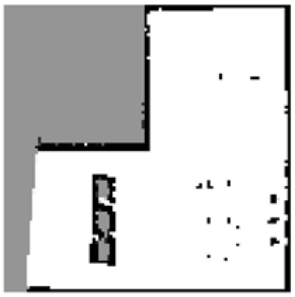
suficiente para obter um bom mapa ao final. Você também deve fazer e apresentar o plot incremental da rota do robô e da leitura do laser, igual ao que foi realizado no TP1. Na documentação, mostre o plot incremental e o Occupancy Grid gerado, por exemplo, como na figura abaixo:



(a) Simulação.



(b) Todos pontos captados e caminho do robô.



(c) Occupancy grid.

Por fim, faça uma breve análise dos resultados obtidos, discutindo a eficiência e eficácia dos algoritmos (mapeamento e estratégia de navegação).

Atenção: lembre-se que o mapa é uma representação global do ambiente, e assim como no TP1 você deverá fazer as transformações entre os diferentes referenciais (laser, robô). Além disso, você também deverá tratar a discretização da leitura no mundo contínuo para a representação no grid.

Dica: lembre-se de utilizar log-odds na etapa de atualização dos valores das células.

2. Documentação:

Entre outras coisas, a documentação deve conter:

1. Introdução: detalhamento do problema e visão geral sobre o funcionamento do programa. Deve constar o robô utilizado, seus parâmetros e as cenas utilizadas.
2. Execução: se necessário, explique brevemente as dependências do seu código e como executá-lo.
3. Navegação: detalhe a sua estratégia de navegação e controle.
2. Implementação: descrição detalhada sobre a implementação. Deve ser discutido as estruturas de dados e algoritmos utilizados (de preferência com diagramas ilustrativos), bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. Lembre-se de explicar como foi implementado o Occupancy Grid (algoritmo) e como a questão da discretização foi trabalhada. Não adicione prints do código a não ser que seja extremamente necessário.
4. Testes: descrição dos testes realizados e ilustração dos resultados obtidos (não edite os resultados). Você deve propor experimentos considerando diferentes cenários.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites, etc.

O que deve ser entregue:

Envie um arquivo ZIP no formato ‘tp3-primeironome(s).zip’, contendo os seguintes arquivos:

- Arquivo README com o nome completo e número de matrícula do aluno (dupla).
- As cenas do CoppeliaSim utilizadas.
- O código fonte do programa em Python bem formatado e comentado.
    - O programa deve ser fácil de executar, ou seja, apenas chamando-se o script.
- A documentação do trabalho em PDF bem escrita e detalhada.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza e boas práticas no programa também serão avaliados.
- Alunos de graduação podem fazer o trabalho em dupla.
- Trabalhos copiados serão penalizados conforme anunciado.

Critérios de avaliação:

- Implementação (10 pts).
    - Funcionamento correto, aplicação dos conceitos, estruturação, eficiência, ...
- Documentação (6 pts).
    - Texto (4 pts): clareza e coesão na explicação, análises, gráficos, ...
    - Apresentação (2 pts): vídeo de até 8min detalhando o trabalho e mostrando os resultados.