

Trabalho Prático 1 - Algoritmo A* com heurística de luminosidade aplicado em mapas de cavernas

Rafael Fernandes Gonçalves da Silva

Departamento de Engenharia Elétrica

Universidade Federal de Minas Gerais

Belo Horizonte, Brasil

rafaelfgs@ufmg.br

I. INTRODUÇÃO

A navegação de robôs móveis em ambientes de cavernas pode-se tornar um desafio, principalmente no aspecto de localização do robô em um dado mapa. Nesse caso, a utilização apenas de odometria das rodas do robô e de sensores de medição inercial se torna impraticável, visto que o escorregamento desses métodos acaba gerando erros que vão aumentando com o passar do tempo. Além disso, não há presença de GPS, por tratar-se de um ambiente fechado. Uma forma de contornar esse problema é utilizar sensores que possuem uma maior interação com o ambiente, como câmeras e sensores a *laser*. Dessa forma, torna-se possível uma localização mais precisa em mapas conhecidos.

O projeto de pesquisa Espelero Robô coordenado pelo Instituto Tecnológico Vale envolve a navegação de robôs móveis de forma autônoma em ambientes confinados (como cavernas, dutos e galerias), conforme mostrado em [1]. Um dos problemas encontrados no projeto relaciona-se com a localização do robô nesses ambientes. As Figuras 1 e 2 apresentam imagens da câmera interna do robô em um teste realizado em uma caverna.



Figura 1. Imagem capturada da câmera interna do robô *espeleo* da Vale durante um teste em caverna.

Um problema referente à localização utilizando câmeras está relacionado à luminosidade do ambiente, sendo que há uma maior facilidade (e precisão) estando em ambientes mais claros. Tendo em vista este problema, o presente trabalho propõe um método de planejamento de caminhos utilizando informações de luminosidade do ambiente. O método utiliza



Figura 2. Imagem capturada da câmera interna do robô *espeleo* da Vale durante um teste em caverna.

o algoritmo A* com uma heurística combinada de distância e de luminosidade aplicado em mapas de cavernas reais.

II. METODOLOGIA

Para o problema apresentado na seção anterior, foi implementado um algoritmo capaz de realizar planejamento de rotas em mapas obtidos através de arquivos no formato *png*. O método escolhido para determinar as rotas foi o A* com modificações de forma a utilizar heurísticas de distância e de luminosidade. O ambiente utilizado para realizar as simulações foi o *StageROS*, onde é utilizado um arquivo de mapa do tipo *.world*, contendo os pontos inicial e objetivo e o nome do arquivo de imagem do mapa.

A implementação do código foi realizada na linguagem *Python*, seguindo um algoritmo principal dado pela sequência de passos dada pelo Algoritmo 1:

Algorithm 1 Principal

- 1: Inicia as funções do ROS
 - 2: Discretiza o mapa de obstáculos e de luminosidade
 - 3: **while not** *rospy.shutdown* **do**
 - 4: Verifica mudanças nas posições do robô e objetivo
 - 5: **if** Houver alguma mudança **then**
 - 6: Aplica o método A*
 - 7: Determina o próximo ponto x_{next} e y_{next}
 - 8: Determina o sinal de controle U_x e U_y
 - 9: Determina v e w do *feedback linearization*
 - 10: Publica as velocidades
-

Para a discretização do mapa realizado na segunda linha, são dados como entrada dois arquivos e imagem: o primeiro contendo uma imagem binária do mapa a ser utilizado (sendo 0 os obstáculos e 1 o espaço livre); e o segundo contendo uma imagem representativa da luminosidade do mapa. Neste ponto, é também realizado um aumento dos obstáculos utilizando o tamanho do robô como parâmetro. A saída deste procedimento é representada por um *grid* contendo os obstáculos e outro com os valores médios de luminosidade em cada célula correspondente, ambos de um tamanho previamente especificado.

Na criação da imagem representativa de luminosidade, foi implementado um pequeno algoritmo capaz de realizar uma convolução entre a imagem binária do mapa equivalente e uma máscara radial de tamanho especificado. A saída foi determinada de forma a ficar de tamanho coincidente com a imagem do mapa, contendo valores entre 0 e 50, algo condizente com os valores de *lux* encontrados em ambientes escuros como cavernas. Com isso, o resultado final é uma representação de luminosidade para o caso em que lugares mais abertos e longe de obstáculos possuem mais iluminação. A utilização do *grid* como heurística deu-se através da formula (1), onde valores menores correspondem a uma maior prioridade na fila e k_{lumin} é um fator de importância da heurística.

$$grid_L[i, j] = (50 - img_L[i, j])^{k_{lumin}} \quad (1)$$

A aplicação do método A*, executada na sexta linha, é realizada da forma mostrado pelo Algoritmo 2.

Algorithm 2 Principal

- 1: Cria o *grid* A com os índices do grafo
 - 2: Inicializa os *grids* F , G , H e L
 - 3: Determina os valores de F , G , H e L do ponto inicial
 - 4: **while** não chegar ao objetivo **or** existir valor de F **do**
 - 5: Determina os índices do valor atual de F
 - 6: **for each** índice i do valor de F **do**
 - 7: Determina os vizinhos de i
 - 8: Determina F , G , H e L dos vizinhos
 - 9: Salva o índice i nos pontos dos vizinhos em A
 - 10: Determina o próximo valor de F
 - 11: Encontra o grafo do objetivo ao robô usando o *grid* A
-

A matriz G utilizada representa a busca *Dijkstra*, a matriz H corresponde a distância de *Manhattan* (modificada para considerar diagonais), a matriz L representa a heurística de luminosidade e a matriz F é dada como a soma das anteriores. É importante ressaltar que o algoritmo do A* implementado realiza uma busca utilizando vizinhança 8, tornando quase obrigatório o aumento dos obstáculos citado anteriormente.

Foram utilizados quatro mapas distintos, sendo o primeiro para a realização de testes e os restantes representados por mapas de cavernas reais, para verificar a eficácia do método. Estes foram chamados de *test*, *catacombs cave*, *lavabeds cave* e *oregon cave* e estão demonstrados nas subseções seguintes. Em [2] está demonstrado um estudo piloto realizado sobre estes três últimos mapas.

A. Mapa test

Para realizar alguns testes de discretização, criação da imagem representativa de luminosidade, aumento dos obstáculos, aplicação da heurística de luminosidade, foi utilizado um mapa e teste, denominado *test*, sendo da forma demonstrada pela Figura 3.

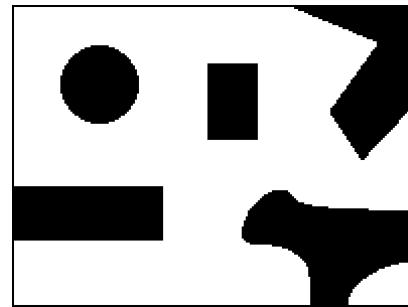


Figura 3. Mapa simples *test* criado com intuito de realizar testes do algoritmo.

Foi determinado que o mapa teria 16 x 12 metros, com um arquivo *png* de tamanho 200 x 150.

A variação de discretização com o aumento de obstáculos produziu os resultados apresentados pela Figura 4.



Figura 4. Aplicação do aumento de obstáculos no mapa *test* utilizando as resoluções 16x12, 80x60 e 200x150.

Para a criação da representação da luminosidade, foi utilizado uma filtragem com uma máscara radial de tamanho 40 (pixels). O resultado obtido está mostrado na Figura 5.

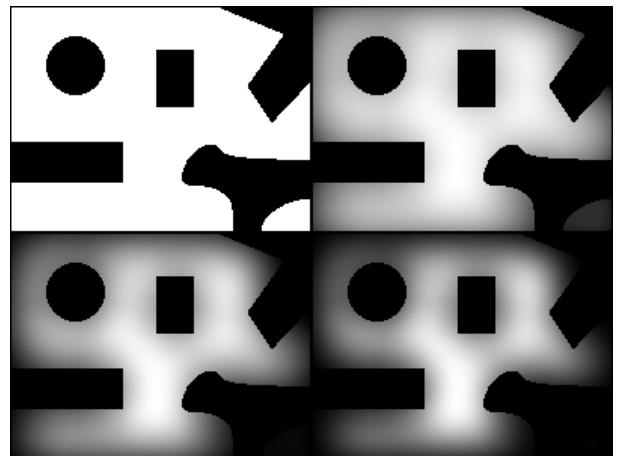


Figura 5. Imagem criada para simular a luminosidade no mapa *test* variando a constante de luminosidade nos valores 0.0, 0.5, 1.0 e 1.5.

B. Mapa *catacombs cave*

O primeiro mapa real utilizado foi o *catacombs cave*. Este mapa apresenta ser mais homogêneo que os outros, possuindo passagens mais largas, o que permite a passagem de robôs em grande parte da caverna.

O arquivo *png* utilizado possui uma resolução de 2128 x 832 pixels e um tamanho em escala de 608 x 238 metros. As Figuras 6 e 7 representam respectivamente o mapa da caverna disponibilizado *online* e a sua transformação em uma imagem binária.

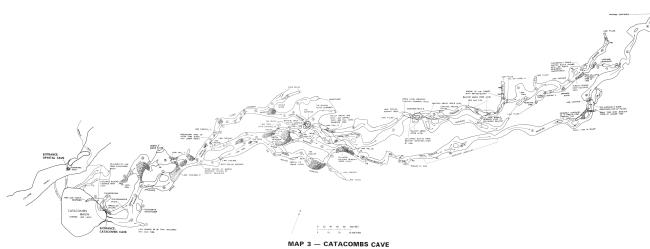


Figura 6. Mapa *catacombs cave* disponível em https://beej.us/labe/images/catacombs_large.png.

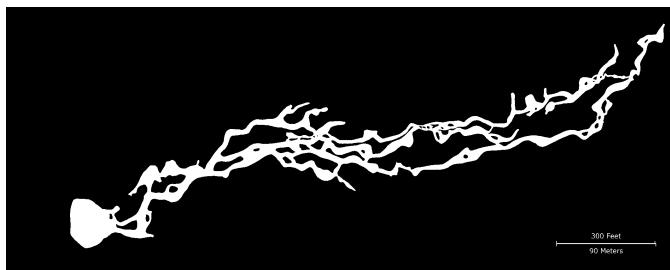


Figura 7. Transformação do mapa *catacombs cave* para valores binários.

Utilizando um filtro radial com tamanho de 70 pixels, obteve-se uma representação da luminosidade conforme mostrado na Figura 8.

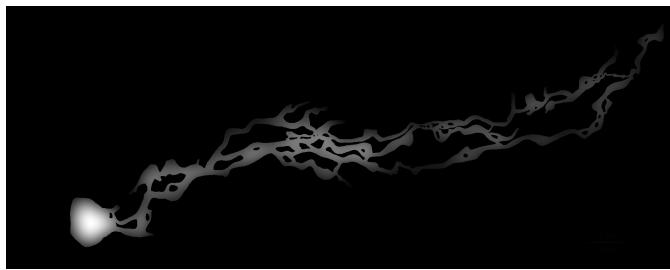


Figura 8. Imagem criada para simular a luminosidade no mapa *catacombs cave*.

C. Mapa *lavabeds cave*

O mapa *lavabeds cave* aqui utilizado apresenta passagens mais estreitas, impossibilitando a passagem de robôs em diversos pontos da caverna.

A resolução do arquivo *png* foi de 2244 x 820 pixels, com um tamanho aproximado de 495 x 181 metros. Uma representação do seu formato e sua transformação em imagem binária estão apresentada nas Figuras 9 e 10, respectivamente.

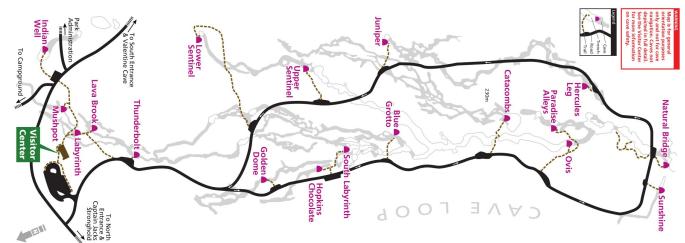


Figura 9. Mapa *lavabeds cave* disponível em <http://npmaps.com/wp-content/uploads/lava-beds-cave-loop-map.jpg>.

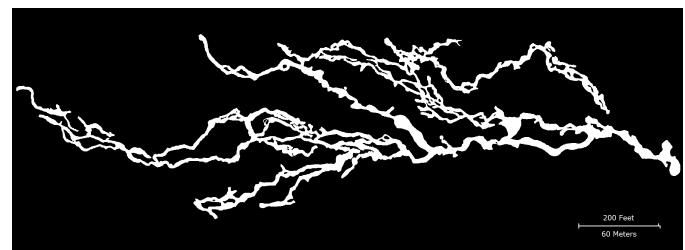


Figura 10. Transformação do mapa *lavabeds cave* para valores binários.

Para este mapa, foi utilizado um filtro radial de tamanho de 60, alcançando a representação da luminosidade mostrada na Figura 11.



Figura 11. Imagem criada para simular a luminosidade no mapa *lavabeds cave*.

D. Mapa *oregon cave*

Por fim, o último mapa utilizado neste trabalho foi o *oregon cave*. Este apresenta uma maior complexidade, com vários caminhos que fecham uma volta.

A resolução da imagem utilizada foi igual à 2244 x 820 para uma mapa de 495 x 181 metros. A representação e a imagem em forma binária desta caverna estão respectivamente apresentada nas Figuras 9 e 10.

Neste mapa, foi utilizado um filtro maior com tamanho igual a 90. O resultado encontrado é mostrado na Figura 14.

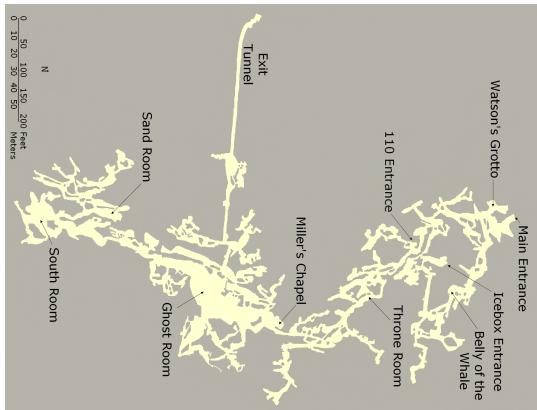


Figura 12. Mapa de *oregon cave* disponível em https://upload.wikimedia.org/wikipedia/commons/7/73/Oregon_Caves_Cave_Map.png.

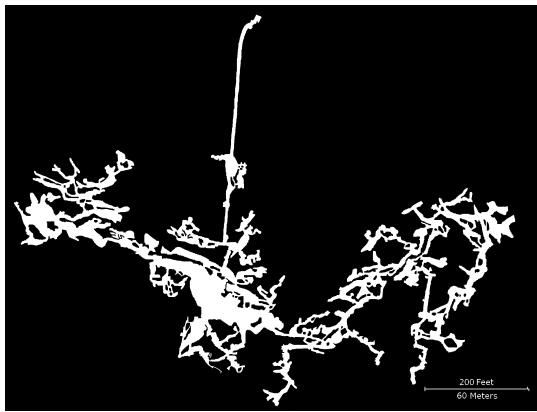


Figura 13. Transformação do mapa *oregon cave* para valores binários.



Figura 14. Imagem criada para simular a luminosidade no mapa *oregon cave*.

III. RESULTADOS E DISCUSSÃO

Os resultados com os mapas descritos anteriormente estão demonstrados nas subseções seguintes.

A. Mapa test

Os parâmetros utilizados para os testes deste mapa estão mostrados na Tabela I.

Tabela I

VALORES DOS PARÂMETROS UTILIZADOS NOS RESULTADOS NO MAPA *test*.

Parâmetro	Valor	Unidade
Resolução da Imagem	200 x 150	pixels
Tamanho do Mapa	16 x 12	metros
Resolução do Mapa	80 x 60	pixels
Tamanho do Robô	0.75	metros
Fator de Luminosidade	1.0	-
Ponto de Controle	0.1	metros
Constantes de Convergência (kp e kd)	1.0 e 0.2	-
Distância de Tolerância	0.1	metros
Tamanho do Filtro	40	pixels

Dado um ponto inicial em (-6.5, -4.5) e um ponto objetivo em (4.5, -0.5) os resultados encontrados variando-se a discretização em três valores: 16x12, 80x60 e 200x150. Estes estão apresentados na Figura 15 e na Tabela II.

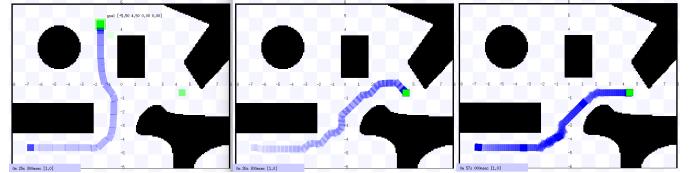


Figura 15. Resultados obtidos no mapa *test* variando-se a resolução do mapa em 16x12 (à esquerda), 80x60 (ao meio) e 200x150 (à direita).

Tabela II

RESULTADOS ENCONTRADOS NO MAPA *test* VARIANDO-SE A RESOLUÇÃO DO MAPA.

Resolução	Tempo do grid	Tempo do A*	Exploração do A*	Tempo do Robô
200 x 150	1.28 s	1.50 s	76.9%	0.56
80 x 60	0.21 s	0.05 s	28.5%	0.26
16 x 12	0.01 s	0.02 s	27.0%	0.25

Primeiramente, é importante ressaltar que para a resolução 16x12 não foi possível encontrar solução e o ponto objetivo teve de ser deslocado para o ponto (-1.5, 4.5). Isso mostra que baixos valores de resolução podem atrapalhar o planejamento. Por outro lado, utilizar uma discretização com alta resolução, implica em um controle com pontos muito próximos, o que reduz a velocidade do robô. Além disso, o código implementado mostrou-se muito sensível à resolução do mapa, com um tempo de execução que cresce exponencialmente com o aumento da resolução.

Trabalhando agora com a luminosidade do ambiente, foi variado do fator de luminosidade com os valores 0.0, 0.5, 1.0 e 1.5. A Figura 16 e a Tabela III demonstram o resultado para essa variação.

Para valores de k_{lumin} iguais a zero, a trajetória do robô corresponde à ótima. À medida que o valor é aumentado, o robô deixa o caminho ótimo e é induzido a passar por caminhos com maior luminosidade (ver Figura 5).

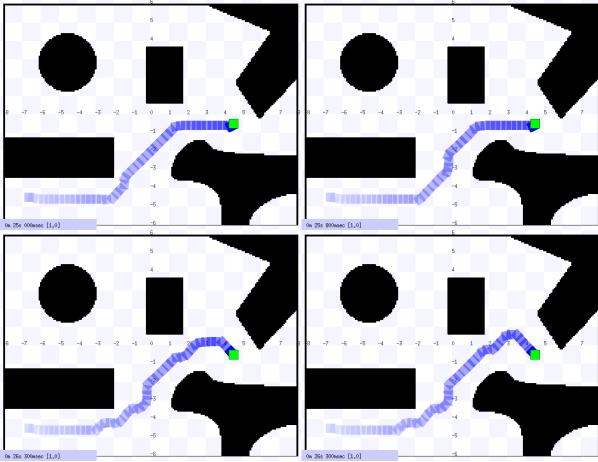


Figura 16. Resultados obtidos no mapa *test* variando-se o valor de k_lumin em 0.0 (acima à esquerda), 0.5 (acima à direita), 1.0 (abaixo à esquerda) e 1.5 (abaixo à direita).

Tabela III

RESULTADOS ENCONTRADOS NO MAPA *test* VARIANDO-SE O VALOR DE k_lumin .

Fator de Luminosidade	Tempo do A*	Exploração do A*
0.0	0.19 s	34.8%
0.5	0.06 s	32.6%
1.0	0.05 s	28.5%
1.5	0.07 s	38.8%

Um resultado interessante alcançado com esse método foi a redução do tempo gasto no A*, juntamente com a porcentagem do mapa que foi explorado pelo método. Além da priorização de busca em direção ao objetivo, o método acaba por priorizar também a busca em regiões com maior luminosidade, reduzindo, assim, o número de células visitadas. Porém, caso o valor de k_lumin for exageradamente alto, a busca em direção ao objetivo perde relevância, aumentando a porcentagem do mapa explorado.

B. Mapa *catacombs cave*

Utilizando o mapa *catacombs cave*, foram gerados resultados para diferentes valores de resolução do mapa e do fator e luminosidade. Os parâmetros utilizados para esse mapa estão mostrados na Tabela IV.

Tabela IV

VALORES DOS PARÂMETROS UTILIZADOS NOS RESULTADOS DO MAPA *catacombs cave*.

Parâmetro	Valor	Unidade
Resolução da Imagem	2128 x 832	pixels
Tamanho do Mapa	608 x 238	metros
Resolução do Mapa	1064 x 416	pixels
Tamanho do Robô	1.0	metros
Fator de Luminosidade	1.0	-
Ponto de Controle	0.1	metros
Constantes de Convergência (kp e kd)	1.0 e 0.2	-
Distância de Tolerância	0.1	metros
Tamanho do Filtro	70	pixels

Utilizando os pontos inicial e final respectivamente em (-67.5, 10.5) e (-228.5, -75.5), foram variadas as resoluções do mapa em 2244x820, 1122x410 e 561x205. Os resultados encontrados estão mostrados na Figura 15 e na Tabela V.

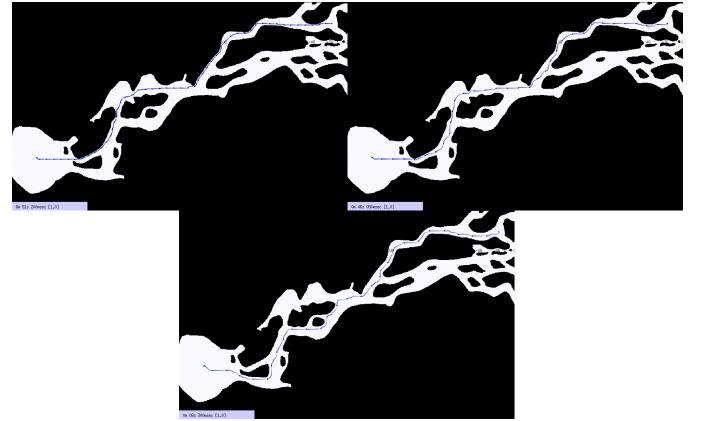


Figura 17. Resultados obtidos no mapa *catacombs cave* variando-se a resolução do mapa em 2244x820 (acima à esquerda), 1122x410 (acima à direita) e 561x205 (abaixo).

Tabela V
RESULTADOS ENCONTRADOS NO MAPA *catacombs cave* VARIANDO-SE A RESOLUÇÃO DO MAPA.

Resolução	Tempo do grid	Tempo do A*	Exploração do A*	Tempo do Robô
2244 x 820	106 s	486 s	34.2%	8:24
1122 x 410	19.9 s	23.9 s	35.1%	6:29
561 x 205	5.2 s	1.1 s	27.5%	4:28

Dado que a resolução da imagem deste mapa é muito maior que a do mapa de teste, a diferença observada na variação da resolução da discretização foi muito mais evidente. O tempo gasto para a discretização para o mapa em resolução completa foi próximo à dois minutos, enquanto que a busca do A* gastou mais de oito minutos. Além disso, redução da resolução em 4x neste caso, implicou em uma redução de 20x no tempo do grid e um mais de 400x no A*. Porém, esta resolução mais baixa provocou a desconexão de alguns caminhos, induzindo o robô a desviar dos obstáculos pelo pior caminho, algo que deve ser levado em consideração.

Com valores do fator de luminosidade variados em 0.0, 0.5, 1.0 e 1.5, foram obtidos os resultados apresentados pela Figura 18 e pela Tabela VI. Nesse caso, o ponto inicial escolhido foi (67.5, 10.5) e o ponto objetivo (-228.5, -75.5).

Tabela VI
RESULTADOS ENCONTRADOS NO MAPA *catacombs cave* VARIANDO-SE O VALOR DE k_lumin .

Fator de Luminosidade	Tempo do A*	Exploração do A*
0.0	13.6 s	14.2%
0.5	11.5 s	14.0%
1.0	9.4 s	10.6%
1.5	5.4 s	6.9%

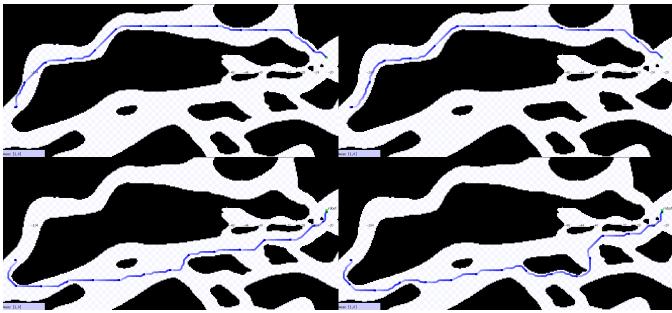


Figura 18. Resultados obtidos no mapa *catacombs cave* variando-se o valor de *k_lumin* em 0.0 (acima à esquerda), 0.5 (acima à direita), 1.0 (abaixo à esquerda) e 1.5 (abaixo à direita).

Assim como no mapa de teste, o tempo gasto e a porcentagem de exploração do método A* ficaram reduzidos para maiores valores do fator de luminosidade, mostrando que há uma maior priorização da busca por pontos de maior luminosidade e em direção ao objetivo. Com esse resultado, foi possível observar que para valores de *k_lumin* maiores que 1.0, houve uma mudança total no caminho encontrado, sendo ainda maior para *k_lumin* igual à 1.5.

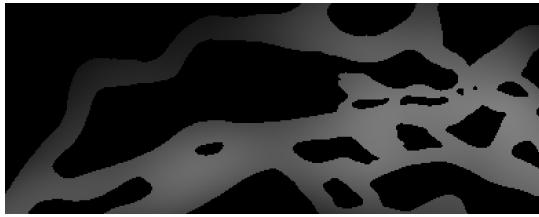


Figura 19. Simulação da luminosidade da região do teste de variação de *k_lumin* realizado no mapa *catacombs cave*.

C. Mapa *lavabeds cave*

Para validar o funcionamento do método em outros mapas, foram realizados testes nos mapas mais complexos, utilizando a resolução completa da imagem como discretização.

Com os parâmetros dados pela Tabela VII, os resultados obtidos estão apresentados na Figura 20 e na Tabela VIII.

Tabela VII

VALORES DOS PARÂMETROS UTILIZADOS NOS RESULTADOS NO MAPA *lavabeds cave*.

Parâmetro	Valor	Unidade
Resolução da Imagem	2244 x 820	pixels
Tamanho do Mapa	495 x 181	metros
Resolução do Mapa	2244 x 820	pixels
Tamanho do Robô	1.0	metros
Fator de Luminosidade	1.0	-
Ponto de Controle	0.1	metros
Constantes de Convergência (<i>kp</i> e <i>kd</i>)	1.0 e 0.2	-
Distância de Tolerância	0.1	metros
Tamanho do Filtro	60	pixels
Ponto Inicial	(-142.5, -19.5)	metros
Ponto Objetivo	(62.5, -16.5)	metros

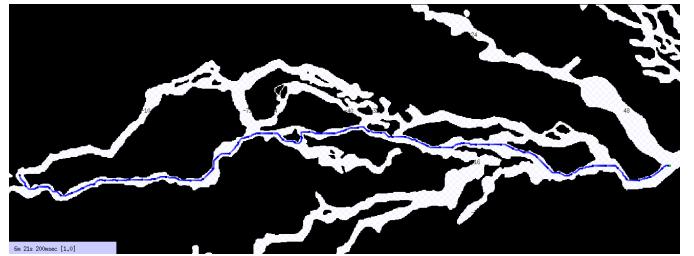


Figura 20. Resultado obtido no mapa *lavabeds cave*.

Tabela VIII
RESULTADOS ENCONTRADOS NO MAPA *lavabeds cave*.

Tempo do grid	Tempo do A*	Exploração do A*	Tempo do Robô
111 s	313 s	21.3%	9:36

Pode-se notar que o tempo para a discretização do mapa e principalmente da busca do A* mostra-se um problema para este caso, alcançando uma soma de aproximadamente sete minutos.

D. Mapa *oregon cave*

Utilizando os parâmetros da Tabela VII, os resultados obtidos para o mapa *oregon cave* estão mostrados na Figura 21 e na Tabela X.

Tabela IX
VALORES DOS PARÂMETROS UTILIZADOS NOS RESULTADOS NO MAPA *oregon cave*.

Parâmetro	Valor	Unidade
Resolução da Imagem	2212 x 1596	pixels
Tamanho do Mapa	312 x 236	metros
Resolução do Mapa	2212 x 1596	pixels
Tamanho do Robô	1.0	metros
Fator de Luminosidade	1.0	-
Ponto de Controle	0.1	metros
Constantes de Convergência (<i>kp</i> e <i>kd</i>)	1.0 e 0.2	-
Distância de Tolerância	0.1	metros
Tamanho do Filtro	90	pixels
Ponto Inicial	(94.5, -59.5)	metros
Ponto Objetivo	(110.5, 5.5)	metros

Tabela X
RESULTADOS ENCONTRADOS NO MAPA *oregon cave*.

Tempo do grid	Tempo do A*	Exploração do A*	Tempo do Robô
151 s	586 s	10.1%	5:22

Assim como no mapa *lavabeds cave*, o tempo para a discretização deste mapa e da busca do A* mostrou-se novamente um problema para este caso, com um tempo total superior a doze minutos. Neste caso pode-se notar que o tempo total de planejamento da trajetória foi maior que o dobro do tempo de movimento do robô até o ponto objetivo.

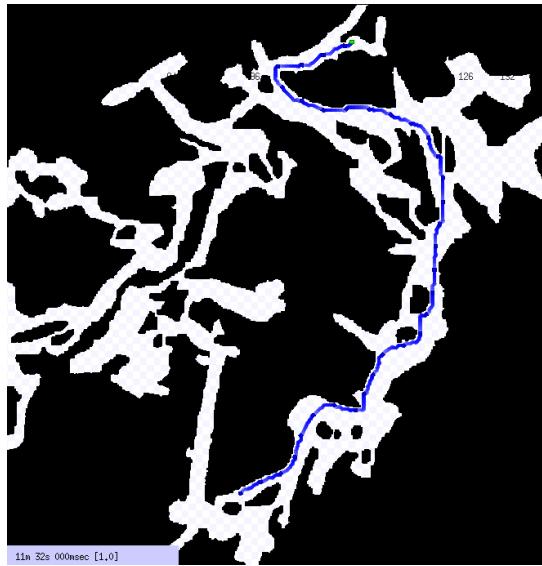


Figura 21. Resultado obtido no mapa *oregon cave*.

IV. OUTRAS APLICAÇÕES

Através dos resultados encontrados, foi observado uma aplicação interessante para a implementação utilizada. Em mapas de cavernas há uma maior facilidade de locomoção nos pontos centrais dos corredores, visto que próximos às paredes o terreno pode estar mais acidentado ou conter pedras, como mostrado na Figura 1.

De forma a contornar esse problema, foi realizado um teste no mapa *catacombs cave* semelhante à Figura 18, porém dessa vez utilizando uma simulação da luminosidade ambiente gerada com um filtro de tamanho 10. Assim como mostrado na Figura 5, os pontos próximos os obstáculos passar a ter valores menores de luminosidade, conforme mostrado na Figura 22.

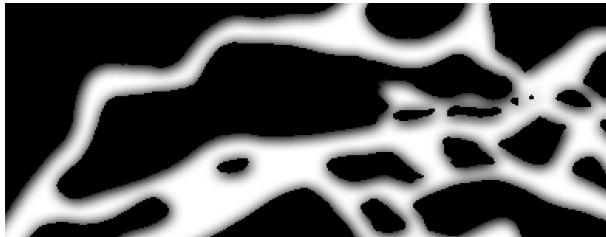


Figura 22. Representação da luminosidade encontrada utilizando um filtro radial de tamanho 10 no mapa *catacombs cave*.

Por possuírem maiores valores de luminosidade, os pontos centrais aos corredores passam a ter prioridade na busca do método A*, sendo então criados caminhos distantes aos obstáculos. Isso está apresentado pela Figura 23, onde o caminho percorrido pelo robô se assemelha a um caminho criado pelas regiões de Voronoi

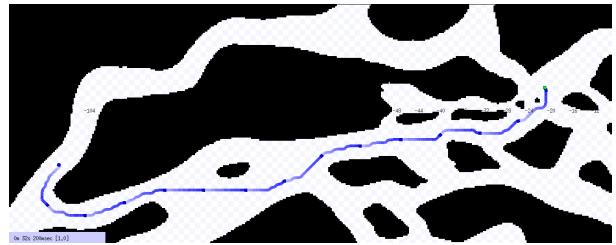


Figura 23. Resultado obtido no mapa *catacombs cave* utilizando a representação de luminosidade criada com um filtro radial de tamanho 10.

V. CONSIDERAÇÕES FINAIS

O método utilizado alcançou resultados satisfatórios, levando em consideração a situação proposta na seção I, onde é possível obter um mapa estático do ambiente e da sua respectiva luminosidade. Foi possível observar que um dado caminho, com uma distância não muito superior ao ótimo e localizado em uma região com maior luminosidade, possui uma maior prioridade no planejamento.

De modo a aprimorar a qualidade do trabalho, é necessário um estudo maior da luminosidade de cavernas, de forma a realizar aplicações com dados práticos. Isso pode ser realizado durante uma exploração do ambiente utilizando um dispositivo capaz de reconhecer a luminosidade local, como, por exemplo, um luxímetro ou até mesmo a própria câmera do robô.

Tendo em mãos essa medida em tempo real da luminosidade ambiente, é possível alterar a implementação de forma a criar um mapa de luminosidade dinâmico, de forma a ser possível decidir qual caminho escolher em casos como ao apresentado na Figura 2, onde à direita há um caminho sem iluminação e à esquerda há um caminho com iluminação artificial, dando preferência para este último.

Por fim, o problema enfrentado que pode ser considerado o mais importante é o tempo de execução do algoritmo utilizado, sendo este muito alto em algumas configurações. Para contornar esse problema, há diversos métodos de otimização, como, por exemplo, a utilização de *grids* variáveis, como o *quadtree*; a utilização de *check points* para restringir a busca para parte do ambiente; a utilização de linguagens de mais baixo nível para reduzir o tempo de execução; entre outros.

REFERÊNCIAS

- [1] G. M. Freitas, “Dispositivo robótico para inspeção de ambientes restritos e confinados,” Universidade Federal de Minas Gerais, Relatório Parcial do Projeto, 2019.
- [2] S. Thomas, “Monitoring cave entrance communities and cave environments in the klamath network,” National Park Service, Pilot Study Results, 2010.