

INFORMATION WINDOW: UMA NOVA ABORDAGEM PARA EXPLORAÇÃO DE FRONTEIRAS

RAFAEL GONÇALVES COLARES*, LUIZ CHAIMOWICZ*

*Departamento de Ciência da Computação
PPGCC - UFMG
Belo Horizonte - MG - Brasil

Emails: rcolares@dcc.ufmg.br, chaimo@dcc.ufmg.br

Abstract— This paper presents a novel approach to the problem of frontier exploration. Instead of focusing on exploration speed, the novel approach focus on increasing the system's information. Using a sliding window to analyze the region around a map cell, it will be proposed a new map evaluation method using an utility function. Several simulation tests were executed to compare the proposed approach with a tradicional frontier explorer approach. Both approaches were tested in relation to exploration time, map quality and information gain, and the results shows that the proposed approach achieves the proposed goal.

Keywords— Frontier Explorer, SLAM, Planning.

Resumo— Este trabalho apresenta uma nova abordagem para o problema de exploração de fronteiras. Ao invés de priorizar a velocidade de exploração, a nova abordagem prioriza a aquisição informação do sistema. Usando uma janela deslizante para analisar a região em torno de uma célula do mapa, será proposto um novo método de avaliação do mapa através de uma função de utilidade. Foram executados testes em simulação para realizar uma comparação com uma abordagem de exploração de fronteiras tradicional. As abordagens foram avaliadas em relação ao tempo de exploração, qualidade do mapa e o ganho de informação, e os resultados mostram que a nova abordagem atinge o objetivo proposto.

Palavras-chave— Exploração de Fronteiras, SLAM, Planejamento.

1 Introdução

Explorar um ambiente desconhecido é um pré-requisito fundamental para um robô autônomo. Em situações em que um mapa do ambiente não está disponível, o robô precisa explorar este ambiente, enquanto aplica uma técnica de *Simultaneous Localization and Mapping* (SLAM) de forma a construir o mapa do mesmo. Para tornar a exploração mais eficiente é preciso acrescentar uma etapa de planejamento no processo do SLAM, transformando assim o SLAM em *Active SLAM* (ASLAM) (Stachniss et al., 2004).

O componente de planejamento do ASLAM deve ser capaz de analisar diversas informações, como o mapa construído até o momento, as informações sensoriais atuais, o estado do robô e, baseado nelas, decidir qual ação o robô deve tomar. Essas ações a serem tomadas pelo robô podem procurar melhorar a qualidade do mapa, explorar o ambiente ou reduzir erros de localização.

O objetivo principal da exploração é obter o máximo de informação do ambiente para construir um mapa confiável. Neste cenário, informação significa descobrir novas áreas do ambiente e mapeá-las com boa qualidade. Para atingir esse objetivo, a técnica mais comum utilizada na literatura é a “exploração de fronteiras” (Yamauchi, 1997).

A exploração de fronteiras consiste em encontrar regiões onde existe o limite entre espaços livres e espaços não explorados do mapa e explorá-las. A técnica se baseia na ideia de que o maior ganho de informação acontece quando o robô ex-

plora regiões desconhecidas. A Figura 1 demonstra as fronteiras demarcadas em um mapa.

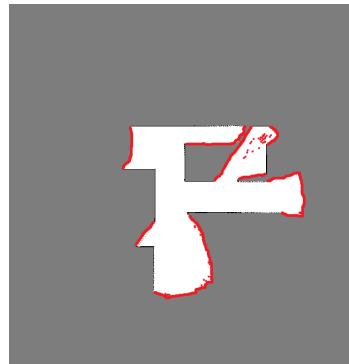


Figura 1: Mapa de um ambiente com as fronteiras marcadas em vermelho.

Após encontrar as fronteiras é preciso decidir um método de explorá-las. Uma forma de realizar essa exploração é simplesmente levar em consideração a distância do robô para cada fronteira e explorar aquela que for mais próxima.

Neste trabalho será proposta uma nova abordagem para detectar e explorar fronteiras baseada na informação do mapa e não na distância das fronteiras, chamada de *Information Window* (IW). O objetivo deste método é obter o maior ganho de informação do mapa para cada planejamento feito. Para realizar isto, uma janela é deslizada pelo mapa obtendo informações sobre fronteiras e sobre as incertezas do mapa, gerando um mapa de informação. A partir deste mapa, a nossa

abordagem decide para onde o robô deve navegar na próxima ação para aumentar a quantidade de informação do sistema.

Este trabalho está assim organizado. A Seção 2 apresenta uma definição de exploração de fronteiras, bem como uma breve revisão de trabalhos sobre exploração de fronteiras. A Seção 3 explica a abordagem proposta. Os resultados obtidos são discutidos na Seção 4 e a Seção 5 apresenta conclusões e melhorias para a abordagem.

2 Exploração de fronteiras

A exploração de fronteiras consiste em detectar regiões de fronteiras do mapa estimado até o momento e escolher uma fronteira para explorar. O trabalho que fundamentou a exploração de fronteiras foi Yamauchi (1997). Neste trabalho, o autor formaliza o que é uma fronteira, caracterizando as células do grid de ocupação do mapa estimado como:

- **células desconhecidas:** são células que não foram exploradas pelo robô;
- **células ocupadas:** são células que possuem obstáculos;
- **células livres:** são células que não possuem obstáculos.

Com estas definições é possível determinar o que é uma **fronteira**. Fronteira, então, é um conjunto de **células desconhecidas** que possuem pelos menos uma **célula livre** como vizinha. Assim, um algoritmo de detecção de fronteiras precisa apenas percorrer o grid de ocupação procurando por células de fronteira. Geralmente, um algoritmo de detecção de bordas é utilizado.

Yamauchi (1997) ainda propôs um método de exploração conhecido por *near-frontier explorer* (NFE), onde, após detectar todas as fronteiras do mapa, o robô explora aquela que é mais próxima. O NFE preocupa-se apenas com a distância das fronteiras, não levando em consideração o ganho de informação possível por visitar a fronteira.

Ao navegar para uma fronteira, o robô continuamente executa um algoritmo de SLAM. Dessa forma, o robô pode ter mapeado completamente uma região antes de chegar na fronteira planejada. Para resolver esta questão, Holz et al. (2010) propuseram uma abordagem que checa constantemente se a fronteira planejada ainda existe, e, caso já tenha sido explorada, escolhe uma nova fronteira para explorar. Holz et al. (2010) ainda definiram que, segmentar o mapa em regiões e dar preferência para regiões próximas ao robô, favorece a exploração.

Faigl et al. (2012) notaram que, dado o alcance do sensor do robô, é interessante notar que

algumas fronteiras que foram detectadas separadamente, podem acabar sendo exploradas conjuntamente quando o robô navega na direção delas. Por esse motivo, pode ser interessante a clusterização dessas fronteiras.

Keidar and Kaminka (2012) acreditam que os métodos tradicionais de detecção de fronteiras, baseados em algoritmos de visão computacional, são um gargalo para o sistema se o ambiente a ser explorado é muito grande. Para resolver esse problema, eles propuseram duas novas abordagens, uma que trata os pontos do mapa como um grande grafo conectado e um algoritmo de busca em largura é usado para encontrar a fronteira mais próxima, e outra que analisa apenas as novas leituras dos sensores.

A exploração de fronteiras também pode ser aplicada para o caso de múltiplos robôs. Kasperski et al. (2013) propuseram uma nova abordagem onde todos os robôs executam um algoritmo de *wavefront* simultaneamente, delimitando regiões no mapa. Dessa forma, um robô só poderá visitar uma fronteira que esteja dentro da sua região. Bautin et al. (2012) aplica uma nova abordagem aonde o *wavefront* é iniciado de cada fronteira, ao invés dos robôs. Já para Burgard et al. (2005), todos os robôs possuem uma função de utilidade baseada na distância do mesmo e na proximidade de outros robôs àquela fronteira.

3 Information Window (IW)

Como dito na Seção 1, o objetivo deste trabalho é desenvolver uma nova abordagem para a detecção e exploração de fronteiras baseado na informação do mapa. Essa abordagem foi chamada de *Information Window*. Para atingir este objetivo, o IW executa uma análise do mapa explorado a partir de uma janela que desliza sobre o mapa obtendo informações sobre fronteiras e incertezas do mapa. Estas informações são utilizadas por uma função de utilidade que destaca regiões interessantes de serem exploradas.

O objetivo do IW é aumentar o ganho de informação após o término de cada planejamento. Ao analisar o mapa retornado pelo algoritmo de SLAM, o robô é capaz de perceber regiões do mapa onde as fronteiras se encontram e regiões onde existe incerteza sobre o mapeamento. Dessa forma, pode-se afirmar que o robô possui dois objetivos distintos quanto ao ganho de informação:

- Exploração: ocorre quando o robô navega para regiões desconhecidas do ambiente, aumentando assim o tamanho do mapa e, por consequência, a quantidade de informação do sistema;
- Qualidade da informação: ocorre quando uma região já foi visitada, mas existem incertezas no mapa de ocupação em pontos da-

quela região. Neste caso nenhuma nova informação é adicionada ao sistema, mas aquelas que já existiam são melhoradas.

Dessa forma, o IW considera que fronteiras que não foram bem mapeadas, ou seja, possuem grandes incertezas, são as melhores regiões para se explorar. Isto se deve ao fato de que tais fronteiras, ao serem exploradas, além de acrescentarem informações ao sistema, ainda reforçam informações existentes.

3.1 Definições

Antes de explicar maiores detalhes do IW, é preciso formalizar alguns conceitos e redefinir outros referentes ao SLAM e à exploração de fronteiras. É prática comum nos algoritmos de SLAM, determinar um *threshold* que delimita quando uma célula é livre ou ocupada. Dessa forma é possível definir um valor binário, 0 ou 1, que diferencie as duas células no grid de ocupação. Para a nossa abordagem esse valor de *threshold* não será utilizado, fazendo com que o grid de ocupação possua valores que variam de 0 a 1. Assim, o IW possui dois mapas diferentes:

- **Mapa real:** que é o grid de ocupação tradicional, com um *threshold* definido. Serve para poder gerar a visualização do mapa estimado.
- **Mapa de ocupação:** que é o grid de ocupação sem o *threshold*. Serve para poder analisar as incertezas do mapa estimado.

O **mapa real** é utilizado apenas por questões de visualização, já que todo o processamento do IW é feito sobre o **mapa de ocupação**. No IW, o algoritmo de SLAM utilizado foi o GridSLAM (Hähnel et al., 2003). O GridSLAM retorna um mapa de ocupação onde valores próximos de 0 representam células livres, valores próximos de 1 representam células ocupadas e valores iguais a -1 representam células desconhecidas.

Com o uso do mapa de ocupação, um novo tipo de célula foi definido, a **célula incerta**. Esta é uma célula que não recebeu um valor 0 ou 1, mas que está entre estes dois valores, portanto não pode ser considerada nem livre, nem ocupada.

3.2 Algoritmo

O algoritmo do IW pode ser explicado da seguinte forma. Inicialmente, um algoritmo de SLAM, no caso deste trabalho, o GridSLAM, entrega para o IW o mapa de ocupação obtido até o momento. Este mapa de ocupação pode ser representado por uma matriz $mapa(width, height)$, onde *width* é a largura máxima do mapa e *height* a altura máxima do mapa. Esta matriz é então percorrida por uma janela deslizante de tamanho $n \times n$, sendo *n* ímpar. Para cada célula do mapa de ocupação em

que a célula central da janela fica sobreposta, um valor de informação, Fu , é calculado gerando um novo mapa chamado de **mapa de informação**. A Figura 2 ilustra uma janela, em vermelho, com *n* = 5, sobre um mapa de ocupação. Na figura é possível ver os valores do mapa de ocupação, e a célula central da janela, onde ficará o valor de Fu no mapa de informação.

-1	-1	-1	-1	-1	0,5	0,1	0
-1	-1	-1	0,5	0,5	0,41	0	0
-1	-1	0,5	0,35	0,23	0	0	0
-1	-1	0,5	0,25	Fu	0	0	0
-1	-1	0,4	0,2	0,1	0	0	0
-1	-1	-1	1	1	0,1	0	0
-1	-1	-1	-1	1	0,9	0	0

Figura 2: Exemplo de um mapa de ocupação com uma janela deslizante, em vermelho. A célula verde representa o valor de informação daquela janela.

Como os valores de Fu são normalizados ao final do processo, pode-se dizer que eles representam o quanto melhor é explorar uma célula do mapa em relação às outras. Em outras palavras, a quantidade de informação que será acrescentada ao sistema, ao visitar uma determinada célula, em relação às outras. O cálculo de Fu leva em consideração todas as células da janela. Assim, o valor de Fu é resultado de uma análise da região que compõe a janela. Pode-se escrever a equação de Fu como:

$$Fu_{(i,j)} = \sum_{x \in n} \sum_{y \in n} f(x, y) \quad (1)$$

onde, *i* e *j* são os índices da célula do mapa de ocupação, *n* é o tamanho da janela, e $f(x, y)$ é uma função que determina o valor daquela célula para a janela.

O cálculo de $f(k)$ consiste em comparar a célula *k* com todos os seus vizinhos, individualmente, e definir valores para cada interação. A Figura 3 ilustra o cálculo de $f(x, y)$. O valor final de Fu é então a soma de todos os valores de $f(k)$ presentes na janela.

Ao se comparar uma célula com seus vizinhos, podem ocorrer 7 interações:**células iguais**, **célula ocupada com célula livre**, **célula desconhecida com célula livre**, **célula desconhecida com célula ocupada**, **célula desconhecida com célula incerta**, **célula ocupada com célula incerta** e **célula livre com célula incerta**. Quando duas células são iguais, a intera-

-1	-1	-1	-1	-1	0,5	0,1	0
-1	-1	-1	0,5	0,5	0,41	0	0
-1	-1	0,5	0,35	0,23	0	0	0
-1	-1	0,5	0,25	0	0	0	0
-1	-1	0,4	f	0,1	0	0	0
-1	-1	-1	1	1	0,1	0	0
-1	-1	-1	-1	1	0,9	0	0

Figura 3: O cálculo de f da célula em azul. As setas verdes representam as interações dos vizinhos com a célula sendo calculada.

ção entre as duas não fornece informação nenhuma ao sistema, do mesmo modo que uma célula livre e uma ocupada, pois estas já não podem ter seus valores reforçados na exploração.

O encontro de uma célula ocupada com uma desconhecida ou incerta pode possuir alguma informação para o sistema, mas, pensando na tarefa de exploração, pode ser perigoso considerar estas regiões. Quando, por exemplo, uma parede no limite externo do ambiente é encontrada, toda a sua extensão do lado de fora do ambiente possui a interação entre células ocupadas e células desconhecida, mas estas regiões não fazem parte do espaço alcançável do robô. Por este motivo, todas as janelas que possuem estas interações são consideradas nulas e não são avaliadas pelo IW.

As três interações que restaram são as mais importantes para o IW. Quando uma célula livre e uma célula desconhecida são vizinhas temos uma fronteira como descrita por Yamauchi (1997). Quando uma célula desconhecida e uma célula incerta ou uma célula livre e uma célula incerta são vizinhas ocorre uma nova interação, pois não é possível saber se ali é uma célula ocupada ou livre. Assim, o IW valoriza mais explorar as duas últimas interações do que a primeira. O valor de $f(k)$ pode então ser calculado pela Equação 2:

$$f(k) = \sum_{i=x-1, j=y-1}^{i=x+1, j=y+1} -mapa(k) + fn(i, j) \quad (2)$$

onde k é a coordenada (x, y) da célula sendo analisada no momento, $mapa(k)$ é o valor contido na célula k da janela e $fn(i, j)$ é uma função com forma normal que, dado o valor do mapa de ocupação na célula (i, j) , retorna um valor variando de 0 a 1, conforme a figura 4. A equação 3 descreve a função fn . A idéia por trás da função fn consiste no fato de que uma célula que possua um valor dde 0,5 é uma célula que já foi mapeada,

mas que nenhuma informação foi obtida sobre a mesma, então ela deve ter prioridade para ser explorada em comparação com as outras. Assim, toda célula com valor 0,5 recebe o valor máximo da função fn , este valor é reduzido gradualmente conforme o valor da célula sendo analisada se aproxima de 0 ou 1, valores onde a célula deixa de ser considerada incerta e passa a ser considerada livre ou ocupada.

$$fn(x, y) = e^{-\frac{(mapa(k)-0,5)^2}{2\sigma^2}} \quad (3)$$

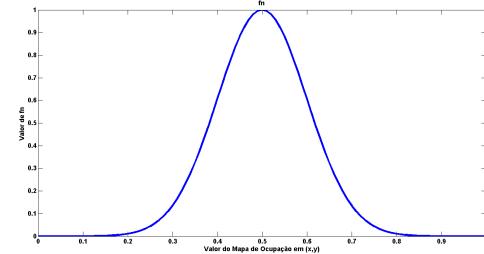


Figura 4: Gráfico demonstrando os possíveis valores da função fn .

Todo esse processo do IW pode ser explicado pelo algoritmo a seguir. O algoritmo precisa apenas do mapa de ocupação e de um valor n para o tamanho da janela. As duas primeiras linhas fazem o loop para janela deslizar por todas as células do mapa de ocupação. As linhas 4 e 5 definem o loop para percorrer todas as células da janela, enquanto que as linhas 7 e 8 fazem o loop de uma célula da janela com seus vizinhos. A linha 9 calcula a interação entre as células vizinhas, enquanto que a linha 12, calcula o valor total da janela. Por fim, na linha 15 o valor de Fu é colocado no mapa de informação.

Require: *mapa_ocupacao*

Require: *n*

```

1: for  $i \leftarrow 1$  to mapa_ocupacao.size() do
2:   for  $j \leftarrow 1$  to mapa_ocupacao.size() do
3:      $Fu = 0;$ 
4:     for  $k \leftarrow i - n/2$  to  $i + n/2$  do
5:       for  $l \leftarrow j - n/2$  to  $j + n/2$  do
6:          $f = 0;$ 
7:         for  $x \leftarrow k - 1$  to  $k + 1$  do
8:           for  $y \leftarrow l - 1$  to  $l + 1$  do
9:              $f = f + (-mapa(k, l)) +$ 
                $fn(x, y));$ 
10:            end for
11:          end for
12:           $Fu = Fu + f;$ 
13:        end for
14:      end for
15:      mapa_informacao( $i, j$ ) =  $Fu$ 
16:    end for
17:  end for

```

Na Figura 5 um mapa de informação é ilustrado em conjunto com o mapa real que o gerou.

Como ilustrado na Figura 5b, o mapa de informação é um grid de ocupação onde os valores de cada célula podem variar de 0 (células pretas) a 1 (células brancas). É possível notar que regiões com células brancas são regiões de fronteira. Assim, células em tons mais próximos de branco são aquelas que possuem maior probabilidade de acrescentarem mais informação ao sistema, então basta ao robô navegar para a região que possui o maior valor no mapa de informação.

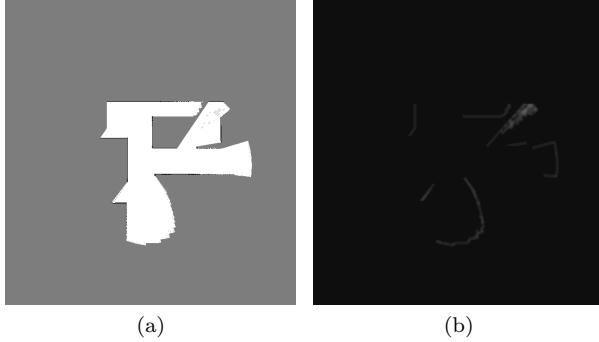


Figura 5: Em (a) o mapa real e em (b) o mapa de informação relativo.

4 Experimentos e Resultados

Para avaliar o desempenho do IW, ele foi comparado com o *near-frontier exploration* (NFE) (Yamauchi, 1997) com a técnica de clusterização de fronteiras proposto por (Keidar and Kaminka, 2012). Para comparar os dois algoritmos, foram definidas 3 métricas:

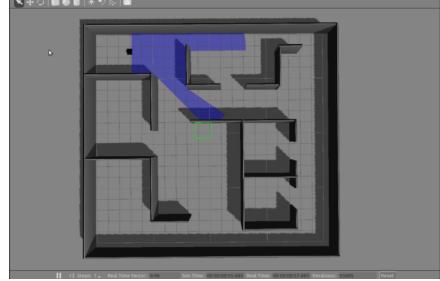
- **Tempo de exploração**
- **Qualidade do mapa**
- **Informação x Planejamento**

Para os realizar os testes, foi utilizado o ambiente de desenvolvimento robótico ROS (Quigley et al., 2009) em conjunto com o ambiente de simulação Gazebo (Koenig and Howard, 2004), simulando um Pioneer3AT com um sensor laser Hokuyo em um ambiente de $20,5 \times 19,7m$. A Figura 6 apresenta o ambiente de simulação utilizado. Na Figura 6a temos o Gazebo com o ambiente utilizado nos testes e na Figura 6b temos o ambiente de simulação e visualização do ROS. Todos os testes foram executados em um computador Pentium I5, com 6 Gb de memória, rodando Ubuntu 12.04.

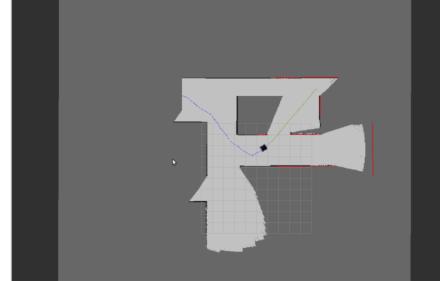
Para comparar o IW e o NFE, foram executados 10 testes com cada método, todos com condições iniciais iguais.

4.1 Tempo de Exploração

O tempo de exploração dos dois métodos foi analisando verificando o clock interno do ROS no ini-



(a)



(b)

Figura 6: Em (a) o Gazebo com o ambiente utilizado e em (b) um teste sendo simulado no ROS.

cio da exploração e no fim. O fim da exploração para o IW acontece quando o mapa de informação não possui mais regiões de interesse, enquanto que para o NFE acontece quando o algoritmo não encontrar mais fronteiras. A Figura 7 apresenta o gráfico com o tempo médio das 10 execuções para cada método.

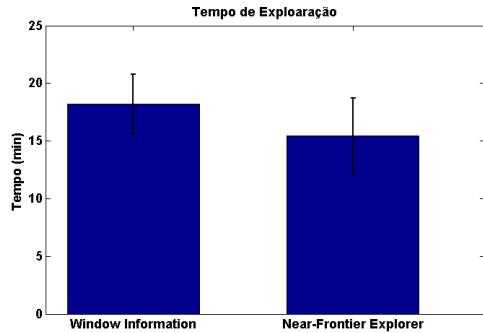


Figura 7: Média do tempo de exploração para os dois métodos.

Como esperado, o NFE possui um tempo médio de exploração menor que IW, pois o NFE prioriza a velocidade de exploração.

4.2 Qualidade do Mapa

Para avaliar a qualidade do mapa, foi utilizada a métrica descrita por Juliá et al. (2012). Nesta métrica, o mapa estimado pelo algoritmo de SLAM é sobreposto sobre um mapa perfeito, sem erros, do ambiente para calcular um valor de qualidade. Esse valor é o número de células livres/ocupadas

no mapa perfeito que possuem um valor diferente no mapa estimado mais o número de células livres/ocupadas no mapa estimado que possuem um valor diferente no mapa perfeito. Este valor é então normalizado pelo número total de células do mapa. Assim, este valor pode ser calculado subtraindo um mapa de outro e contando o número de células diferentes obtidas. Para a métrica funcionar, é preciso que os mapas estejam alinhados. A Figura 8 mostra o mapa perfeito do ambiente (a), juntamente com um mapa com um valor de qualidade bom (b) e um mapa com valor de qualidade ruim (c).

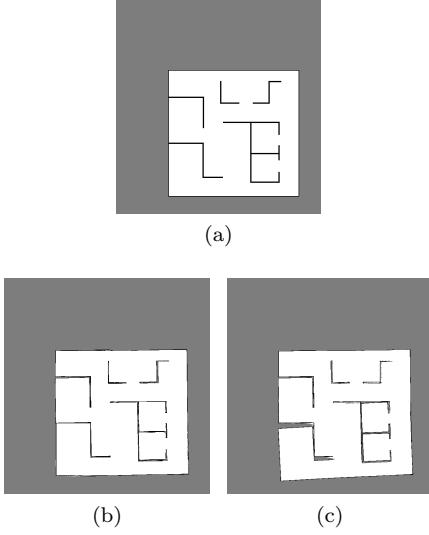


Figura 8: Em (a) o mapa perfeito. Em (b) um exemplo de mapa bom e em (c) um exemplo de mapa ruim.

A Figura 9 apresenta o gráfico obtido para a qualidade do mapa. É possível notar no gráfico que não existe uma diferença significativa entre os dois métodos na métrica de qualidade do mapa. Isto pode ser explicado pelo fato de que o algoritmo de SLAM utilizado, o GridSLAM, é um algoritmo muito eficiente e possui um bom *loop closure*, ou seja, o GridSLAM consegue corrigir a maior parte dos erros que ocorrem durante a navegação do robô.

4.3 Informação x Planejamento

A métrica de Informação x Planejamento procura avaliar a quantidade de informação adicionada ao sistema para cada vez que o robô planeja explorar uma fronteira. Os dois algoritmos foram implementados de forma a somente re-planejar um novo objetivo quando chegar no objetivo anterior. Assim, toda informação adicionada ao sistema durante a exploração de uma determinada fronteira é avaliada. Dessa forma, esta métrica pode ser calculada verificando-se a porcentagem do mapa explorado ao fim de um planejamento. A Figura 10 apresenta o gráfico com a porcentagem média

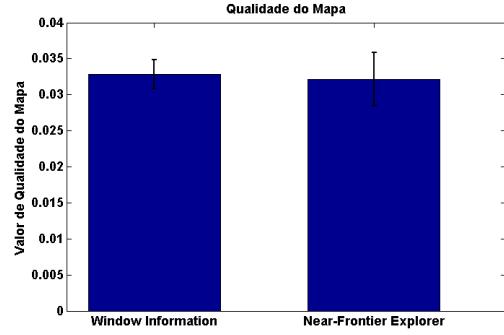


Figura 9: Média da qualidade do mapa para os dois métodos.

de exploração do mapa pelo número de planejamentos.

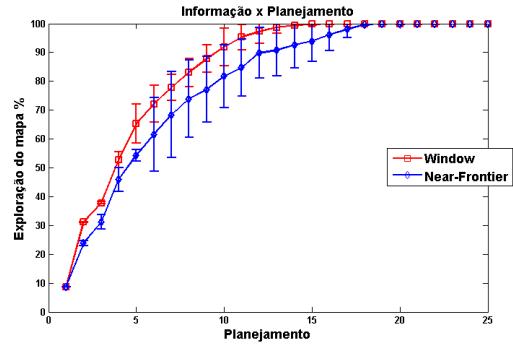


Figura 10: Porcentagem explorada do mapa pelo número de planejamentos.

Observando a Figura 10 pode-se concluir duas coisas. A primeira diz respeito à velocidade de exploração do ambiente. Explorando, aproximadamente, 14 fronteiras, mais de 99% do ambiente é mapeado com o IW, enquanto que o NFE necessita explorar, aproximadamente, 18 fronteiras. Este é um fato que contradiz a primeira métrica de tempo de exploração.

Para explicar este fato, é preciso entender o funcionamento prático dos dois métodos. O IW, ao escolher uma nova fronteira, não leva em consideração a distância da mesma. Assim, o IW pode navegar pelo ambiente, explorando fronteiras distantes, mas mais interessantes. Por este motivo, fronteiras próximas ao robô, mas que não são interessantes ao IW podem ficar para trás, fazendo com que o robô, ao chegar em 99% de exploração, tenha que voltar para mapeá-las posteriormente. Já o NFE, sempre explora a fronteira mais próxima. Dessa forma, ao chegar em 99% do ambiente explorado, não existem fronteiras por explorar longe do robô, fazendo com que o mapeamento termine mais rápido.

Outra observação a ser feita sobre a Figura 10 tem relação com a quantidade de informação obtida por cada planejamento. É possível notar no gráfico que para cada planejamento, o IW adiciona

mais informação ao sistema que o NFE, sendo esta a proposta do método.

5 Conclusões

Uma nova abordagem de exploração de fronteiras foi apresentada neste trabalho. Ao avaliar não o mapa estimado do ambiente, mas o grid de ocupação estimado pelo algoritmo de SLAM, a nova abordagem procura, ao escolher uma fronteira, aumentar a quantidade de informação do sistema.

Utilizando uma janela deslizante sobre o grid de ocupação, foi proposto um método de avaliação do grid de ocupação, através de uma função de utilidade que leva em conta todas as células da janela. Este método possui a vantagem de analisar as interações entre as células do grid e não apenas cada célula individualmente.

Ao comparar a abordagem proposta com uma abordagem muito utilizada na literatura, o NFE, melhorado com o uso da técnica de clusterização de fronteiras, foi mostrado que o IW, para explorar quase todo o ambiente, precisa visitar menos fronteiras que o NFE, sem interferir na qualidade do mapa apresentado.

Mesmo precisando explorar menos fronteiras, o IW utiliza mais tempo de exploração que o NFE. Isto ocorre porque o NFE é mais objetivo na escolha de fronteiras para explorar, enquanto que o IW navega mais pelo ambiente, podendo percorrer grandes percursos sempre que escolhe uma nova fronteira para explorar. Este problema pode ser resolvido adicionando-se à abordagem uma análise de distância das fronteiras.

Como trabalho futuros, pretende-se aplicar à função de utilidade um novo termo que relaciona a distância do robô para cada fronteira, para que o tempo de exploração do IW possa ser melhorado. Pretende-se também, aplicar o IW de forma a ordenar múltiplos robôs na tarefa de mapeamento cooperativo.

Agradecimentos

Os autores gostariam de agradecer o apoio da Fapemig, CNPq e CAPES no desenvolvimento desse trabalho.

Referências

- Bautin, A., Simonin, O. and Charpillet, F. (2012). Minpos : A novel frontier allocation algorithm for multi-robot exploration, *Intelligent Robotics and Applications*, Vol. 7507 of *Lecture Notes in Computer Science*, pp. 496–508.
- Burgard, W., Moors, M., Stachniss, C. and Schneider, F. (2005). Coordinated multi-robot exploration, *Robotics, IEEE Transactions on* **21**(3): 376–386.
- Faigl, J., Kulich, M. and Preucil, L. (2012). Goal assignment using distance cost in multi-robot exploration, *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3741–3746.
- Hähnel, D., Burgard, W., Fox, D. and Thrun, S. (2003). An efficient fastslam algorithm for generating maps of large-scale cyclic maps of large-scale environments from raw laser range measurements, *IN PROC. OF THE IEEE/RSJ INT. CONF. ON INTELLIGENT ROBOTS AND SYSTEMS (IROS)*, pp. 206–211.
- Holz, D., Basilico, N., Amigoni, F. and Behnke, S. (2010). Evaluating the efficiency of frontier-based exploration strategies., *ISR/ROBOTIK*, pp. 1–8.
- Juliá, M., Gil, A. and Reinoso, O. (2012). A comparison of path planning strategies for autonomous exploration and mapping of unknown environments, *Autonomous Robots* **33**(4): 427–444.
- Kasperski, S., Eich, M. and Wollenberg, J. (2013). Evaluation of cooperative exploration strategies using full system simulation, *ICAR '13*.
- Keidar, M. and Kaminka, G. A. (2012). Robot exploration with fast frontier detection: Theory and experiments, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pp. 113–120.
- Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator, *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2149–2154.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A. Y. (2009). Ros: an open-source robot operating system, *ICRA workshop on open source software* **3**(3.2).
- Stachniss, C., Hähnel, D. and Burgard, W. (2004). Exploration with active loop-closing for fastslam, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 2, pp. 1505 – 1510.
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration, *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pp. 146–151.