An Approach for Environment Mapping and Control of Wall Follower Cellbot Through Monocular Vision and Fuzzy System

Karoline de M. Farias, Wilson Leal R. Junior, Ranulfo P. Bezerra Neto,
Ricardo A. L. Rabelo and Andre M. Santana
Departmanto de Computação
Universidade Federal do Piauí
Brasil, Teresina 64049–550
Emails: kdemourafarias@gmail.com, wilsonlealjunior@hotmail.com, ranulfo0s@gmail.com,
ricardoalr@ufpi.edu.br, andremacedo@ufpi.edu.br

***Abstract*—This paper presents an approach using range measurement through homography calculation to build 2D visual occupancy grid and control the robot through monocular vision. This approach is designed for a Cellbot architecture. The robot is equipped with wall following behavior to explore the environment, which enables the robot to trail objects contours, residing in the fuzzy control the responsibility to provide commands for the correct execution of the robot movements while facing the adversities in the environment. In this approach the Cellbot camera works as a sensor capable of correlating the images elements to the real world, thus the system is capable of finding the distances of the obstacles and that information is used for the occupancy grid mapping and for fuzzy control input. Experimental results with V-REP simulator are presented to validate the proposal, and the results were favorable to the use in robotics and in acceptable computing time.**

***Index Terms*—Occupancy-Grid Mapping, Wall-Following, Fuzzy System, Monocular Vision.**

I. Introdução

Robótica móvel é um importante tópico de pesquisa por muitas razões. Entre elas, o surgimento de propostas de soluções para diversas tarefas do dia-a-dia, sejam elas domésticas (aspiradores de pó e cortadores de grama), industriais (Transporte automatizado e veículos de carga autônoma), militares (sistemas de monitoramento aéreo remoto - VANTs, transporte de suprimentos e de armazenamento de zonas de guerra, sistemas táticos de guerra) e segurança pública (controle e patrulhamento de ambientes, resgates e exploração em ambientes hostis).

Um dos grandes desafios enfrentados pela robótica é a autonomia. Um sistema robótico é considerado autônomo se é capaz de reconhecer o ambiente na qual está inserido, inferir informações a partir dele, mover-se independentemente, adaptar-se às suas condições, além de ter habilidades de planejamento e raciocínio. Outro grande desafio da robótica é o custo de aquisição tanto dos robôs quanto de seus sensores. Atualmente estão sendo propostas soluções mais baratas de construção de robôs, de forma a popularizar o estudo da robótica. Uma dessas soluções é o *cellbot* [1], que basicamente consiste em um robô móvel que se utiliza de um dispositivo móvel como ambiente de execução ou fonte de sensoriamento. A grande vantagem do uso dos *cellbots* é a popularização dos *smartphones* na sociedade, a gama de sensores que disponibilizam e o poder de processamento comparável a um computador pessoal.

Para que um robô possa navegar de formar autônoma pelo ambiente, é necessário que ele seja capaz de construir um mapa usando seus sensores. O Mapeamento de Ambientes consiste em um conjunto de dados coletados através de sensores com o objetivo de gerar, a partir de características do ambiente, um modelo computacional levando em consideração o método de aquisição, o tipo de armazenamento e a estrutura na qual os dados serão armazenados. Dentro desse conceito, os sensores são determinantes na formulação do problema.

As primeiras implementações de mapeamento utilizavam sensores de alcance, que fornecem informação de distância e orientação entre o robô e o obstáculo. Com o tempo outras formas de aquisição foram sendo exploradas, um exemplo são os sensores visuais. No campo da Robótica, esse tipo de sensor vem se tornando comum, pois, segundo [2], possuem a vantagem de serem mais baratos e permitirem trabalhar com uma gama maior de informações quando comparados com sensores de alcance. Além do mapeamento outro fator importante é a navegação de ambientes, uma técnica utilizada para este fim é o comportamento de seguidor de parede. Esse comportamento habilita o robô seguir contornos de objetos, como por exemplo, paredes e obstáculos em um ambiente e também pode ser combinado com outros comportamentos inteligentes para alcançar tarefas de alta complexidade [3] .

Um problema relevante, e que deve ser levado em consideração em navegação autônoma, é lidar com uma grande quantidade de incertezas [4]. O problema se torna mais complexo quando, além da tarefa da navegação, o robô deve atuar no ambiente lidando com incertezas inerentes a leitura do sensor, sendo que muitas dessas leituras influenciam o desempenho

978-1-5386-3057-0/17/$31.00 ©2017 IEEE

do sistema de navegação autônoma.

Para superar o problema das leituras incertas realizadas pelos sensores, soluções baseadas em Sistemas de Inferência *fuzzy* têm sido apresentadas como uma abordagem robusta, flexível e capaz de lidar com navegação em tempo real. Isso por que a lógica *fuzzy* ao contrário da lógica clássica é tolerante a imprecisões, incertezas e verdades parciais [5] [6] [7].

Este artigo apresenta uma abordagem de mapeamento de ambientes em grade de ocupação e um sistema de controle baseado na logica *fuzzy*(CLF), para a navegação de um robô seguidor de parede em um ambiente estático. Para o mapeamento e controle do robô é utilizado um sistema visual monocular baseado em homografia, que fornece a distância do robô em relação aos obstáculos que se encontram no ambiente. A distância fornecida pelo sistema visual monocular é usada como entrada tanto para o controlador *fuzzy* como também para o algoritmo mapeamento.

As contribuições deste trabalho são o uso de um único sensor, a câmera do *cellbot*, para realização do mapeamento e navegação do ambiente, proporcionando assim uma diminuição no custo computacional quando comparado com sistemas que utilizam sensores diferentes para o mapeamento do ambiente e controle do robô. Além disso, outra grande contribuição é utilização de uma proposta robótica, o *cellbot*, capaz de processar tarefas robóticas com eficiência comparável a um computador pessoal a um custo monetário menor e com potencial de realizar tarefas ainda mais complexas.

II. Fundamentação Teórica

A teoria de conjuntos *fuzzy* foi concebida por L.A. Zadeh com o objetivo de fornecer um ferramental matemático para o tratamento de informações de caráter impreciso ou vago [8]. Foi desenvolvida e inspirada na capacidade humana de tomar decisões diante de processos indecisos e ambíguos. Dentre as principais características dos controladores baseados na lógica *fuzzy*, está sua habilidade de tomar decisões adequadas para um processo de inferência linguística [9]. Os processos de inferência linguística são baseados em regras do formato seentão baseado em experiencia, heurística e conhecimento fornecido por um especialista ou extraído por dados numéricos.

Sistemas baseados em lógica *fuzzy* para a navegação autônoma de robôs móveis têm sido desenvolvidos por muitos pesquisadores, que utilizam diferentes tipos de sensores para mensurar as distancias dos obstáculos existentes no ambiente. [3], [5] e [10] propõem controladores, dentre eles, o controlador baseado em lógica *fuzzy* para um robô seguidor de parede que usa rodas com o acionamento diferencial. Em [3] e [5] são usados sensores ultrassônicos, onde em [3] o controlador foi testado em ambiente real, validando assim a proposta. Já em [5] os controladores foram testados em ambiente simulado e todos obtiveram resultados satisfatórios, resultando em uma análise crítica dos controladores usados. [10] propõem um sistema de navegação visual baseado na lógica *fuzzy*, utilizando como sensor uma câmera para detectar obstáculos na frente, na esquerda e na direita do robô, os resultados dos experimentos mostram que o controlador tem um bom desempenho na navegação robótica.

As informações obtidas pelos sensores são de extrema importância para o mapeamento de ambientes, dentre os tipos mais comuns de mapeamento está a grade de ocupação. Esse tipo de mapeamento é baseado na discretização dos espaços contínuos do ambiente de maneira que este possa ser representado em uma matriz multi-dimensional. Nessa matriz, cada elemento é conhecido como célula onde é guardado informações probabilísticas de ocupação.

A formulação clássica para este modelo é representada pela Equação (1) [11] [12]. Esse cálculo pressupõe que a pose do robô e as medições do sensor são conhecidas, sendo $m_{x,y}$ uma célula do mapa, $z_{1:t}$, o conjunto de medidas do sensor até o instante t, e $x_{1:t}$, o caminho do robô definido através de todas as suas poses durante o processo de mapeamento.

$$P(m_{x,y}|z_{1:t}, x_{1:t}) \tag{1}$$

Para todas as células que caem dentro do alcance dos sensores são atualizadas as suas probabilidades através da modelagem inversa das medidas dos sensores representada pela Equação (2):

$$P(z_t|z_{1:t-1}, m_{x,y}) \tag{2}$$

Com isso o valor de probabilidade pode ser estimado a qualquer instante t por meio da regra de Bayes aplicada a Equação (1), como demonstrado pela Equação (3):

$$P(m_{x,y}|z_{1:t}, x_{1:t}) = \frac{P(z_t|z_{1:t-1}, m)P(m_{x,y}|z_{1:t-1}, x_{1:t})}{P(z_t|z_{1:t}, x_{1:t})} \tag{3}$$

$P(m_{x,y}|z_{1:t-1}, x_{1:t})$ representa a probabilidade de ocupação da célula até o instante $t-1$ e $P(z_t|z_{1:t}, x_{1:t})$ o valor real medido pelo sensor.

Alguns trabalhos recentes como [13] [14], propõem uma nova modelagem de sensor que melhor se adapta as novas tecnologias de sensores de alcance existentes, bem como oferece um conjunto de dados maior para o mapeamento. Outras abordagens [15] [16] [17] visam o SLAM(*Simultaneous Localization and Mapping*) e utilizam a modelagem clássica de sensores de alcance para laser 2D.

No contexto de visão monocular como sensor para mapeamento, o trabalho de [18] utiliza visão monocular para mapeamento 2D aplicado a robôs móveis. Nesta abordagem é feita a segmentação da imagem para classificação em células livres e ocupadas, onde o mapeamento dos obstáculos é feito com valores do plano real obtidos pelo cálculo da matriz homografia. Em [2], é apresentado uma proposta de mapeamento 2,5D em grade de ocupação e elevação utilizando visão estéreo para obtenção de informações, onde cada célula no mapa armazena uma probabilidade de ocupação, a altura do espaço mapeado e a variância do valor da altura. O trabalho de [19], baseia-se em mapeamento utilizando visão estéreo com objetivo de encontrar cantos, saídas e obstáculos de maneira a criar uma navegação local segura e planejamento de caminho para robôs com rodas. Em [20] é usado visão monocular para

encontrar ponto de interesse nas imagens e estimar a pose da câmera. Os pontos encontrados são reconstruídos a partir de duas imagens sequenciais para realização do mapeamento e localização 3D. [21] traz uma abordagem de visão monocular onde informações de distância são obtidas das imagens e é feita uma relação entre essas medidas de maneira que se assemelhem a sensores de alcance. Essas informações servem de entrada para um esquema de mapeamento em grade de ocupação com o objetivo de obter atualizações mais precisas para o mapa criado.

O grande diferencial da abordagem proposta neste trabalho com as demais exemplificadas, é o fato de ser voltado para uso em *cellbot*. O sistema de sensoriamento é provido pelo *smartphone* acoplado ao *cellbot*, sendo a câmera o sensor utilizado. As imagens capturadas pela câmera sofrem um processo de extração de informações de distâncias de maneira semelhante a sensores de alcance. Essas informações servem de entrada para um sistema de controle e mapeamento. A vantagem desse modelo é o uso de apenas um sensor para realização de dois processos independentes, o que gera uma economia no processamento total.

III. Abordagem

Este trabalho propõe uma abordagem que transforma informação visual recebida por uma câmera monocular em informação de distância. Essa informação é usada como entrada em um sistema de controle *fuzzy* para um robô seguidor de parede e para o algoritmo de mapeamento. Para isso, a imagem provinda da câmera sofre projeção para o plano real com dimensão 2D por meio da matriz de homografia. Depois disso, a imagem sofre uma segmentação que separa o que é ou não obstáculo. Logo em seguida, as distâncias entre o robô e as células oclusas são extraídas e usadas como entrada para o CLF.

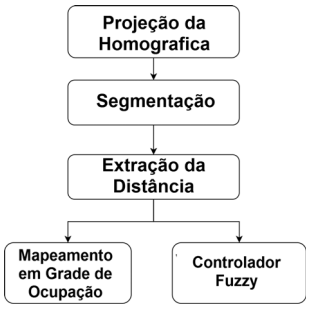


Figura 1. Passos da abordagem.

Esta seção está dividida em 5 etapas: Projeção da Homografia, Segmentação, Extração da Distância, Mapeamento em Grade de Ocupação e Controlador *Fuzzy*. Figura 1 ilustra as etapas da abordagem.

A. Projeção da Homografia

Inicialmente, a localização do robô e as imagens do ambiente são extraída do simulador V-REP [22]. Depois disso, as imagens recebidas do sensor visual são projetadas em

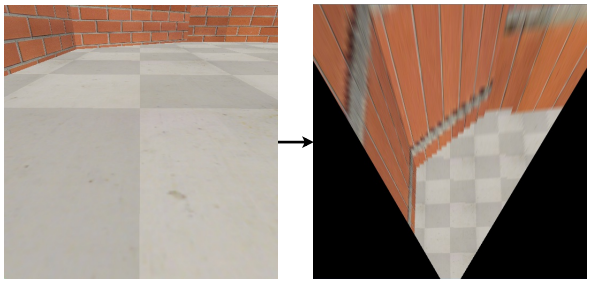


Figura 2. Exemplo de projeção de homografia. Imagem à esquerda representa a captura feita pela câmera e a da direita, o resultado da projeção.

um plano 2D utilizando a matriz de homografia como pode ser visto na Figura 2. Este procedimento é realizado de tal forma que os *pixels* da imagem sejam representados como medidas do mundo real. A projeção da imagem pode então ser computada como uma aproximação da distância do robô ao obstáculo. Esse processo pode ser obtido através da seguinte formulação matemática:

$$s_i \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4}$$

Como o sistema desenvolvido é voltado para o plano 2D, os pontos analisadas são representados em coordenadas homogêneas. Pela Equação (4), s_i representa um fator de escala, x_i', y_i' representam os pontos no plano real obtidos pela multiplicação de uma matriz Homografia representada por H (matriz 3X3 com informações de projeção imagem-mundo) pelos pontos x_i, y_i que representam as coordenadas dos pontos no plano da imagem.

B. Segmentação

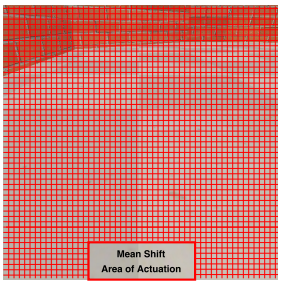


Figura 3. Área de atuação do *Mean shift*.

Na etapa de segmentação, a imagem é dividida em células com um tamanho predeterminado, como pode ser visto na Figura 3. As células são classificadas utilizando o componente de cor RGB da imagem, assim como em [18]. Por meio desse processamento cada célula é classificada em oclusa ou livre. Portanto, é utilizada a área mais próxima do robô como padrão de espaço vazio. A Figura 3 mostra a área padrão predeterminada.

Uma vez computada a média e variância dos componentes R, G e B da área padrão utilizando o algoritmo *Mean Shift*,

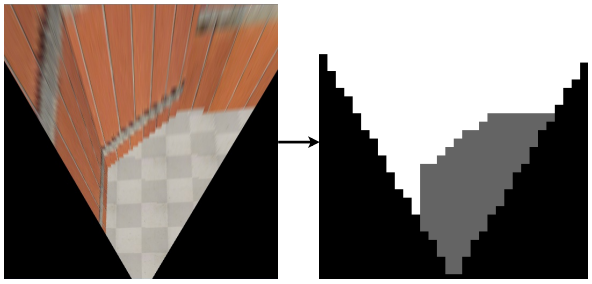


Figura 4. A imagem da direita é resultado do processo de segmentação da imagem da esquerda, onde a área branca representa uma área de oclusão, a área cinza representa espaço livre e a área preta representa área fora do campo de visão do robô.

estes valores são utilizados para comparar cada célula, afim de determinar quais células são caracterizadas como oclusas. A Figura 4 mostra o resultado da segmentação da imagem projetada em 2D.

C. Extração da distância

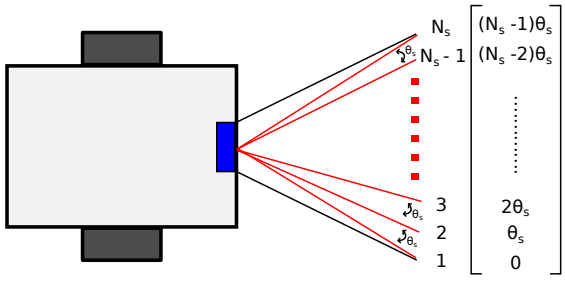


Figura 5. Ilustração do vetor de ângulos.

Esta fase ocorre durante a segmentação. Inicialmente, definimos quantos sensores serão usados no experimento. Em seguida, criamos um vetor de ângulos correspondente a cada sensor, como é ilustrador na figura 5. Durante a fase de segmentação, cada bloco rotulado como ocupado é mapeado em um dos ângulos. Como a segmentação ocorre de cima para baixo, os últimos blocos marcados corresponderão aos obstáculos mais próximos associados a cada ângulo como pode ser visto na Figura 6, onde a distância de cada bloco ao centro de visão é mapeado no vetor de ângulos.

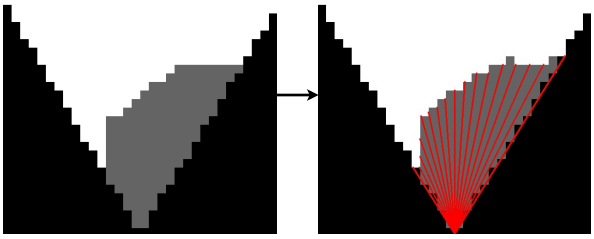


Figura 6. A imagem da direita representa o resultado da interpretação de sensores de alcance, onde de $N_s = 15$, da imagem segmentada da esquerda.

D. Mapeamento em Grade de Ocupação

Depois de obter a lista de ângulos e as distâncias relacionadas durante a extração do sensor, é dado início ao mapeamento. Como as informações da imagem foram convertidas de maneira a se comportar como sensores de alcance, a formulação probabilística clássica proposta por [11] é utilizada sem grandes modificações. O que é uma vantagem dessa abordagem, considerando a sua adaptabilidade a um modelo já vigente e de fácil implementação. Assim, a modelagem do sensor dos sensores de alcance podem ser dadas por uma função Gaussiana, como demonstrado pela Equação (2):

$$P(r|d_{x,y}) = \frac{1}{\sqrt{2\pi\sigma_r}} exp\left[-\frac{1}{2}\left(\frac{(r-d_{x,y})}{\sigma_r^2}\right)\right] \quad (5)$$

Onde r é a distância medida pelo sensor, $d_{x,y}$ é a distância euclidiana entre a célula em que o sensor se encontra e a célula $m_{x,y}$ que está sendo analisada e σ_r^2 é a variância que representa a imprecisão do sensor. O cálculo da variância pode ser obtido pela Equação (6). Os valores das constantes a e b foram calculadas por experimentação.

$$\sigma_r^2 = a.e^{b.r} \quad (6)$$

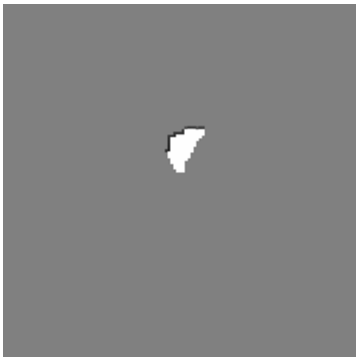


Figura 7. Exemplo de mapeamento dos dados apresentados na Figura 6

A Figura 7 apresenta o mapeamento da leitura dos sensores de alcance extraídos na Figura 6. Nela, pode-se observar que a área cinzenta representa uma área não explorada, a branca representa uma área livre e a preta representa uma área de obstáculos.

E. Controlador Fuzzy

A lista de sensores e a distância relacionada com cada um dos ângulos foi usado para obter a distância da frente, da esquerda e da direita do robô. A figura 8 ilustra como a lista de sensores foram distribuídas, as linhas em vermelho, verde e azul representam os sensores da direita, frente e da esquerda do robô, respectivamente. O CLF proposto foi modelado com base em conhecimento heurístico baseado no erro entre um referencial, que é a distancia que o robô deve se locomover em relação a parede (fixado em 0.5), e a distância mensurada pelo sensor lateral(que é a distancia mensurada pelo sensor da

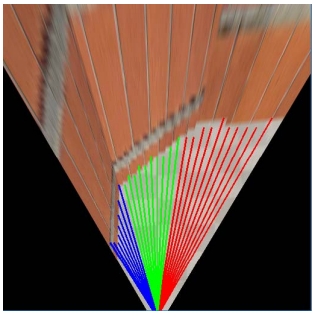


Figura 8. Ilustração da distribuição da lista de ângulos com Ns= 30 sensores.

direita ou da esquerda dependendo do lado que o robô está seguindo a parede).

$$\begin{bmatrix} w_d \\ w_e \end{bmatrix} = \begin{bmatrix} 1/r_d & b/(2r_d) \\ 1/r_e & -b/(2r_e) \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} \tag{7}$$

- V - Velocidade linear do robô
- W - Velocidade angular do robô
- r_d - Raio da roda direita
- r_e - Raio da roda esquerda
- w_e - Velocidade angular da roda esquerda
- w_d - Velocidade angular da roda direita
- b - Diâmetro da roda do robô

O objetivo é modelar um CLF que forneça como saída V e W, com a finalidade de encontrar os valores para w_e e w_d de acordo com a formulação mostrada na equação matricial 7.

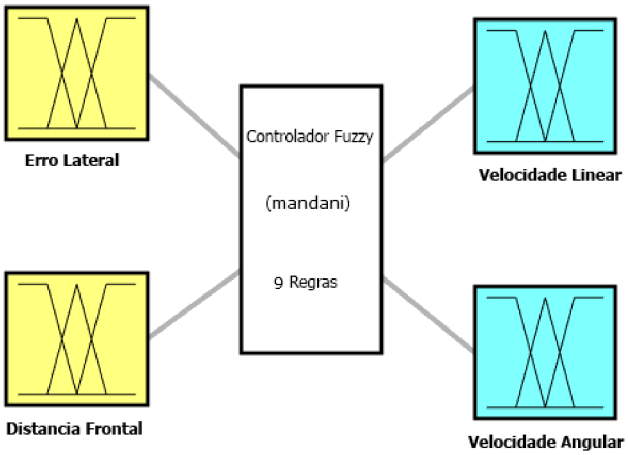


Figura 9. Sistemas de controle *Fuzzy*: 2 entradas, 2 saídas,9 regras.

Para o controle da distancia do robô em relação a parede foram necessárias duas variáveis linguísticas para representar as entrada, são elas: o erro lateral, que é a diferença entre o referencial e a distância medida pelo sensor lateral e a distância frontal, que é a distância medida pelo sensor frontal. O erro lateral possui três funções de pertinência: negativo, zero e positivo, que indicam o resultado da subtração entre o referencial e medida mensurada pelo sensor lateral. O sensor frontal também possui três funções de pertinência: baixo, médio e alto, que indicam a proximidade de algum obstaculo a frente do robô.

Tabela I
REGRAS FUZZY

Erro Lateral	Distância Frontal	Velocidade Linear	Velocidade Angular
Negativo	Baixo	Baixo	Positivo
Negativo	Médio	Médio	Negativo
Negativo	Alto	Médio	Negativo
Zero	Baixo	Baixo	Positivo
Zero	Médio	Alto	Zero
Zero	Alto	Alto	Zero
Positivo	Baixo	Baixo	Positivo
Positivo	Médio	Médio	Positivo
Positivo	Alto	Médio	Positivo

As variáveis linguísticas que representam as saídas do CLF são a velocidade linear, que possui 3 funções de pertinência (baixo, médio e alto) e a velocidade angular do robô que também possui 3 funções de pertinência (negativo, zero e positivo). O CLF proposto resultou em um total de 9 regras *fuzzy*, que são listadas na Tabela I. A estrutura do CLF proposto é mostrada na figura 9.

IV. RESULTADOS E DISCUSSÕES

Para validar o sistema foram feitos testes com dados gerados pelo simulador V-REP, tendo como cenário simulado do Departamento de Computação da Universidade Federal do Piauí como demonstrado na Figura 10.

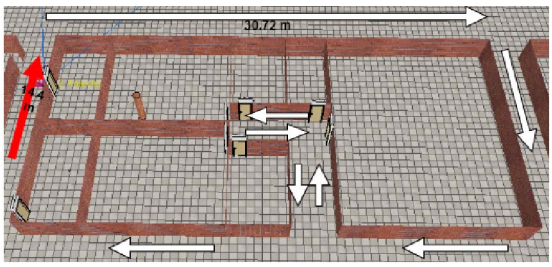
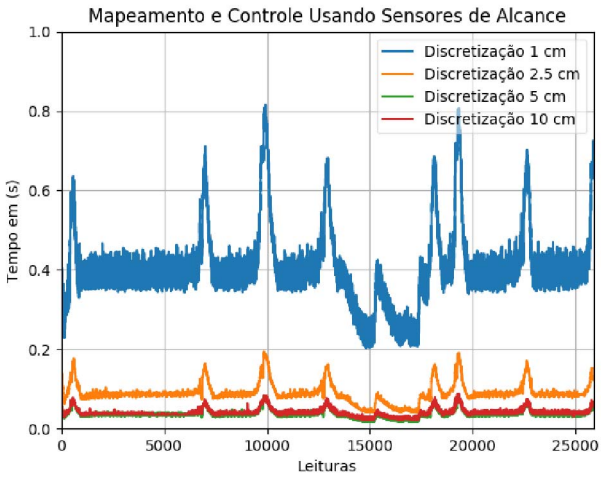
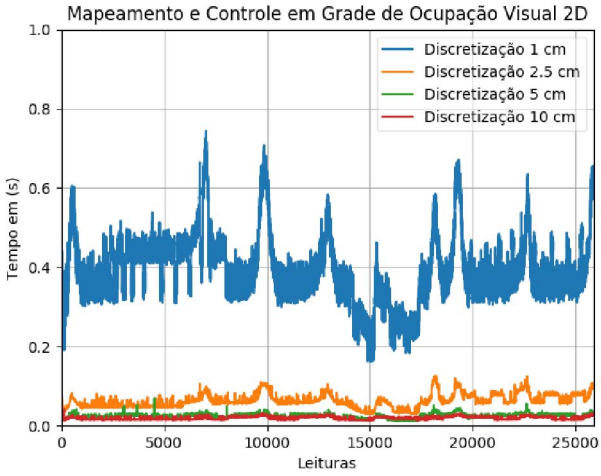


Figura 10. Cenário 1 utilizado nos experimentos.

Para implementação do sistema foram utilizadas a linguagem C++, a biblioteca de visão computacional OpenCV [23]*(Opensource Computer Vision library)* versão 3.1 para *Android* e a eFLL - uma biblioteca *fuzzy* para arduino e sistemas embarcados [24]. Os testes foram feitos em um *smartphone* LG Nexus 5 Quad-Core 2.3 GHz com 2GB de RAM e *Android* 6.1 [25]. Foram utilizadas 25880 amostras de imagens com resolução 512x512, com discretizações de 10, 5, 2.5 e 1 centímetros e uma lista de 30 sensores de alcance. Os gráficos da Figura 11, mostram o resultado em relação a tempo de execução. Para efeito de comparação, as mesmas amostras de imagens, submetidas às mesmas condições de discretização, foram testadas em uma implementação de mapeamento visual proposta por [18].



(a)

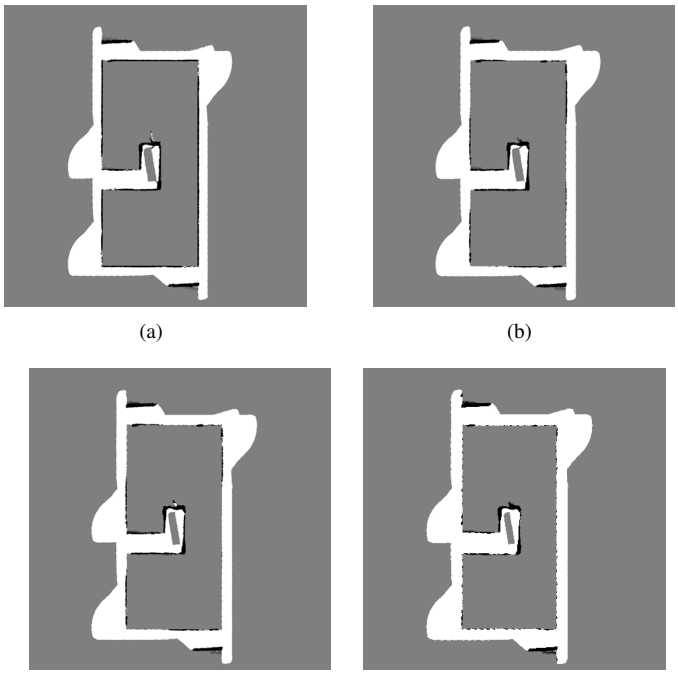


(b)

Figura 11. Gráficos com Tempos de Execução do Cenário 1: (a) Mapeamento em Grade com extração de Sensor de Alcance , (b) Mapeamento Visual em Grade Ocupação 2D

Como pode ser observado nos gráficos da Figura 11, os tempos de execução de ambas implementações se assemelham nas discretizações testadas. Durante a execução, há diversas variações no tempo de processamento representadas por picos e vales. Essas variações se devem à fase de cálculo dos sensores de alcance que, dependendo da quantidade de obstáculos encontrados, podem retornar mais ou menos leituras de sensores dentre a quantidade estabelecida, e isso pode pesar ou não no processamento total do sistema. Além disso, o tempo de processamento do controlador também influi nesses resultados. Com isso, pode-se afirmar, tomando como base a afirmação de [18] sobre sua abordagem de Mapeamento Visual, que os resultados são satisfatórios com funcionamento em tempo computacional aceitável para aplicações robóticas.

Além do tempo de execução, foram consideradas a discretização do mapa. A discretização diz respeito a quanto do



(a) (b) (c) (d)

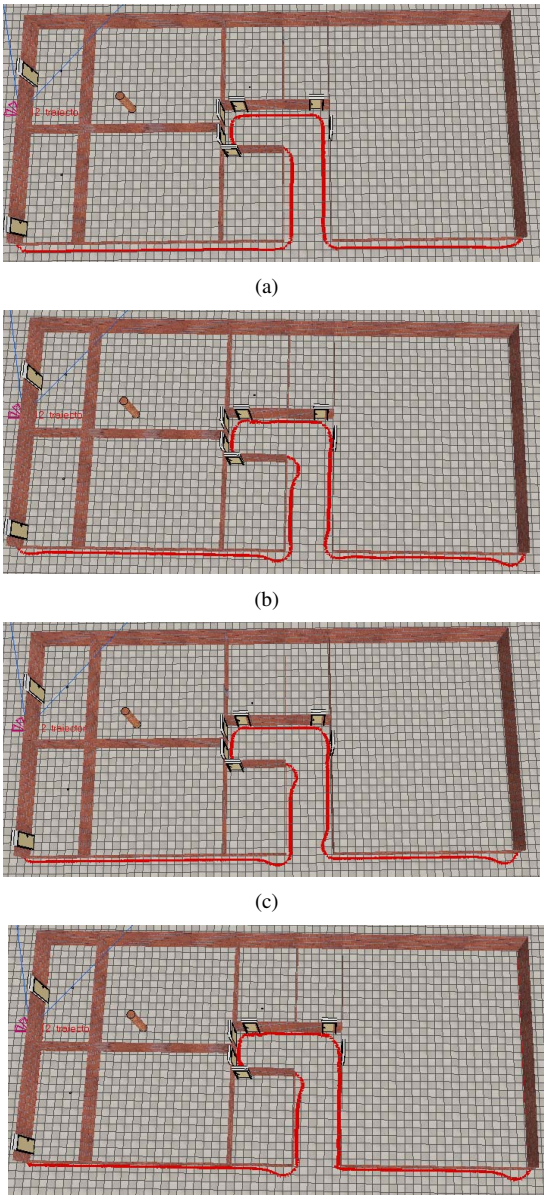
Figura 12. Mapas com diferentes discretizações: (a) Discretização de 1 cm, (b) Discretização de 2,5 cm (c) Discretização de 5 cm (d) Discretização de 10 cm. Essas imagens demonstram os efeitos da discretização na qualidade do mapa gerado

espaço do mundo real uma célula representa. Como o mapa resultante é uma imagem com informações de probabilidade, uma célula do mapa corresponde a um pixel da imagem. Assim, quando um mapa apresenta discretização de 1 cm, isso significa que cada pixel da imagem representa 1 cm do mundo. As discretizações são fatores importantes a serem considerados no mapeamento pois influenciam diretamente na qualidade dos mapas criados. Isso pode ser comprovado com os resultados da Figura 12, onde é possível observar que conforme a discretização da imagem aumenta, menos detalhado é o mapa e por consequência mais impreciso ele é. Porém, o tempo necessário para processar uma leitura para um mapa de 10 cm de discretização é bem menor do que a de um mapa de 1 cm, isso pode ser observado na Figura 11(a), onde o tempo para processar a leitura mais custosa (representada pelo terceiro pico) levou mais de 0.8 segundos com discretização de 1 cm e menos de 0.1 segundos com discretização 10 cm. Diante desses fatos, é possível concluir que o grande desafio, em se tratando de mapeamento em grade de ocupação, é aliar precisão com eficiência em relação a tempo processamento.

A Tabela II apresenta um conjuntos de métricas que avaliam o CLF. Dentre as métricas avalidas estão a velocidade linear média(VLM), velocidade angular média (VAM) e a distância média em relação a parede (DMRP). Neste trabalho a DMRP foi usado com o propósito de verificar a proximidade em que o robô se locomove em relação a parede, ou seja, quanto maior o DMRP maior é o risco do robô colidir com alguma parede. Para essa métrica o robô obteve uma boa avaliação

Tabela II
MÉTRICAS DE AVALIAÇÃO

DISCRETIZAÇÃO	DMRP	VLM	VAM
1	$0,788305 \pm 0,474917$	$0,217253 \pm 0,052613$	$-0,009731 \pm 0,09752$
2,5	$0,836973 \pm 0,560512$	$0,190484 \pm 0,070336$	$0,001832 \pm 0,117782$
5	$0,865524 \pm 0,523841$	$0,171744 \pm 0,072716$	$-0,008012 \pm 0,135449$
10	$0,787821 \pm 0,643775$	$0,175454 \pm 0,080704$	$0,032451 \pm 0,133655$



(a)

(b)

(c)

(d)

Figura 13. Parte do trajeto percorrido pelo robô com a discretização: a) discretização 1 b) discretização 2.5 c) discretização 5 d)discretização 10

pois o mesmo consegue se estabilizar em uma distancia segura em relação a parede. A VAM tem o proposito de avaliar a suavidade com que o robô consegue executar a trajetória, verificando também a suavidade em que o robô percorre as curvas. Nessa métrica, quanto mais próximo de zero, significa que o robô se estabiliza mais rápido em um posição segura em relação a parede, voltando a seguir a parede de forma linear. A VLM é uma métrica usada para avaliar a velocidade em que o robô percorreu a trajetória. Diante dos dados expostas na Tabela II e do que foi explanado sobre as métricas, nota-se que o CLF conseguiu obter resultados desejados para todos os valores de discretização, seguindo a parede do cenário proposto sem colidir com nenhum obstáculo. Para um resultado visual é mostrado na Figura 13 parte do trajeto executado pelo robô, com diferentes tamanhos de discretização, nota-se que quanto maior for o tamanho da discretização maior será a dificuldade para controlar o robô de forma suave e estável para o comportamento de seguidor de parede.

V. CONCLUSÃO

Este artigo apresentou uma nova abordagem de sensoriamento utilizando somente uma câmera como sensor, a partir das imagens coletadas foi desenvolvido uma técnica de extração de informações de distâncias de maneira que elas se comportassem como sensores de alcance. Essas informações são utilizadas como entrada para um controlador baseado em lógica *fuzzy* e para mapeamento em grade de ocupação. O sistema foi implementado para um *cellbot*, com o objetivo de apresentá-lo como uma ferramenta científica e demonstrar suas potencialidades para ensino e desenvolvimento de aplicações robóticas de fácil implementação e baixo custo.

De acordo com os resultados apresentados, pôde-se observar que o uso das informações das imagens em forma de sensores de alcance apresentou resultados satisfatórios quando comparado com outras técnicas que utilizam visão monocular para mapeamento e navegação. A grande vantagem dessa abordagem para o mapeamento visual é a facilidade de implementação, pois o uso de sensores de alcance extraídos da imagem permite utilizar o algoritmo clássico de mapeamento em grade de ocupação sem grandes modificações. A utilização das distancias extraídas da imagem também têm como vantagem a economia no uso de sensores, pois essas distancias são usadas tanto para o mapeamento quanto para o controle, resultando também em um baixo custo computacional.

Como trabalhos futuros temos, a realização de testes em um ambiente real e comparação com outros tipos de controlador como por exemplo o controlador fuzzy hierárquico e o controlador proporcional integral derivativo.

REFERÊNCIAS

[1] R. V. Aroca, R. B. Gomes, D. M. Tavares, A. A. S. Souza, A. M. F. Burlamaqui, G. A. P. Caurin, and L. M. G. Goncalves, "Increasing students' interest with low-cost cellbots," *IEEE Trans. on Educ.*, vol. 56, no. 1, pp. 3–8, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1109/TE.2012.2214782

[2] A. A. Souza and L. M. Goncalves, "2.5-dimensional grid mapping from stereo vision for robotic navigation," in *Brazilian Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012*. IEEE, 2012, pp. 39–44.

[3] C.-H. Kuo *et al.*, "Development of a fuzzy logic wall following controller for steering mobile robots," in *International Conference on Fuzzy Theory and Its Applications (iFUZZY), 2013*. IEEE, 2013, pp. 7–12.

[4] T. Kumbasar and H. Hagras, "A type-2 fuzzy cascade control architecture for mobile robots," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2013*. IEEE, 2013, pp. 3226–3231.

[5] W. L Rodrigues Junior, D. M Reis, R. P Bezerra Neto, R. S Machado, W. AS Silva, J. O Brito Neto, R. AL Rabêlo, and A. M Santana, "Aplicação de controladores fuzzy e proporcional para um robô seguidor de parede autônomo em ambiente estático," *Revista de Sistemas e Computação-RSC*, vol. 6, no. 1, 2016.

[6] H. A. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Transactions on Fuzzy systems*, vol. 12, no. 4, pp. 524–539, 2004.

[7] M. A. Melgarejo, C. M. Munoz, and L. Leottau, "A hierarchical design approach for interval type-2 fuzzy controllers applied to mobile robots," *International Journal of Robotics and Automation*, vol. 27, no. 3, p. 330, 2012.

[8] R. Tanscheit, "Sistemas fuzzy," *Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro*, 2004.

[9] W. Gueaieb and M. S. Miah, "An intelligent mobile robot navigation technique using rfid technology," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1908–1917, 2008.

[10] B. Achmad and M. N. Karsiti, "Visual-based fuzzy navigation system for mobile robot: Wall and corridor follower," in *International Conference on Intelligent and Advanced Systems, 2007. ICIAS 2007.* IEEE, 2007, pp. 244–248.

[11] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *1985 IEEE International Conference on Robotics and Automation. Proceedings.*, vol. 2, Mar 1985, pp. 116–121.

[12] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[13] C. Yu, V. Cherfaoui, and P. Bonnifait, "An evidential sensor model for velodyne scan grids," in *13th International Conference on Control Automation Robotics & Vision (ICARCV), 2014*. IEEE, 2014, pp. 583–588.

[14] M. Häselich, B. Jöbgen, F. Neuhaus, D. Lang, and D. Paulus, "Markov random field terrain classification of large-scale 3d maps," in *IEEE International Conference on Robotics and Biomimetics (ROBIO), 2014*. IEEE, 2014, pp. 1970–1975.

[15] S. Jia, H. Shen, X. Li, W. Cui, and K. Wang, "Autonomous robot exploration based on hybrid environment model," in *International Conference on Information and Automation (ICIA), 2012*. IEEE, 2012, pp. 19–24.

[16] M. Liu, S. Huang, and G. Dissanayake, "Feature based slam using laser sensor data with maximized information usage," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*. IEEE, 2011, pp. 1811–1816.

[17] J.-D. Fossel, K. Tuyls, and J. Sturm, "2d-sdf-slam: A signed distance function based slam frontend for laser scanners," in *International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ*. IEEE, 2015, pp. 1949–1955.

[18] A. M. Santana, K. R. Aires, R. M. Veras, and A. A. Medeiros, "An approach for 2d visual occupancy grid map using monocular vision," *Electronic Notes in Theoretical Computer Science*, vol. 281, pp. 175 – 191, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571066111001824

[19] A. Murarka and B. Kuipers, "A stereo vision based mapping algorithm for detecting inclines, drop-offs, and obstacles for safe local navigation," in *International Conference on Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ*. IEEE, 2009, pp. 1646–1653.

[20] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Monocular vision based slam for mobile robots," in *18th International Conference on Pattern Recognition, 2006. ICPR 2006.*, vol. 3. IEEE, 2006, pp. 1027–1031.

[21] C. Plagemann, F. Endres, J. Hess, C. Stachniss, and W. Burgard, "Monocular range sensing: A non-parametric learning approach," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.* IEEE, 2008, pp. 929–934.

[22] C. R. GmbH, "V-REP," http://www.coppeliarobotics.com/, 2010, [Online; Acessado em 29/04/2017].

[23] I. Corporation, W. Garage, and Itseez, "OpenCV," http://opencv.org/, 2006, [Online; Acessado em 29/04/2017].

[24] A. J. de Oliveira Alves, "efll - a fuzzy library for arduino and embeded systems," http://www.zerokol.com/2012/09/arduinofuzzy-fuzzy-library-for-arduino.html, 2012, [Online; Acessado em 10/05/2016].

[25] G. Inc. and O. H. Alliance, "Android," http://www.android.com/, 2008, [Online; Acessado em 29/04/2017].