

# Navegação Baseada em Visão Omnidirecional Utilizando Campos Potenciais

Douglas G. Macharet  
Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais, Brasil  
doug@dcc.ufmg.br

## Resumo

*Esse trabalho apresenta o desenvolvimento de um sistema de navegação para robôs móveis que baseia-se em imagens omnidirecionais adquiridas através de um sistema catadióptrico. As imagens são analisadas e então fornecidas à um controlador que faz uso da técnica de campos potenciais, permitindo assim ao robô realizar uma navegação segura em um ambiente desconhecido.*

## 1. Introdução

Uma das principais linhas de pesquisa na área da robótica móvel está voltada ao estudo de como dar uma maior autonomia aos robôs, principalmente nas questões ligadas à sua locomoção.

A navegação consiste basicamente em levar um robô de um ponto a ponto de forma segura. Para realizar a navegação o robô pode utilizar diversos tipos de sensores que irão fornecer informações relacionadas ao ambiente onde ele está inserido, e dessa forma permitir com que determinadas decisões sejam tomadas baseadas nessas informações.

As informações do ambiente estão principalmente ligadas ao fato de existirem objetos que interferem ou impedem a navegação do robô, objetos esses que são considerados como obstáculos.

Diversos sensores podem ser utilizados para a detecção de obstáculos no caminho do robô, principalmente *range sensors*, por exemplo, sonares e *lasers*, além de abordagens que utilizam outros tipos de sensores como câmeras. É importante salientar que a escolha de qual tipo de sensor será usado está diretamente ligada à atividade que o robô deverá realizar, assim como o ambiente onde essa tarefa será realizada.

Neste trabalho deseja-se fazer com que um robô seja capaz de realizar uma navegação autônoma em um ambiente interno, para isso será utilizada uma lente que permite obter uma visão de 360° ao redor do robô, a Figura 1 apre-

senta uma imagem capturada. Decidiu-se por utilizar esse tipo de visão pois assim é possível ao robô examinar grande parte da região ao seu redor, nota-se por exemplo através da imagem que obstáculos dispostos em diferentes posições em volta do robô podem ser visualizados.

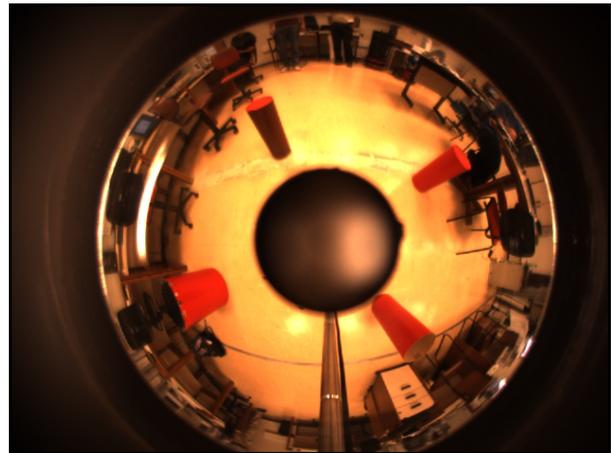


Figura 1. Visão omnidirecional.

Durante a navegação do robô, a partir das imagens coletadas, o robô deve ser capaz de identificar possíveis obstáculos, e para o desvio deverá ser usado a técnica de campos potenciais. As imagens da câmera serão utilizadas como se fossem um “*range sensor* visual”, uma vez deve-se calcular e utilizar a distância que o obstáculo está do robô para se tomar uma decisão.

Utilizando a câmera temos um ganho em relação aos *range sensors* convencionais pois com apenas um sensor já é possível adquirir dados de toda a área ao redor do robô. Outra vantagem é que sensores como *laser* efetua suas leituras em apenas um plano, já a câmera consegue extrair mais informações, como por exemplo é possível observar que através da câmera seria possível inferir a altura do obstáculo presente na Figura 1.

Ao longo do texto a forma como as características da cena são identificadas e utilizadas para a navegação segura do robô será melhor detalhada.

## 2. Trabalhos Relacionados

A principal idéia desse trabalho está na detecção de determinadas características da imagem que possam auxiliar na navegação, pretendem-se utilizar por exemplo detecção de bordas, considerando-se que obstáculos presentes no ambiente terão bordas bem detectáveis.

A imagem omnidirecional pode ser utilizada basicamente de duas maneiras, diretamente como foi capturada (Figura 1), ou então, após já ter sido retificada.

Em [6] é apresentada uma técnica que permite trabalhar diretamente com a imagem capturada pela câmera omnidirecional e realizar a detecção de características. Apesar de ter se mostrado bem eficiente, a técnica pode ser utilizada apenas para a detecção de linhas verticais na cena, ou seja, linhas que sejam radiais na imagem, o que restringe muito a sua utilização, ainda mais se a maior parte da informação coletada é basicamente do piso onde o robô está trafegando (como é o caso deste trabalho).

Apesar de se conseguir extrair determinadas características da imagem original, essas são limitadas devido à grande deformação apresentada na imagem omnidirecional, logo, é interessante utilizar-se de técnicas de retificação permitindo assim a utilização dos algoritmos convencionais

Duas formas de retificação podem ser utilizadas, a retificação *Panorâmica* [3], em que a imagem é transformada em uma imagem quadrangular, ou então, a transformação da imagem na dita *Bird's Eye View*, que realiza uma correção radial e deixando a imagem como se tivesse sido obtida através de uma câmera convencional posicionada acima da cena.

No trabalho apresentado em [4] podemos observar a utilização de imagens panorâmicas para a navegação de um robô em corredores. Inicialmente o robô realiza uma navegação coletando as imagens que em seguida são adicionadas a um mapa, então, caso o robô tenha novamente uma determinada seqüência de “visões” ele saberá onde está. Os principais problemas envolvidos nessa técnica estão na necessidade de uma captura prévia das imagens, e também na necessidade do ambiente ser completamente estático, ou seja, durante a navegação o que o robô está “vendo” deve ser exatamente igual ao que foi capturado anteriormente.

Imagens omnidirecionais transformadas em *Bird's Eye View* produzem bons resultados, principalmente quando a câmera está capturando informações do piso. Em [5] é apresentado um processo de calibração de câmeras omnidirecionais, assim como remapeamento das imagens para *Bird's Eye View* e a utilização dessas imagens para a detecção

da pose de um robô móvel. O método apresentado é bem genérico e pode ser utilizado em diversos tipos de sistemas catadióptricos. A detecção da pose dos robôs é feita por meio de cor, sendo que a câmera está fixa no alto do ambiente onde os robôs estão se movimentando.

Já em [10] podemos observar a utilização de imagens omnidirecionais de uma câmera que está afixada sobre o robô, como é o caso deste trabalho. As imagens são remapeadas em *Bird's Eye View*, e por essas imagens é realizada uma busca por características que identifiquem o corredor, fazendo assim que com o robô trafegue sempre pelo meio do corredor. Essa técnica é interessante, uma vez que não necessita de um conhecimento prévio do ambiente, entretanto, esse trabalho foi aprofundado em [9], onde também é abordado o problema das diferentes quantidades de iluminação que um ambiente pode receber durante um dia, sendo esse um dos grandes problemas na utilização de sistemas baseados em visão.

Em [7] também é apresentado uma técnica de remapeamento de imagens omnidirecionais em *Bird's Eye View*. Através dessas imagens remapeadas foi desenvolvido um método que detectava determinadas características no ambiente e permitia ao robô executar uma tarefa de *path following*. Os resultados encontrados foram muito bons, principalmente uma vez que mesmo o caminho não sendo contínuo o robô não se desviava. Entretanto, a detecção de marcos era auxiliada, uma vez que o caminho que devia ser detectado se diferenciava bastante do restante do ambiente.

O principal problema envolvido na utilização de técnicas de retificação está no custo computacional envolvido, por isso, pretende-se desenvolver um método que utilize as imagens da maneira como elas são coletadas.

## 3. Metodologia

As diferentes técnicas de visão computacional empregadas no desenvolvimento do trabalho serão detalhadas nesta seção.

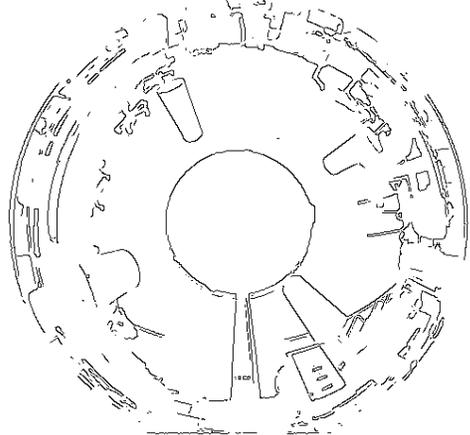
### 3.1. Detecção de obstáculos

Um dos problemas envolvidos em navegação autônoma está relacionado à capacidade de se conseguir identificar objetos que vão interferir na locomoção do robô, ciente desses obstáculos deve-se então traçar uma estratégia de maneira a conseguir-se superar esses obstáculos e continuar com a navegação.

Nesse trabalho, a detecção de obstáculos no ambiente consistirá em utilizar um detector de bordas sobre a cena capturada, deve ser calculada então a distância entre os pontos classificados como pertencentes à uma borda e o centro do imagem, e a partir disso o controle baseado em campos potenciais deve inferir a força de repulsão exercida por

cada obstáculo. Para a detecção de bordas é utilizado o algoritmo de *Canny*.

A Figura 2 apresenta a imagem original e o resultado obtido após a detecção de bordas executado sobre a cena apresentada na Figura 1.



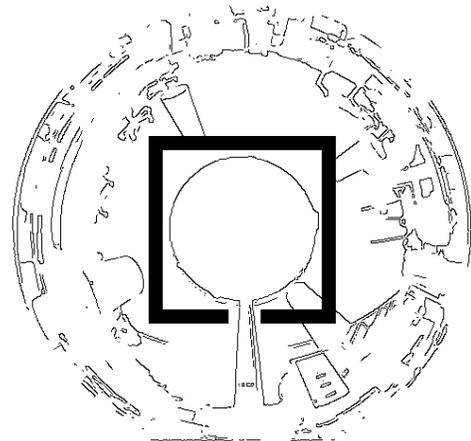
**Figura 2. Resultado da execução do algoritmo de *Canny*.**

Observa-se através da Figura 2 após a detecção das bordas presentes na cena é possível indentificar os quatro obstáculos dispostos na região próxima ao redor do robô, entretanto, a câmera também captura informações de objetos muito distantes do robô e que não exercem nenhuma influência ou restrição à navegação local do robô.

Para solucionarmos o problema da detecção de objetos irrelevantes localmente ao robô decidiu-se por criar uma janela de análise na vizinhança próxima ao robô, ou seja, só serão considerados como obstáculos objetos que estejam suficientemente próximos ao robô. Essa janela deve ignorar tanto os *pixels* de borda encontrados no círculo central quanto os presentes na haste de sustentação da câmera, logo, a partir desses fatos decidiu-se por utilizar uma região quadrada com uma abertura na região da haste.

A Figura 3 apresenta a área onde é verificada a existência de obstáculos.

Após a criação da janela de análise é possível observar que mesmo todos os obstáculos ao redor do robô serem percebidos apenas três teriam algum tipo de influência em sua navegação.



**Figura 3. Região onde verifica-se a presença de obstáculos.**

### 3.2. Cálculo do centro da imagem

Optou-se por realizar o cálculo do centro da imagem ao invés de simplesmente basear-se no *pixel* central, isso porquê a lente utilizada pode possuir uma inclinação em relação à câmera que ela está acoplada, o quê mudaria o centro da imagem efetivamente capturada e prejudicaria os demais algoritmos que serão executados sobre ela. Dessa forma também evita-se a necessidade de re-calibração à cada vez que o sistema fosse desmontado. Apesar da possibilidade desse cálculo ser realizado apenas um vez decidiu-se por executá-lo à cada iteração, tornando o sistema mais robusto a qualquer variação do centro da imagem.

Como pode ser visto na Figura 1 o centro da imagem capturada possui um círculo, esse círculo é referente à uma haste presente no centro da lente. Logo, como esse círculo sempre representará o centro correto da imagem e por ser bem definido (permitindo assim uma boa detecção) ele será utilizado como referência.

Para a localização desse círculo central utilizou-se a transformada de *Hough* circular, sendo que o centro do círculo detectado será o centro da imagem. A entrada da transformada de *Hough* deve ser uma imagem binária, no caso desse trabalho a imagem a ser utilizada é a imagem contendo as bordas do ambiente já previamente calculada, o que tende a tornar a execução do algoritmo um pouco mais eficiente.

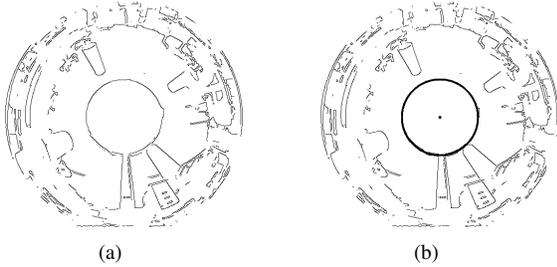
A Equação 1 representa um círculo.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (1)$$

Onde  $a$  e  $b$  é o centro do círculo na direção  $x$  e  $y$  e  $r$  é o raio. Assim, a representação paramétrica do círculo pode ser vista na Equação 2.

$$\begin{aligned} x &= a + r\cos(\theta) \\ y &= b + r\sin(\theta) \end{aligned} \quad (2)$$

A Figura 4 apresenta o resultado após a execução do algoritmo de detecção de bordas e da transformada de *Hough* sobre a cena vista na Figura 1.



**Figura 4. 4(a) Bordas detectadas. 4(b) Centro circular detectado.**

O resultado obtido é muito bom, pois, como pode ser observado foi possível detectar o círculo central de maneira bem precisa.

Entretanto era necessário realizar um teste de robustez do cálculo para se observar como ele se comportaria caso o círculo central não fosse detectado. Para realizar esse experimento seria necessário introduzir o robô em um ambiente onde não fosse possível realizar a detecção do círculo central, para isso o robô foi colocado em um local com um piso bem escuro, fazendo assim com que o círculo central fosse considerado como parte do piso, e dessa forma não fosse possível detectar bordas ao seu redor.

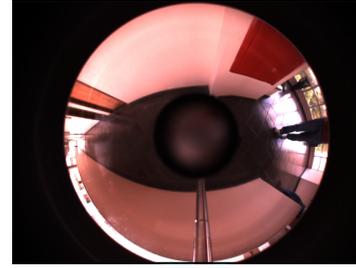
A Figura 5 apresenta esse teste realizado, assim como os resultados obtidos.

Observar-se que nessas condições o círculo central realmente não foi detectado, entretanto, a própria imagem em si possui um formato circular, e ao ser adicionada à uma moldura quadrada acaba por deixar uma grande área sem dados válidos (preta) ao seu redor, e dessa forma possuindo uma borda externa bem detectável, como ocorrido no caso acima.

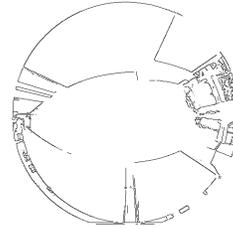
Logo, temos que independentemente do ambiente por onde o robô transitar sempre será possível calcular-se o centro da imagem capturada.

### 3.3. Campos potenciais

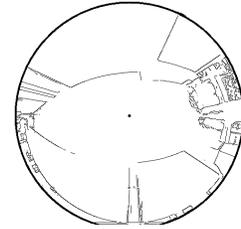
A principal característica da técnica de campos potenciais, é que não há a necessidade de nenhuma pré-computação antes de o robô dar início a sua navegação, ela funciona de uma maneira extremamente reativa.



(a)



(b)



(c)

**Figura 5. 5(a) Ambiente com piso escuro. 4(a) Bordas detectadas (círculo central não detectado). 5(c) Borda externa circular detectada.**

Toda a movimentação do robô é baseada nas forças que atuam sobre ele, onde o goal realiza uma força de atração e os obstáculos realizam uma força de repulsão. As forças são somadas e a força resultante é fornecida à um controlador que irá transformar essas forças na velocidade angular e linear que devem ser aplicadas em determinado momento.

Temos a seguinte fórmula para o cálculo da força repulsiva exercida pelos obstáculos:

$$\begin{aligned} f_x &= -K_r * \cos(\text{angle}) * 1/\text{range} \\ f_y &= -K_r * \sin(\text{angle}) * 1/\text{range} \end{aligned}$$

Onde  $K_r$  é uma constante de repulsão,  $\text{angle}$  é o ângulo entre o robô e o obstáculo e  $\text{range}$  é a distância a qual o obstáculo se encontra.

Já para o cálculo da força atrativa realizada pelo goal temos:

$$\begin{aligned} f_x &= K_a * \text{distance} * \cos(\text{angle}) \\ f_y &= K_a * \text{distance} * \sin(\text{angle}) \end{aligned}$$

Onde  $K_a$  é uma constante atrativa,  $\text{distance}$  é a distância entre a posição atual do robô e o goal, e  $\text{angle}$  é o ângulo entre a orientação do robô e o goal.

Calculada a resultante das forças é aplicado o seguinte controlador:

$$v = K_l * (fx * \cos(\theta) + fy * \sin(\theta))$$

$$\omega = K_w * (\arctan 2(fy, fx) - \theta)$$

Onde  $K_l$  e  $K_w$  são constantes, e  $\theta$  é a orientação do robô, lembrando que o *range* é a distância calculada a partir da detecção de bordas, conforme visto na Seção 3.1.

#### 4. Experimentos

A Figura 6 apresenta a montagem do robô e câmera utilizados para a realização dos testes. Os experimentos utilizando o robô foram realizados em um ambiente com diferentes tipos de obstáculos presentes.

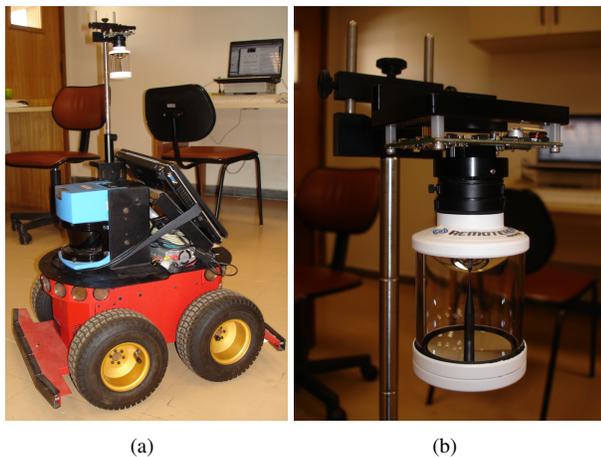


Figura 6. 6(a) Robô completo utilizado. 6(b) Sistema catadióptrico.

Durante os experimentos realizados o robô foi capaz de desviar de grande parte dos obstáculos encontrados durante sua trajetória, entretanto, algumas dificuldades foram encontradas para determinadas situações.

Um problema inicialmente encontrado foi para obstáculos que se encontravam à mesma altura da lente, como por exemplo um tampo de mesa. O algoritmo não é capaz de identificar bordas nessa situação, fazendo assim com que o robô colida com esses tipos de obstáculos.

Também foi possível observar que o algoritmo não é capaz de detectar as bordas de certos objetos em determinadas posições, como por exemplo objetos muito arredondados ou com cores semelhantes ao da região onde o robô está inserido. Ou seja, os melhores resultados foram obtidos principalmente na presença de objetivos não circulares e com cores bem distintas do ambiente, como por exemplo um pé de cadeira preto em um piso claro.

Outro problema esperado e evidenciado no uso da técnica de campos potenciais é a presença de mínimos locais, ou seja, em determinados momentos as forças existentes tendem a entrar em equilíbrio e o robô acaba por entrar em *loop*.

A Figura 7 mostra uma visão do local onde o robô realizou sua navegação.



Figura 7. Ambiente de navegação.

#### 5. Conclusões e Trabalhos Futuros

A implementação realizada obteve bons resultados e foi capaz de fazer com que o robô navegasse em um ambiente *indoor* em segurança desviando-se dos obstáculos presentes em sua trajetória.

Entretanto, algumas dificuldades foram encontradas, principalmente na etapa de detecção de bordas, o que fazia com que certos objetos em determinadas posições não tivessem suas bordas detectadas, e assim o robô não fosse capaz de se desviar desses obstáculos. Sendo assim, seria interessante o desenvolvimento de uma nova maneira de se identificar obstáculos presentes na cena, deixando assim o algoritmo como um todo mais robusto. Outro problema que leva à não detecção de bordas está relacionado à cor do objeto em relação ao ambiente, ou seja, se eles possuírem cores semelhantes também torna a detecção de bordas mais difícil, portanto, um próximo passo seria uma segmentação das cores da imagem de forma a permitir que as bordas sejam melhor detectadas.

Outra melhoria possível seria ao invés de se analisar uma área quadrada ao redor do robô ser analisada uma área circular (mais parecida com o formato do robô), permitindo detectar assim apenas objetos realmente importantes para a navegação do robô. Na implementação atual a principal evidência desse problema está na avaliação que é feita nas quinas da área quadrada que detectam objetos relativamente distantes do robô (apesar de que esses objetos por estarem longe exercerão pouca influência no controlador baseado em campos potenciais).

Ao se utilizar uma região circular para a análise seria interessante também não haver a necessidade de preocupar com a haste de sustentação da câmera, portanto, sugere-se também realizar uma subtração de fundo na imagem, permitindo assim deixar a imagem apenas com os dados importantes.

Já a navegação através de campos potenciais permite com que o robô possa agir de maneira mais autônoma, inclusive desviando de obstáculos que não se encontravam no ambiente, o que faz com que ele seja muito eficiente para ambientes mais dinâmicos. Porém, não é garantido que um caminho para o goal será sempre encontrado, esse problema ocorre principalmente devido ao aparecimento de mínimos locais, o que faz com que as forças atuando sobre o robô se anulem e ele pare de navegar. Assim, seria importante desenvolver maneiras para que caso o robô esteja em um mínimo local ele identifique essa situação e consiga resolvê-la, de forma com que ele sempre alcance o objetivo.

## Referências

- [1] J. Gaspar, E. Grossmann, and J. Santos-Victor. Interactive reconstruction from an omnidirectional image. In *Proc. of the 9th International Symposium on Intelligent Robotic Systems (SIRS2001)*, 2001.
- [2] J. Gaspar and J. Santos-Victor. Visual path following with a catadioptric panoramic camera. In *Proc. of the 7th International Symposium on Intelligent Robotic Systems (SIRS'99)*, 1999.
- [3] V. G. Junior and J. O. Junior. Development of an omnidirectional vision system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28(1):58–68, 2006.
- [4] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Exploration and navigation in corridor environment based on omniview sequence. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS'00)*, 2000.
- [5] F. G. Pereira, C. C. Gava, R. F. Vassallo, and M. S. Filho. Calibração de sistemas catadióptricos e detecção da pose de robôs móveis por segmentação de imagens omnidirecionais. In *Anais do VII Simpósio Brasileiro de Automação Inteligente (SBAI/IEEE-LARS'05)*, 2005.
- [6] D. Scaramuzza, N. Criblez, A. Martinelli, and R. Siegwart. Robust feature extraction and matching for omnidirectional images. In *Proc. of the 6th International Conference on Field and Service Robotics (FSR'07)*, 2007.
- [7] R. Vassallo, L. Encarnação, J. Santos-Victor, and H.-J. Schneebeli. Bird's eye view remapping and path following based on omnidirectional vision. In *Anais do XV Congresso Brasileiro de Automática (CBA'04)*, 2004.
- [8] R. F. Vassallo, J. Santos-Victor, and H. J. Schneebeli. A general approach for egomotion estimation with omnidirectional images. *omnervis*, 00:97, 2002.
- [9] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omnidirectional vision for robot navigation. In *Proc. of the IEEE Workshop on Omnidirectional Vision (Omnivis'00)*, 2000.
- [10] N. Winters and J. Santos-Victor. Omni-directional visual navigation. In *Proc. of the 7th International Symposium on Intelligent Robotic Systems (SIRS'99)*, 1999.