

# Medição de distância e altura de bordas horizontais com visão monocular linear para robôs móveis.

Nicolau Leal Werneck  
nwerneck@usp.br

Flavio Sales Truzzi  
flavio.truzzi@poli.usp.br

Anna Helena Reali Costa  
anna.reali@poli.usp.br  
LTI-PCS-USP

Escola Politécnica da Universidade de São Paulo  
Av. Prof. Luciano Gualberto, trav.3, n.158  
São Paulo - SP - Brasil

## Resumo

*Este artigo apresenta um método para localizar espacialmente objetos detectados na coluna central de imagens obtidas por uma câmera em movimento. O movimento é restrito para a mesma direção do eixo ótico, e em velocidade constante, como ocorre com frequência em certos robôs móveis. Pontos de interesse correspondentes são encontrados em pares de imagens através de correlação, e as coordenadas do ponto no espaço são determinadas por triangulação. São apresentados os resultados da aplicação do método em uma simulação e em um experimento com uma câmera sobre trilhos.*

## 1. Introdução

Este artigo apresenta um sistema direcionado para a detecção de bordas horizontais em um ambiente a partir de imagens obtidas por uma câmera em movimento. Estes dados podem ser utilizados para localizar obstáculos como paredes ou ainda *obstáculos negativos*, como penhascos e escadas para descida. Trata-se de uma aplicação de técnicas clássicas de Visão Computacional, porém com restrições poucas exploradas: epipolos nos centros das imagens e análises restritas a uma reta da imagem.

A detecção de obstáculos próximos de uma câmera através de bordas horizontais que se assumem estar no nível do solo é uma possibilidade já explorada por vários pesquisadores da área de Robótica Móvel [7, 4, 3]. Estas técnicas possuem em comum a desvantagem de não poderem diferenciar entre bordas que delimitam paredes, quedas, ou simples desenhos no chão. Elas também não li-

dam corretamente com obstáculos suspensos, onde existe um vão entre a borda e o chão.

O sistema aqui proposto pretende superar estas limitações de duas formas: permitindo realizar a medição tanto da distância quanto da altura a que estas bordas se encontram do observador, e ainda possibilitando a detecção de oclusões.

A assunção de que se tratam de objetos retilíneos que cruzam o caminho da câmera é explorada com a limitação da análise apenas à reta vertical central da câmera. Esta restrição tem potencial para tornar o algoritmo bastante veloz, permitindo criar um programa que funcione ininterruptamente em robôs móveis.

A restrição das análises a uma única reta sobre as imagens de uma câmera a transforma em um sensor visual linear. Como o estudo apresentado aqui também faz uso de apenas uma câmera, justifica-se assim o termo “visão monocular linear” do título.

### 1.1. Aplicação em robótica móvel

O problema mais fundamental da robótica móvel é a localização de objetos no ambiente do robô para permitir que este se locomova com segurança e realize com sucesso suas tarefas. Para tal é preciso que os robôs sejam dotados de sensores que determinem a localização de obstáculos e de objetos relevantes ao seu redor. Os dados destes sensores são geralmente obtidos do ponto de vista do robô, e variam conforme este se locomove.

Alguns dos sensores mais comuns para localização de obstáculos em robótica móvel são os que retornam, para determinada direção, o valor da distância até a obstrução mais próxima. É o caso de sensores a laser, sonares e armações estereoscópicas. Para atividades como reconhecimento de

objetos e condições ambientais outros sensores são mais indicados, tal como câmeras fotográficas.

Câmeras possuem vantagens como o baixo custo e facilidade de manutenção, e por isso é antigo o desejo de roboticistas em aplicá-las até mesmo para tarefas a que elas não são totalmente adequadas, como a medição da distância de objetos. Para utilizar armações multioculares é necessário realizar um levantamento dos parâmetros de seu modelo [6]. Elas também impõem dimensões mínimas consideráveis ao robô, já que o aumento da distância entre as câmeras beneficia a princípio o funcionamento da técnica.

Uma outra possibilidade é a visão monocular. Técnicas de navegação com visão monocular podem ser necessárias em ambientes vastos, onde não se pode estimar distâncias sem mover o robô. Mas a visão monocular também é interessante por permitir criar robôs bastante pequenos. O funcionamento do sistema também é diferente por não possuir uma etapa claramente definida de processamento de imagens para produção de dados. A visão monocular permite portanto explorar formas de construção e operação de robôs móveis que não são possíveis nem estimuladas pela visão multiocular.

Na visão monocular os dados obtidos em cada instante são apenas as direções em que se encontram cada ponto detectado, mas não a distância. Para determinar completamente a posição destes pontos é preciso integrar estes dados ao longo do tempo utilizando algum algoritmo. Já existem, por exemplo, algoritmos que realizam a tarefa de localização e mapeamento simultâneos utilizando apenas visão monocular [1, 2]. Estas técnicas são bastante diferentes das utilizadas com visão multiocular, onde é mais fácil aproveitar técnicas para sensores de distância convencionais.

O sistema aqui apresentado busca explorar idéias que podem ser utilizadas por sistemas monoculares mais sofisticados. O sistema desenvolvido também aponta para uma forma simples e auto-contida de detectar obstáculos no caminho do robô, o que pode ser útil até para aplicações que não realizam navegação de alto nível.

## 1.2. Trabalhos similares

A dificuldade de utilizar visão monocular para detectar obstáculos suspensos e diferenciar entre simples desenhos no chão e verdadeiros obstáculos é bem conhecida [7], mas ainda existe pouca pesquisa em formas simples de superar estas limitações.

Qualquer avanço nesta área deverá ter a forma de algoritmos que realizam análises simples nas imagens para obter dados de localização mais completos dos obstáculos detectados. Um exemplo interessante é a localização de obstáculos negativos apresentada recentemente em [5]. Este artigo mostra um sistema que, além de utilizar estereosco-

pia, analisa o movimento relativo das bordas detectadas e a oclusão e aparição destas para determinar quais delas são limites de obstáculos negativos.

O sistema apresentado aqui busca igualmente explorar simples variações no tempo dos dados analisados para determinar a estrutura do ambiente. Mas a proposta é realizar apenas uma detecção da proximidade da câmera a obstáculos locomotores, e não realizar atividades mais complexas como o mapeamento de uma região.

## 2. Metodologia

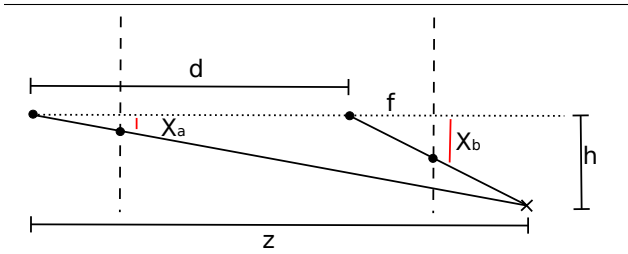
O sistema é baseado em uma câmera obscura simples, ou modelo *pinhole*. Esta câmera desloca-se na direção de seu eixo ótico e a uma velocidade constante. A única parte da imagem que é analisada é o segmento de reta vertical entre o centro de projeção e a extremidade inferior da imagem. O centro de projeção da câmera utilizada é considerado conhecido.

### 2.1. Processamento de sinais

O algoritmo é composto por dois estágios. Primeiro o de processamento de imagens, que encontra pontos correspondentes em pares de imagens. A seguir o de localização, que fornece as coordenadas no espaço de cada ponto encontrado. O primeiro estágio é composto pelos seguintes passos:

1. Imagem é filtrada por um filtro de mediana para reduzir ruídos, atenuar texturas e realçar as bordas.
2. A metade inferior da coluna do meio da imagem é extraída. Mais especificamente, foram utilizadas imagens com 480 linhas e 640 colunas, de onde se extraíram os pixels da 320ª coluna a partir da 240ª linha.
3. Calculam-se as diferenças sucessivas deste vetor em cada canal, e os picos mais proeminentes são tomados como os pontos de interesse a serem localizados.
4. Estes picos são buscados através de correlação, nos três canais, de uma janela de 7 pixels ao redor do pico em uma região de busca de 0 a 7 pixels de deslocamento para baixo na imagem seguinte.
5. O ponto de maior correlação, somando os resultados de cada canal, é considerado o ponto correspondente daquele pico na imagem seguinte se o valor da correlação for superior a um limiar, senão o ponto é descartado.

O resultado deste estágio é uma lista de pares de coordenadas que devem corresponder às projeções de um mesmo ponto no espaço nas duas imagens analisadas. A partir destes valores encontra-se a distância e altitude em que se localiza cada ponto no referencial da câmera.



**Figura 1. Variáveis envolvidas no processo de triangulação.**

## 2.2. Localização dos pontos

Dados dois pontos correspondentes em um par de imagens é possível encontrar sua localização no espaço através de uma triangulação. Se a câmera foi deslocada por uma distância  $d$ , e as coordenadas do ponto na imagem forem  $x_a$  e  $x_b$ , o cálculo irá fornecer a distância  $z$  entre o ponto e a câmera na posição inicial e a profundidade  $h$  em relação à trajetória da câmera — que também é o eixo óptico. O cálculo depende ainda da distância focal  $f$  e de outros parâmetros intrínsecos, mas eles interferem apenas na escala dos valores resultantes.

A Figura 1 demonstra a relação geométrica entre as variáveis. Para cada imagem obtida é formado um triângulo retângulo entre o eixo óptico, o ponto focal da câmera e o ponto no espaço. Um triângulo semelhante a este é formado pelos mesmos eixo óptico e ponto focal, e ainda pelo plano da imagem. O plano da imagem intercepta a reta que liga o ponto no espaço ao ponto focal. Este ponto de interseção é onde se localiza a projeção na imagem do ponto no espaço que foi percebido pela câmera.

Pela semelhança de triângulos tem-se

$$\frac{x_a}{f} = \frac{h}{z} \quad \frac{x_b}{f} = \frac{h}{z-d} \quad (1)$$

$$zx_a = hf \quad (z-d)x_b = hf \quad (2)$$

$$zx_a = (z-d)x_b \quad (3)$$

$$z = -d \frac{x_b}{(x_a - x_b)} \quad (4)$$

$$z = d \frac{x_b}{(x_b - x_a)} \quad h = \frac{zx_a}{f} = \frac{x_a x_b}{f(x_b - x_a)} \quad (5)$$

Pode-se observar nestas equações que as grandezas desejadas,  $z$  e  $h$ , são diretamente proporcionais aos fatores  $d$  e  $f^{-1}$ . Isto significa que é possível normalizar estas grandezas para permitir utilizar durante o funcionamento do sistema apenas os valores das coordenadas dos pontos nas imagens. A localização de pontos no espaço produzida desta forma pode já servir para algumas aplicações, mas se for desejável obter valores de distância em uma escala específica

basta realizar uma multiplicação dos valores produzidos pelo sistema por parâmetros obtidos de antemão.

O princípio que se pretende explorar é a restrição geométrica das coordenadas nas imagens de um ponto no espaço conforme é realizada uma movimentação na direção do eixo óptico da câmera. Esta forma de movimentação é comum na operação de alguns robôs móveis. Se ela ocorrer em velocidade constante as coordenadas nas imagens poderão ser ajustadas por fórmulas quadráticas que desenhem hipérbolas no tempo.

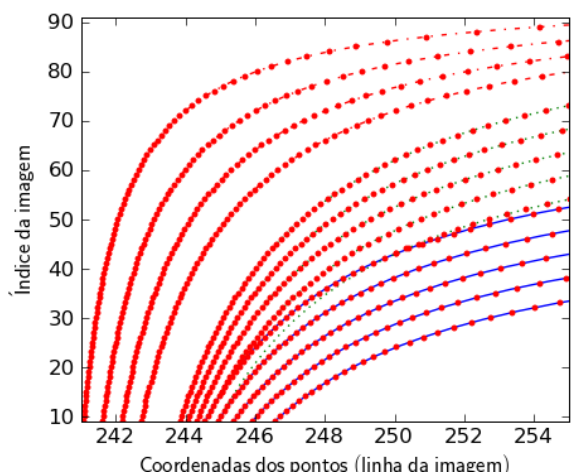
A Figura 2 apresenta um exemplo do que ocorre quando são obtidas imagens de um grupo de pontos. Existem três grupos de pontos colineares neste ambiente, dois horizontais em diferentes altitudes, e um vertical ao fundo, como uma parede. O conjunto mais próximo da parede é o menos elevado.

As colunas de cada imagem obtida correspondem às linhas na Figura 2(a). É possível distinguir os grupos de pontos porque dentro de cada grupo apenas um dos parâmetros de suas hipérbolas correspondentes é variado. É possível notar ainda a ocorrência de oclusão neste gráfico, nos pontos em que cada hipérbole relativa aos pontos mais distantes encontra o ponto que seria a borda do conjunto mais próximo de pontos.

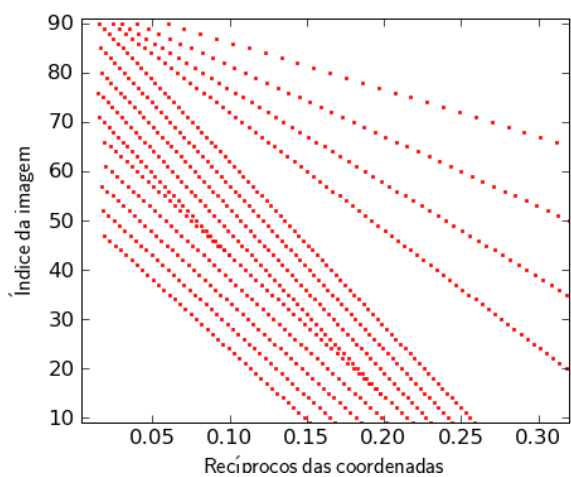
A análise destas curvas se torna mais fácil ao se realizar uma substituição de variáveis. São utilizados os valores recíprocos (inversos multiplicativos) das coordenadas, tendo o centro de projeção como referencial da imagem. Neste caso cada ponto irá corresponder a uma reta, sendo todas decrescentes. Retas mais inclinadas indicam pontos menos elevados no espaço, e quanto mais próximas da origem, mais próximas do ponto onde se iniciou a trajetória da câmera. A Figura 2(b) demonstra o gráfico correspondente à cena descrita anteriormente. É possível notar mais claramente a oclusão dos pontos mais afastados e rebaixados, que correspondem às retas intermediárias do gráfico. Pontos localizados em retas paralelas ao deslocamento da câmera correspondem a retas paralelas neste gráfico.

## 3. Análise

O sistema foi testado com dois conjuntos de imagens, o primeiro criado com um programa de computação gráfica convencional (Blender v2.49), onde a câmera foi deslocada sobre uma mesa até sua borda, eventualmente podendo observar a junção entre a parede e o chão da sala de um cenário virtual. O segundo conjunto de imagens foi obtido com uma câmera de resolução  $640 \times 480$  pixels deslocada a intervalos de 50mm montada em um carrinho sobre trilhos. A câmera se encontrava a aproximadamente 30cm do solo, e o percurso total percorrido foi de 1,6m. Este experimento ocorreu em uma plataforma com um parapeito, de onde é possível observar alguns objetos distantes no andar inferior.



(a)



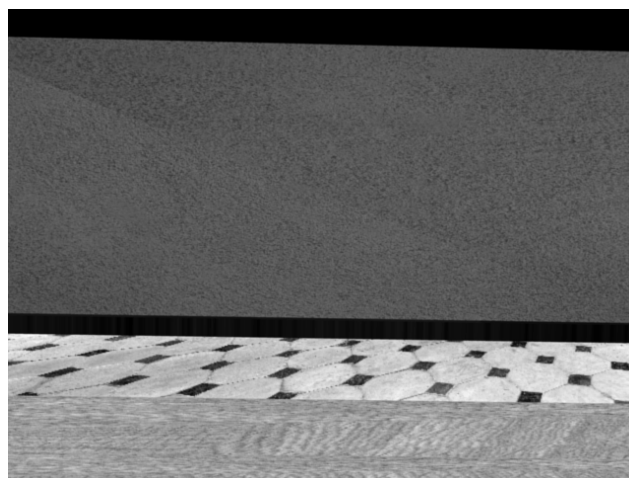
(b)

**Figura 2. Coordenadas de pontos em imagens ao longo do tempo (a), e as retas formadas pelos valores recíprocos (b).**

A Figura 3 mostra um exemplo de cada grupo de imagens estudado. Note que a entrada para o algoritmo desenvolvido é apenas a metade inferior da coluna mediana de pixels de cada imagem.

### 3.1. Simulação

O resultado da análise das imagens geradas por computação gráfica está na Figura 4. Os pares de pontos correspondentes encontrados pelo primeiro estágio estão apresentados na Figura 4(a). O eixo vertical indica o número de cada imagem na seqüência, a partir do



(a)



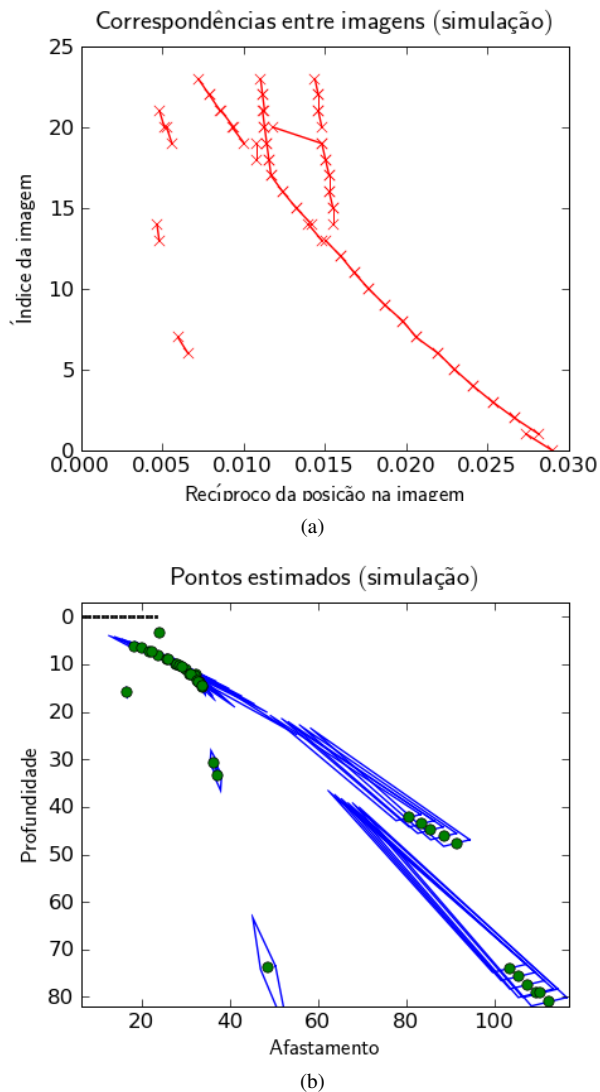
(b)

**Figura 3. Uma das imagens da simulação (a), e uma das fotografias reais (b).**

ponto 0. O eixo horizontal é o valor recíproco das coordenadas encontradas. Cada par de pontos corresponde a um segmento de reta neste gráfico, partindo de uma linha e terminando na linha seguinte, nas colunas relativas às coordenadas encontradas.

A grande curva que corta o gráfico em diagonal corresponde à borda da mesa do cenário. As outras duas curvas quase verticais que brotam dela para cima correspondem ao rodapé da parede do cenário, que só é visível a partir da décima-quinta imagem. Existem ainda alguns dados espúrios, relativos a outros objetos e a correspondências falsas.

A Figura 4(b) apresenta os dados obtidos pelo segundo estágio. Cada ponto neste gráfico corresponde a um dos segmentos de reta do gráfico anterior, e aos pares de pontos en-



**Figura 4. Correspondência de pontos em cada par de imagens da simulação (a), e as localizações obtidas (b).**

contrados no primeiro estágio. Os pontos negros no canto superior esquerdo do gráfico correspondem às coordenadas do ponto focal da câmera em cada imagem. Ao redor de cada ponto foi desenhado um quadrilátero que demonstra a sensibilidade de cada vetor em relação às coordenadas do ponto em cada imagem, que são os parâmetros de entrada do cálculo.

É possível observar três grupos distintos de pontos neste gráfico, que correspondem à borda da mesa e ao rodapé da sala. Estes grupos deveriam estar localizados sobre pontos específicos neste gráfico: os valores reais da posição de cada um deles no referencial da câmera no início de sua tra-

jetória. O espalhamento se deve a imperfeições no funcionamento do sistema.

Objetos localizados a uma distância grande se comparada ao deslocamento da câmera geram coordenadas muito próximas, ou mesmo iguais no par de imagens. Isto torna relevante o ruído introduzido pela resolução limitada da câmera e pelo uso de métodos de correlação imprecisos, e impede a medição de distâncias muito grandes. Esta incerteza também é concentrada em uma direção, o que é coerente com a distribuição sobre segmentos de reta dos pontos recuperados pela triangulação ao longo do tempo.

### 3.2. Experimento com câmera real

Os resultados da análise das imagens do experimento com uma câmera real se encontram na Figura 5. Apesar de haver muitos desvios nos dados é possível distinguir no gráfico a localização de alguns objetos do ambiente.

O grupo de 5 pontos com altura de  $-20$  a  $-30$  e afastamento 20 unidades da origem se refere a uma das duas barras transversais do trilho utilizado. O grupo de pontos distantes de 20 a 40 e altura de  $-15$  a  $-30$  se refere a outra destas barras. Outro grupo de pontos a seguir, mais esparsos, se refere à borda do fim do chão. Acima deste existe outro conjunto de pontos que vai desde o próprio eixo óptico da câmera até aproximadamente a altura  $-10$ , e posição horizontal variando de 30 a 60. Esta é uma barra horizontal do parapeito sobre a borda.

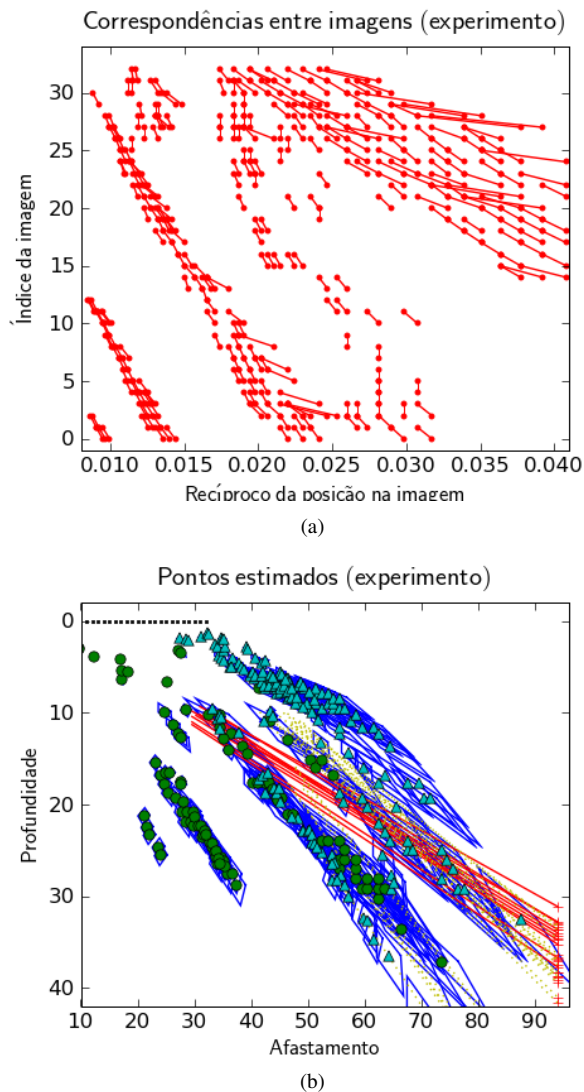
Os triângulos na Figura 5(b) provêm de leituras realizadas após a décima-quinta imagem desta série, assim como as retas tracejadas<sup>1</sup>. Estas retas à direita do gráfico indicam a direção de pontos cujos correspondentes possuem coordenadas idênticas, caso em que  $z$  diverge tornando impossível estimar a posição.

Não foi possível detectar precisamente pontos além da beirada do chão que indicassem o fim deste. Os objetos no fundo da cena criaram apenas alguns pontos espúrios, além dos pontos com disparidade muito reduzida. Entretanto, é possível notar no gráfico que as retas indicam uma região do espaço onde não ocorre a oclusão do fundo da cena pelo chão. Este fato pode ser aproveitado para determinar que aquela última borda é o fim do chão, enquanto que a área antes dela deve ser segura por ser opaca.

## 4. Conclusão

Este artigo apresentou um sistema que realiza a detecção e localização no espaço de pontos selecionados em pares de imagens obtidas em um deslocamento retilíneo. O método é

<sup>1</sup> O fato da décima-quinta imagem dividir a análise de ambas séries é apenas uma coincidência.



**Figura 5. Correspondência de pontos em cada par de imagens do experimento (a), e as localizações obtidas (b).**

direcionado para ambientes onde existem muitas bordas horizontais, ou com alta autocorrelação nesta direção. Isto permite a princípio uma análise simplificada da imagem, sobre apenas uma coluna.

O fato dos dados analisados serem unidimensionais torna a operação mais rápida do que algoritmos que utilizam toda a área de imagem. Este baixo custo operacional torna a técnica atraente para a detecção de obstáculos por robôs móveis em condições onde hoje são utilizados algoritmos que não lidam corretamente nem com obstáculos suspensos, nem negativos. O interesse nestas aplicações muitas vezes é justamente detectar bordas horizontais, o que torna ade-

quado o uso desta técnica.

É evidente, no entanto, que os dados produzidos pela técnica não são muito precisos ainda, e seria necessário utilizar uma técnica complementar como um filtro de Kalman para obter valores mais adequados da localização no ambiente dos obstáculos detectados. Desta forma os dados gerados pelo sistema poderiam ser ainda aproveitados por um outro programa operando paralelamente para realizar tarefas como a localização e mapeamento simultâneos.

Algumas pesquisas que podem seguir a apresentada neste artigo são a melhoria dos dados produzidos com o uso de melhores técnicas de processamento de imagens e de métodos probabilísticos, a realização de um controle para ajustar a velocidade do robô em conjunto com a taxa de obtenção de imagens, análises de grupos de imagens, e ainda a interoperação com algoritmos para o mapeamento tridimensional do ambiente.

## Agradecimentos

Esta pesquisa foi conduzida sob o projeto FAPESP Log-Prob (Nº 2008/03995-5) e o projeto CNPq Ob-SLAM (Nº 475690/2008-7). Anna H. R. Costa agradece ainda ao CNPq pelo financiamento Nº 305512/2008-0. Nicolau Werneck agradece à CAPES.

## Referências

- [1] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.
- [2] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering higher level structure in visual SLAM. *IEEE Transactions on Robotics*, 24(5):980–990, October 2008.
- [3] A. Howard and L. Kitchen. Fast visual mapping for mobile robot navigation. In *Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conference on*, volume 2, pages 1251–1255, Oct 1997.
- [4] S. Lenser and M. Veloso. Visual sonar: fast obstacle avoidance using monocular vision. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 886–891, Oct. 2003.
- [5] A. Murarka, M. Sridharan, and B. Kuipers. Detecting obstacles and drop-offs using stereo and motion cues for safe local motion. In *IROS*, pages 702–708. IEEE, 2008.
- [6] D. Murray and J. J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.
- [7] I. Ulrich and I. R. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, pages 866–871. AAAI Press / The MIT Press, 2000.