

Sistema de medição de distância baseado em visão computacional utilizando laser de linha

Walber Conceição de Jesus Rocha¹, Samuel Rebouças de Jesus¹
João Carlos Nunes Bittencourt¹

¹Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Recôncavo da Bahia
Av. Rui Barbosa, 710 - 44380-000 - Cruz das Almas, Brasil.

{walber-jesus,samuelreboucas07}@hotmail.com, joaocarlos@ufrb.edu.br

Abstract. *Robotic systems are in constant development, proposing automated solutions for everyday tasks. From its first generations, robotic platforms started to perform dynamic movements, in order to have autonomy in the accomplishment of several tasks. In such a context, computer vision applications have provided to robotic platforms, navigation capabilities, and mapping through image processing. The work presented herein highlights the results of the development and performance analyzes of a laser line detection system for distance measure in robotic systems. The system was deployed over the computer platform Raspberry Pi model B+.*

Resumo. *A robótica é uma área de estudo em crescente desenvolvimento, propondo soluções automatizadas para diversas tarefas do cotidiano. Com o advento da robótica móvel, plataformas robóticas passaram a executar movimentos dinâmicos, de forma a possuírem autonomia na realização de um vasto conjunto de tarefas. Nesse contexto, aplicações de visão computacional têm proporcionando à plataformas robóticas a capacidade de navegação e mapeamento através da captura e processamento de imagens. Este trabalho apresenta um sistema de detecção de laser de linha, a partir da utilização de técnicas de processamento de imagem, para determinar a distância de obstáculos e analisar o seu desempenho na plataforma computacional Raspberry Pi.*

1. Introdução

A robótica é uma área de estudo em crescente desenvolvimento que busca proporcionar ao ser humano uma vida mais confortável, propondo soluções para tarefas do seu cotidiano [Romano and Dutra 2002]. Com o advento dos robôs móveis, as plataformas robóticas passaram a executar movimentos dinâmicos, de forma a apresentarem autonomia na realização de suas tarefas [Marchi et al. 2001]. Um robô móvel pode ser definido como um autômato mecânico, estruturado sobre uma base não estática, controlada por um sistema computacional capaz de gerir sensores e atuadores, permitindo que o mesmo possa interagir com o ambiente [De Pineri 2002]. Para uma plataforma robótica ser capaz de navegar e realizar operações de mapeamento de forma dinâmica, obter informações sobre o ambiente no qual está inserido é de extrema importância para execução de suas ações. Nesse contexto, os estudos na área de Visão Computacional buscam proporcionar a uma plataforma computacional a capacidade de perceber o ambiente, compreender o

que está sendo observado, tomar ações apropriadas e aprender com essa experiência a fim de melhorar o seu desempenho no futuro [Sebe et al. 2005].

Dentre os métodos empregados para navegação e mapeamento destaca-se a identificação de obstáculos e a distância que o mesmo se apresenta frente à plataforma robótica. As medidas de distância podem ser obtidas através de sensores infravermelhos e ultrassônicos, no entanto ambos os sensores são direcionais, somente sendo capazes de identificar a presença de objetos que estejam apontados à linha de percepção do sensor [Jesus et al. 2015]. Outro fator observado está no fato de que elementos como temperatura, umidade e presença de ruídos na linha de ação dos sensores, provocam retardos nos sinais de reflexão, prejudicando as medições [Flynn 1988]. Por outro lado, sensores visuais não são direcionais, de modo que a distância para o objeto no qual o feixe de laser está presente pode ser mensurado, independente do modo como o mesmo se apresenta na cena.

Sistemas robóticos geralmente são constituídos de plataformas computacionais dotadas de baixa capacidade de processamento, tal qual os microcontroladores de baixo custo. Com o advento dos computadores de placa, a exemplo da Raspberry Pi e Beaglebone, aplicações robóticas ganharam uma nova roupagem, onde a capacidade de processamento foi aliada ao baixo custo de implementação. Todavia, aplicações de visão computacional tipicamente se caracterizam por apresentar altas demandas computacionais, uma vez que esse tipo de aplicação costuma realizar operações sobre um grande volume de dados. Nesse contexto, este trabalho apresenta um sistema de medição de distância, baseado na detecção de laser de linha a partir da utilização de técnicas de processamento de imagem. Além disso, o presente artigo apresenta uma análise de desempenho, baseado no tempo de execução e consumo de memória, de dois métodos computacionais de medição, quando executados na plataforma computacional Raspberry Pi B+.

2. Fundamentação Teórica

Em geral, métodos de extração de parâmetros em imagens utilizam de reconhecimento de bordas e detecção de linhas. Dentre esses métodos, destacam-se técnicas como Gradiente de Sobel, Transformada de Hough e Threshold. Outros métodos são baseados na detecção dos padrões apresentados nos pixels, como o método de Agregação de Pixels. As próximas seções destacam os conceitos fundamentais associados a cada um dos métodos empregados no desenvolvimento do sistema de medição.

2.1. Gradientes de Sobel

O Gradiente de Sobel é um operador de diferenciação capaz de calcular diferenças horizontais e verticais na intensidade do gradiente entre pixels de uma imagem [Selhorst 2014]. Para tanto, multiplicam-se duas matrizes de dimensão 3×3 com a imagem original I_m . Estas matrizes são chamadas de máscaras de convolução e são multiplicadas a cada pixel da imagem no sentido de descobrir a intensidade vertical e horizontal. Dessa forma, mudanças horizontais são obtidas ao multiplicar a imagem I_m com a máscara (G_x), por sua vez, variações na vertical são obtidas ao multiplicar a imagem com a máscara (G_y) [Gonzalez and Woods 2002].

2.2. Crescimento de Região por Agregação de Pixels

Este processo tem como finalidade associar novos pixels a um dado conjunto de pixels pré-selecionados, de acordo com suas características predominantes [Junior 1997]. Para tanto, um pixel precisa ser designado como semente. A partir desta semente, de forma iterativa, é realizado o crescimento das regiões, agregando cada pixel vizinho que apresente atributos similares, tais como, intensidade, textura ou cor [Saldanha and FREITAS 2009]. Este processo é ilustrado na Figura 1.

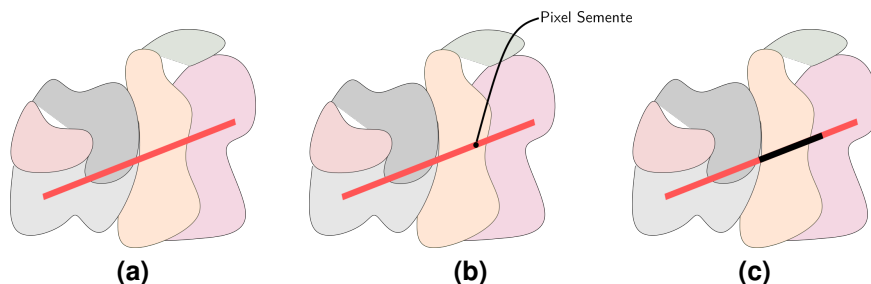


Figura 1. Aplicação do crescimento de região por agregação de pixels.

A Figura 1a apresenta uma imagem contendo uma reta vermelha. A partir da escolha do pixel sementem indicado na Figura 1b, os pixels vizinhos que possuem características semelhantes, são agregados, dando início à construção da região, conforme observado na Figura 1c. Essa região se expande por completo, destacando a tonalidade das bordas, separando-a das regiões vizinhas.

2.3. Threshold

Threshold, também conhecida como limiarização, é uma técnica aplicada como critério de seleção dos pixels que compõe uma imagem digital. O processo consiste na utilização de uma constante de brilho T , a qual é considerada como critério de seleção. Todos os pixels da imagem são comparados a este critério e então alterados conforme a necessidade [Gonzalez and Woods 2002].

Seja $F(x, y)$ uma imagem em tons de cinza e cada pixel da imagem denotado por $F(x_i, y_i)$, então:

$$F(x_i, y_i) = \begin{cases} 255, & \text{se } F(x_i, y_i) \geq T \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

De acordo com a Equação 1, quando o valor da intensidade do pixel é maior ou igual ao critério de seleção estabelecido, o pixel correspondente passa a apresentar intensidade máxima (branco), caso contrário a intensidade do pixel torna-se mínima (preto). Como resultado obtém-se uma imagem binária, realçando a região da imagem onde encontra-se o elemento que se deseja destacar.

2.4. Transformada de Hough Padrão para Retas

A transformada Hough é uma técnica que tem por objetivo detectar linhas retas e curvas em imagens digitais. De acordo com a Figura 2a, cada pixel na imagem é utilizado para

traçar uma curva senoidal, tal qual apresentada na Figura 2b. Cada uma das curvas possui um acumulador. Deste modo, se dois pontos distintos se cruzam no plano (ρ, θ) , significa que ambos os pontos pertencem à uma mesma linha, e então há um incremento no acumulador [Guedes 2014].

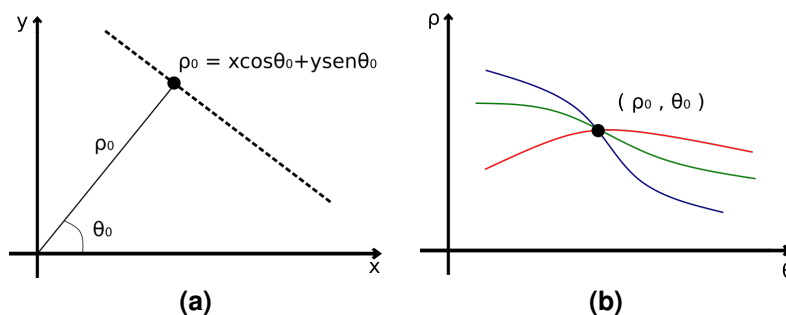


Figura 2. Transformada de Hough em coordenadas polares.

A Transformada Probabilística de Hough [Leavers 1993] utiliza o mesmo algoritmo da transformada padrão, buscando reduzir a quantidade de ruídos. Além disso, esse método é capaz de reduzir significativamente o tempo de execução em comparação à transformada clássica [Leavers 1993]. Esse algoritmo introduz melhorias nos estágios de cômputo e acumulação durante o processamento, uma vez que, ao invés de utilizar todos os pixels de borda, utiliza apenas uma fração deles.

2.5. Considerações finais

A escolha de tais técnicas fundamentou-se nas características apresentadas por cada método. No método do Gradiente de Sobel, busca-se uma reta a partir da intensidade que os pixels apresentam, quando convertido para escala de cinza, enquanto no método de Agregação de Pixels, a partir do sistema de cor que cada pixel apresenta. A extração dos pixels consiste na aplicação do Threshold simples. A Transformada Probabilística de Hough propõe a filtragem dos ruídos oriundos da extração.

3. Arquitetura de hardware e software

A plataforma de hardware do sistema proposto é apresentada na Figura 3a. Tal plataforma é dotada de uma placa Raspberry Pi B+, uma câmera de 5 MP, um emissor laser de linha vermelho de 5W e um sensor de luminosidade. O sistema foi desenvolvido em linguagem Python (versão 2.7.15), utilizando como base para as aplicações de processamento de imagem a biblioteca OpenCV. A plataforma de software foi analisada no sistema operacional GNU/Linux Raspbian (kernel versão 4.14.62).

O funcionamento geral do sistema, assim como a sequência em que são executados é apresentado no fluxograma da Figura 3b. Inicialmente é verificado o funcionamento do emissor laser de linha, uma vez satisfeita, um novo quadro é capturado, seguido da execução das etapas: (i) pré-processamento, composto por operações de otimização na imagem capturada; (ii) detecção e extração da linha, onde são aplicados os métodos Gradiente de Sobel e Agregação de Pixels; (iii) filtragem da linha, utilizando a Transformada Probabilística de Hough; e (iv) cálculo e determinação da distância.

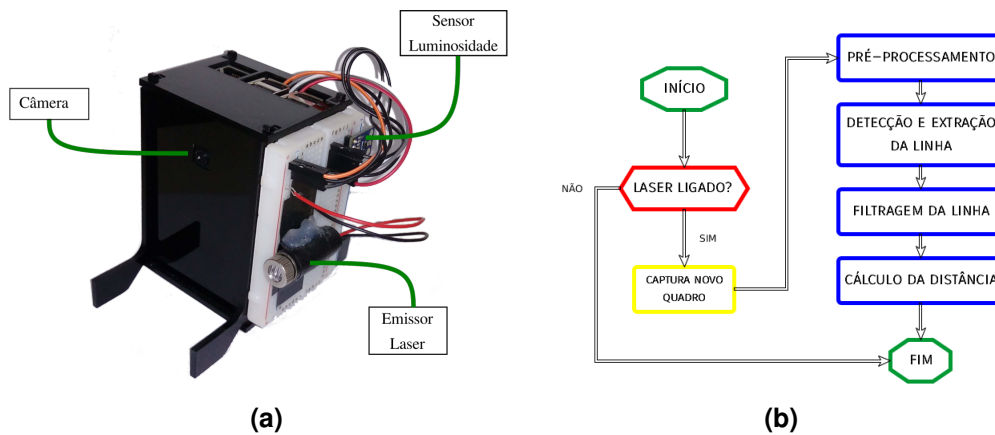


Figura 3. Arquitetura de hardware e software.

3.1. Pré-processamento

A finalidade das operações de pré-processamento é obter uma imagem com dimensões reduzidas e com níveis de cores balanceados. Uma vez que o laser apresenta altura fixa em relação ao plano focal da câmera, realiza-se um corte na altura da imagem capturada, a fim de reduzir sua dimensão sem perder nenhuma informação do laser.

Tendo em vista que as imagens podem ser adquiridas sob diferentes condições de iluminação, o sistema implementa a normalização da imagem capturada. Em seguida, um filtro de desfocagem gaussiana é aplicado, de forma que os pixels apresentem uniformidade nas cores. Este processo facilita a determinação do feixe de laser na imagem. O resultado desta etapa de pré-processamento é uma imagem com dimensões reduzidas e paleta de cores ajustada, conforme as Figuras 4a e 4b.

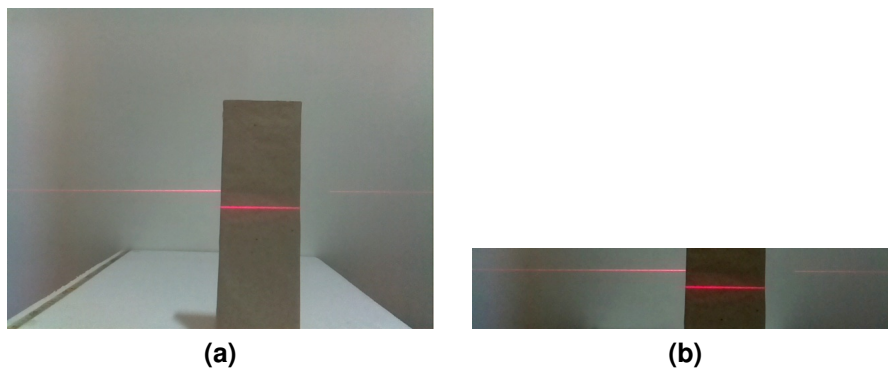


Figura 4. Pré-processamento da imagem de entrada.

3.2. Detecção e Extração da Linha

Dois mecanismos foram utilizados para detecção da linha: (i) Gradientes de Sobel e (ii) Agregação de Pixels. Para o Gradiente de Sobel, inicialmente é efetuada uma transformação na escala de cores da imagem, transformando-a para tons de cinza. Em seguida, calcula-se as diferenças na intensidade horizontal dos pixels, resultando no realce da linha do laser. A partir do resultado oriundo do gradiente de Sobel, a função

Threshold, realiza a análise, pixel a pixel da imagem, caracterizando-a conforme o limiar de brilho estabelecido. Os resultados desta etapa de processamento são ilustrados na Figura 5.



Figura 5. Resultados oriundos da etapa de detecção e extração da linha. (a) Resultado do Gradiente de Sobel sobre uma imagem de teste. (b) Resultado do Threshold.

No método de Agregação de Pixels, a primeira etapa consiste na transformação da imagem para o sistema de cores Hue, Saturation e Value (HSV), tendo em vista que este sistema apresenta maior abrangência sobre as cores do espectro emitido pelo laser. Para realizar a agregação dos pixels pertencentes ao feixe de laser, utilizou-se a função `inRange`, disponível na biblioteca OpenCV, responsável por localizar e agregar os pixels pertencentes ao intervalo de cores especificado.

A região agregada passa pelo processo de abertura morfológica. Tal técnica combina (i) erosão, rotina que efetua a remoção dos pixels em determinado raio; e (ii) dilatação, responsável pelo acréscimo de possíveis segmento de linha remanescente do processo de erosão. Por fim, o feixe de laser é destacado através da função `bitwiseAnd`, disponível na biblioteca OpenCV. Tal função recebe como parâmetro uma imagem de entrada e a região na qual se encontra o feixe de laser. Como resultado, todos os pixels que pertencem à região são realçados, conforme apresentado na Figura 6. Assim como no método do Gradiente de Sobel, a caracterização dos pixels ocorre a partir da função `Threshold`.

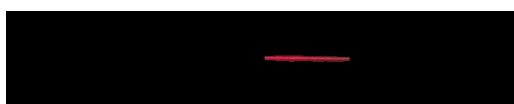


Figura 6. Resultado do processo de Agregação de Pixels.

3.3. Filtragem da linha

Frequentemente, pixels não pertencentes a linha do laser estão presentes nos resultados oriundos do Gradiente de Sobel e Agregação de Pixels. O surgimento de tais ruídos está relacionado às superfícies de reflexão dos objetos em questão. A filtragem da linha é realizada a partir da função `HoughLinesP`, disponível na biblioteca OpenCV.

Essa rotina é responsável por encontrar os segmentos de linha utilizando o método da Transformada Probabilística de Hough. Em seguida, os pixels referentes a altura da linha do laser são armazenados no vetor `alturas`. Uma vez que a distância até o objeto é calculada a partir da altura máxima em que a projeção da linha do laser se apresenta na imagem, o vetor `alturas` é percorrido a fim de encontrar a altura máxima, a qual corresponde ao ponto do feixe de laser que é refletido no objeto mais próximo à câmera.

3.4. Cálculo da Distância

O cálculo da medida de distância até um objeto que se encontra no campo de visão da câmera foi determinado com base no modelo apresentada na Figura 7. Em função da estrutura da plataforma, apresentada na Figura 3a, o emissor laser de linha foi posicionado ao lado da câmera, a uma distância vertical de 2,8 cm. Na Figura 7, a distância do centro da imagem até a posição em que a linha do laser é projetada no objeto é definida como a Distância ao Plano Focal (DPF), e θ representa a medida do ângulo de visão (em radianos) formado pela projeção da linha do laser na imagem.

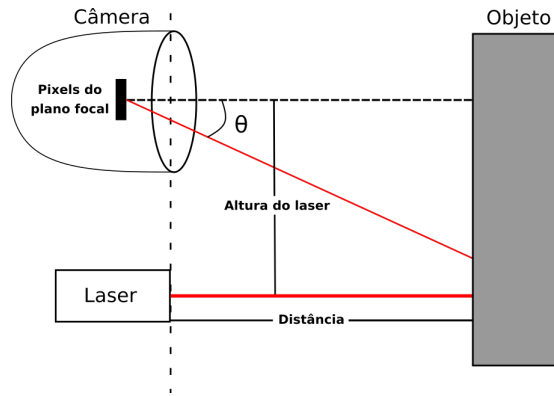


Figura 7. Modelagem para obtenção da distância ao objeto.

A relação para o cálculo de determinação da distância foi descrita a partir da análise trigonométrica do sistema, tal como apresenta a Equação 2. A expansão da relação da apresentada na Equação 2 é utilizada para converter cada pixel da imagem, em seu valor correspondente medido em centímetros. Para tanto, determina-se a expressão que relaciona a quantidade de ângulo por pixels da imagem, também chamada de radianos por pixel (RPP), e a DPF, equivalente a altura máxima da linha projetada sobre o objeto. Em função da existência de deformações na lente da câmera, considera-se o parâmetro que fornece a correção angular, denotado por Deslocamento Radiano (DR).

$$dist = \frac{alturaLaser}{\tan(\theta)}; \quad \theta = (DPF \times RPP) + DR \quad (2)$$

4. Resultados

Para validar a aplicação proposta pelo sistema apresentado na Seção 3, foram realizados testes de precisão, cujo intuito foi estimar um intervalo de confiança para as medições realizadas a partir do sistema desenvolvido. Além disso, foram realizados testes do tempo de execução e consumo de memória, com o objetivo de analisar o impacto da implementação do sistema em plataformas computacionais de baixo custo. Tais testes de desempenho foram realizados em uma placa Raspberry Pi modelo B+.

O sistema foi avaliado sob a perspectiva dos métodos: (i) Gradiente de Sobel; e (ii) Agregação de Pixels. Inicialmente, a distância para o objeto foi aferida usando uma fita métrica e então comparada com o resultado obtido com o sistema proposto. Durante a realização dos testes constatou-se que para distâncias inferiores à 20 cm, o sistema torna-se incapaz de estabilizar o foco na linha do laser. Por outro lado, para distâncias superiores

à 100 cm, a projeção da linha não é perceptível para a câmera devido à sua relativa baixa intensidade luminosa. Tomando por base as restrições do protótipo desenvolvido, a faixa de variação das medições foi estabelecida entre as distâncias 20 cm e 100 cm. Os testes consistiram na captura de cinco imagens correspondentes a cada medida de distância, durante medições sucessivas, em passos de 10 cm. A Tabela 1 apresenta os resultados de precisão obtidos durante a análise da execução do sistema usando os métodos Gradiente de Sobel e Agregação de Pixels. Todos os valores apresentados na Tabela 1 são descritas em medidas de centímetros.

Tabela 1. Resultados de medida de distância.

Distância	Gradiente de Sobel			Agregação de Pixels		
	Média	ϵ	σ	Média	ϵ	σ
20 cm	20,216	0,216	0,679	20,992	0,992	1,562
30 cm	30,428	0,428	0,893	30,460	0,460	1,745
40 cm	40,494	0,494	0,973	40,678	0,678	1,092
50 cm	50,506	0,506	0,819	51,828	1,828	2,297
60 cm	60,52	0,520	1,284	62,074	2,074	1,208
70 cm	69,362	0,638	1,306	68,686	1,314	0,877
80 cm	80,676	0,676	2,140	80,726	0,726	2,224
90 cm	91,258	1,258	1,212	90,676	0,676	1,681
100 cm	101,38	1,380	1,658	99,112	0,888	3,381

Com os resultados oriundos do Gradiente de Sobel Tabela 1, observa-se que o erro médio das medidas apuradas apresenta um comportamento crescente. Dessa forma, ao aumentar-se a distância do objeto à câmera, amplia-se o erro apresentado pelo sistema. Com relação ao método de Agregação de Pixels, no intervalo de 20 cm até aproximadamente 70 cm, os erros também se apresentam de forma crescente. Os erros apresentados pelas demais medidas são resultados das características do conjunto câmera/laser, tal que, a baixa percepção do feixe de laser interfere na operação do sistema.

Tabela 2. Comparação com métodos presentes na literatura.

Trabalho	Método	Erro médio (cm)
[Zhang et al. 2013]	Filtro Chebyshev	0,143
[Zhang et al. 2013]	Série polinomial de Fourier	0,411
[Jesus et al. 2015]	Binarização da imagem	0,747
[Hamad et al. 2014]	Segmentação por cores	2,642
Nosso	Gradiente de Sobel	0,681
Nosso	Agregação de Pixels	1,071

Com a finalidade de estipular a taxa de precisão dos métodos propostos, foi realizada uma análise comparativa com os trabalhos presentes na literatura, orientados à medição de distância utilizando emissores lasers. Os resultados oriundos desta análise são apresentados na Tabela 2. Em [Zhang et al. 2013] propõe-se a utilização da série polinomial de Fourier e do filtro Chebyshev, aplicados como ajustes da curva de detecção

do emissor laser de ponto. O erro médio foi mensurado para a faixa de 29 cm à 90 cm, alcançando erro médio de 0,143 cm e 0,411 cm, respectivamente. O método proposto em [Jesus et al. 2015] utiliza um emissor laser de ponto, utilizando técnicas de binarização para detecção do ponteiro laser. A faixa de detecção do erro médio ocorreu no intervalo de 20 cm à 70 cm, atingindo erro médio de 0,747 cm. O trabalho apresentado por [Hamad et al. 2014] introduz a metodologia de busca do emissor laser de ponto através do espectro de cor emitido pelo mesmo. A faixa de detecção avaliada consistiu no intervalo de 29 cm até 100 cm, alcançando erro médio de 2,642 cm.

Para cada uma das imagens dentro do intervalo, foram executadas as rotinas do módulo Line Profiler [Kern 2017] e Memory Profiler [Pedregosa 2018]. Para cada iteração da rotina, foram geradas folhas de dados contendo informações acerca do tempo de execução para cada um dos métodos analisados. De posse desses dados, o tempo médio de execução, bem como o consumo médio de memória foram apurados, conforme descrito na Tabela 3.

Tabela 3. Tempo de execução e consumo de memória do Gradiente de Sobel e Agregação de Pixels.

Distância	Tempo médio de execução (ms)		Consumo médio de memória (MB)	
	Gradientes de Sobel	Agregação de Pixels	Gradientes de Sobel	Agregação de Pixels
20 cm	152,84	237,44	0,39	0,25
30 cm	155,34	241,55	0,40	0,25
40 cm	157,20	239,03	0,41	0,24
50 cm	154,43	234,98	0,40	0,25
60 cm	155,80	234,53	0,40	0,25
70 cm	155,00	233,25	0,39	0,24
80 cm	157,53	231,40	0,39	0,24
90 cm	157,06	231,09	0,40	0,24
100 cm	154,50	228,35	0,40	0,25

A partir da análise da Tabela 3, observa-se que o método do Gradiente de Sobel possui um tempo de execução menor devido a sua característica de execução. No que se refere ao tempo de execução, os métodos analisados neste trabalho apresentam resultados favoráveis ao uso em plataformas robóticas. Percebe-se que não há uma distinção evidente entre o consumo de memória por parte dos métodos, com variação máxima de até 160 KB.

5. Conclusão

Com base nos resultados obtidos, observa-se que ambos os métodos apresentam uma precisão notável na faixa de operação do sistema. O uso do emissor laser de linha introduz um aspecto positivo à detecção, tal que, diferente do emissor laser de ponto, não necessita que os objetos estejam dispostos na sua linha de ação. O contraste da acurácia entre os resultados obtidos e os trabalhos presentes na literatura é aproximadamente 0,58 cm, ratificando a precisão dos métodos propostos. O sistema ainda é capaz de fornecer as

informações com tempo de resposta aproximadamente de 153 ms. Em condições normais, o método do Gradiente de Sobel apresenta tempo de resposta cerca de 68 % mais rápido quando comparado ao método de Agregação de Pixels. O hardware nativo da Raspberry Pi B+ se mostrou adequado para o processamento das imagens capturadas utilizando os módulos disponíveis na biblioteca de visão computacional OpenCV. Com base nas análises de desempenho efetuadas, este sensor mostra-se adequado para aplicações em sistemas embarcados, devido a sua simplicidade, velocidade e baixo consumo.

Referências

- De Pireri, E. R. (2002). Curso de robótica móvel. Santa Catarina, Florianópolis.
- Flynn, A. M. (1988). Combining sonar and infrared sensors for mobile robot navigation. *The International Journal of Robotics Research*, 7(6):5–14.
- Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing*, volume 455. Beijing: Publishing House of Electronics Industry, 2 edition.
- Guedes, L. S. (2014). Estudo e aplicação de métodos de segmentação de imagens para o diagnóstico de falhas na fabricação de equipos.
- Hamad, A. A., Dhahir, D. A., Mhdi, B. R., and Salim, W. H. (2014). Laser distance measurement by using web camera. *Laser*, 4(04).
- Jesus, F. B., Joel Filho, P. C., Bittencourt, J. C. N., and Jesus, T. C. (2015). Modeling and simulation of laser rangefinder architecture. *system*.
- Junior, R. H. (1997). *Segmentação de imagens por morfologia matemática*. PhD thesis, Instituto de Matemática e Estatística da Universidade de São Paulo, 07.03. 97.
- Kern, R. (2017). Line profiler.
- Leavers, V. (1993). Which hough transform? *CVGIP: Image understanding*, 58(2):250–264.
- Marchi, J. et al. (2001). Navegação de robôs móveis autônomos: estudo implementação de abordagens.
- Pedregosa, F. (2018). Memory profiler.
- Romano, V. and Dutra, M. (2002). Introdução a robótica industrial. *Robótica Industrial: Aplicação na Indústria de Manufatura e de Processo*, São Paulo: Edgard Blucher, pages 1–19.
- Saldanha, M. F. and FREITAS, C. (2009). Segmentação de imagens digitais: Uma revisão. *Divisão de Processamento de Imagens-Instituto Nacional de Pesquisas Espaciais (INPE), São Paulo*.
- Sebe, N., Cohen, I., Garg, A., and Huang, T. S. (2005). *Machine Learning in Computer Vision*, volume 29. Springer Science Business Media.
- Selhorst, S. (2014). Utilização da visão computacional e detecção de características para auxiliar na navegação de pessoas com deficiência visual em ambientes internos.
- Zhang, X., Yang, Y., Liu, Z., and Zhang, J. (2013). An improved sensor framework of mono-cam based laser rangefinder. *Sensors and Actuators A: Physical*, 201:114–126.