

Identificação e Localização de Objetos em Navegação Robótica Utilizando Cores e Estereoscopia

Rafael N. Pacheco¹, Júlio M. Kawai¹, Anna H. R. Costa¹

¹Laboratório de Técnicas Inteligentes – Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158, tv. 3 – 05508-900 São Paulo, SP

{rafael.pacheco, julio.kawai, anna.reali}@poli.usp.br

Abstract. *This paper presents a computer vision system capable of identifying and locating obstacles and targets in dynamic environments, through the use of color based region segmentation and stereoscopic vision. This system was developed in order to extend the sensorial capabilities of an architecture named REACT. With the introduction of the computer vision system, the robot is capable of following moving targets and detecting obstacles which were undetectable when only sonars were used. A comparison between the sonar's and the developed system's precision is presented.*

Resumo. *Este trabalho apresenta um sistema de visão computacional capaz de identificar e localizar obstáculos e alvos em ambientes dinâmicos, através do uso de segmentação de regiões baseada em cores e estereoscopia. Este sistema foi desenvolvido no intuito de estender a capacidade sensorial de uma arquitetura reativa para navegação robótica denominada REACT. Com a introdução do sistema de visão, o robô é capaz de seguir alvos móveis e detectar obstáculos antes não detectáveis pelos sonares. Uma comparação entre a precisão do sistema desenvolvido e a dos sonares é apresentada.*

1. Introdução

A detecção de obstáculos e de alvos em ambientes dinâmicos e imprevisíveis são tarefas fundamentais para a navegação em robótica móvel. Estas tarefas dividem-se em duas partes fundamentais: a identificação do objeto de interesse (seja um alvo ou um obstáculo), e sua localização no ambiente, determinando sua posição em relação ao robô. Sonares são frequentemente utilizados por robôs para localizar objetos no ambiente, porém estes possuem extremas limitações para permitir a identificação dos objetos, impossibilitando a distinção entre alvos e obstáculos. Deste modo, os sonares normalmente são utilizados apenas para localizar obstáculos, sendo os alvos fornecidos por agentes externos ou pré-definidos através de um mapa do ambiente.

Este trabalho apresenta um sistema de visão computacional capaz de identificar e localizar obstáculos e alvos em ambientes dinâmicos, através do uso de segmentação de regiões baseada em cores e estereoscopia. Este sistema foi desenvolvido no intuito de estender a capacidade sensorial de uma arquitetura reativa denominada REACT, desenvolvida previamente pelos autores [Pacheco and Costa, 2002], e descrita resumidamente na seção 2. Nas seções 3 e 4 são apresentados os fundamentos teóricos das duas técnicas

utilizadas – segmentação baseada em cores e estereoscopia – e como elas lidam com os problemas da identificação e da localização, respectivamente, dos objetos de interesse no ambiente. Os resultados da localização dos objetos pelo sistema desenvolvidos são então comparados com os fornecidos pelos sonares do robô *Pioneer 2DX*, da *ActivMedia Robotics*, na seção 5. Finalmente, a seção 6 apresenta as discussões e conclusões do trabalho.

2. A Arquitetura REACT

A arquitetura REACT [Pacheco and Costa, 2002] é uma arquitetura reativa composta por três comportamentos primitivos: *avoidCollision*, *moveAhead* e *moveToGoal*. O comportamento *avoidCollision* é responsável pelo desvio dos obstáculos; o comportamento *moveAhead* minimiza as oscilações do comportamento *avoidCollision*, dotando o robô de um certo tipo de inércia; e o comportamento *moveToGoal* atrai o robô para o alvo. Os comportamentos são modelados por MotorSchemas e coordenados dentro do paradigma cooperativo [Arkin, 1999]. O comportamento global resultante permite ao robô navegar por entre obstáculos em direção a um alvo fixo fornecido por um agente externo.

O sistema de visão computacional apresentado neste trabalho tem por objetivo permitir que o alvo do robô seja detectado dinamicamente, além de viabilizar o desvio de obstáculos não detectáveis pelos sonares, como quinas, pés de cadeiras etc. Com a introdução do sistema de visão, a arquitetura REACT é modificada de modo que este sistema passa a alimentar os módulos de percepção dos comportamentos *avoidCollision* e *moveToGoal*, como pode ser visto na Figura 1.

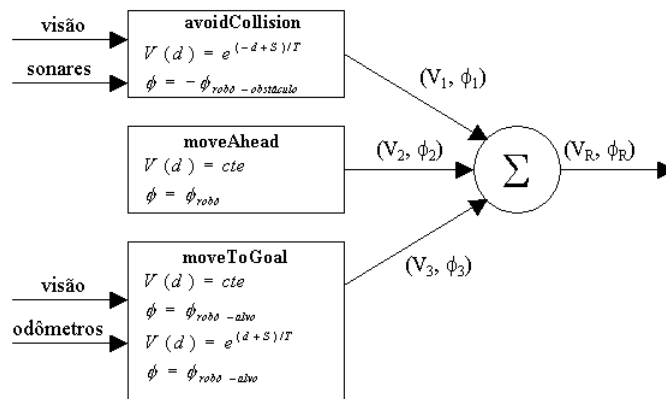


Figura 1: A Arquitetura REACT.

3. Identificação Visual dos Alvos e Obstáculos

Um dos atributos da imagem de um objeto muito utilizado para a sua identificação é a sua cor, ou a cor de marcadores posicionados no mesmo para este fim, uma vez que os algoritmos para a segmentação de regiões baseada em cores são relativamente simples, permitindo resposta em tempo real do robô.

Na segmentação de imagens, a propriedade cor é explorada para efetuar a classificação de pixels como pertencentes ou não a uma determinada classe de cor, para

posteriormente efetuar procedimentos de conexão de pixels adjacentes que possuam as mesmas cores. Isso porque objetos na cena normalmente estão relacionados a pixels adjacentes, que compartilham propriedades similares.

Neste trabalho foi utilizada uma biblioteca de funções que realiza a segmentação das regiões com base em suas cores, denominada CMVision (*Color Machine Vision*) [Bruce et al., 2000]. Um programa foi desenvolvido em JAVA para possibilitar a captura das imagens e o controle dos limiares. A integração deste programa com o CMVision – escrito em C++ – foi feita através da API JNI (Java Native Interface) do JAVA.

Os resultados da segmentação baseada em cores são os centróides das regiões encontradas na imagem, acompanhados dos identificadores das cores detectadas e da quantidade de pixels presentes em cada região, além das coordenadas do menor retângulo envolvente que inclua a região (*Bounding Box*). Estas informações permitem determinar se a região é um obstáculo ou um alvo (com base na cor), se não é ruído (regiões com pequeno número de pixels, em relação às dimensões do objeto de interesse), e qual é o ponto de referência do objeto na imagem (centróide).

4. Localização Visual dos Alvos e Obstáculos

Além de identificar o alvo e os obstáculos, é necessário localizá-los no ambiente, ou seja, determinar quais as suas posições em relação ao robô, para que este possa reagir adequadamente a eles.

Uma técnica freqüentemente utilizada para recuperar, a partir de imagens, a informação tridimensional de uma cena – especialmente quando há a necessidade de precisão – é a estereoscopia. Esta técnica permite inferir a profundidade de um ponto de interesse no ambiente a partir de duas ou mais imagens deste ponto, capturadas a partir de diferentes pontos de observação, através da triangulação entre as duas câmeras e o ponto [Faugeras, 1987].

Para realizar a triangulação, é preciso conhecer as características geométricas internas das câmeras utilizadas, bem como a relação entre a localização e orientação das câmeras que capturaram as imagens. A este conjunto de parâmetros é dado o nome de *modelo geométrico da câmera*. As características geométricas (físicas) internas da câmera são chamadas de *parâmetros intrínsecos*, e a relação geométrica entre as posições e orientações de captura são chamados de *parâmetros extrínsecos*. O processo de obtenção destes parâmetros é conhecido como *calibração de câmeras*.

4.1. Calibração de Câmeras: modelos e métodos

O modelo geométrico que se deseja obter com a calibração é derivado do modelo *pin-hole* geralmente adotado para as câmeras [Faugeras, 1987]. Neste modelo, o processo de formação da imagem na câmera é aproximado por uma projeção em perspectiva, onde as coordenadas de um ponto do ambiente e a sua projeção no plano de imagem são relacionadas pela seguinte equação:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (1)$$

onde x, y são as coordenadas do ponto(pixel) no plano de imagem, X, Y, Z são as coordenadas do ponto no ambiente, R e T são a matriz de rotação e o vetor de translação, respectivamente, entre o sistema de coordenadas do ambiente e o sistema de coordenadas da câmera, cuja origem coincide com o centro de projeção da mesma, e s é um fator de escala. A matriz A , conhecida como matriz intrínseca da câmera, é dada por:

$$A = \begin{bmatrix} \alpha_u & \theta_c \cdot \alpha_u & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

onde α_u e α_v são os fatores de escala horizontal e vertical, respectivamente, u_0 e v_0 são as coordenadas do ponto principal da imagem e o θ_c codifica o ângulo formado pelos eixos da matriz CCD da câmera.

No entanto, o modelo *pinhole* é apenas uma aproximação da formação da imagem na câmera. Quando precisão é um requisito, um modelo mais fiel deve ser utilizado. Normalmente, o modelo *pinhole* é estendido para considerar a distorção causada pelas lentes. As principais correções utilizadas são para as componentes radial e a tangencial da distorção, através da adição de alguns coeficientes ao equacionamento exposto acima.

Com a introdução da distorção, o mapeamento entre as coordenadas do ponto no ambiente e no plano da imagem deixa de ser linear e pode ser descrito como segue.

Seja $[XYZ]^T$ um ponto do ambiente no sistema de coordenadas da câmera (obtido aplicando-se a rotação R e a translação T sobre o ponto no sistema de coordenadas do ambiente). $[X/Z \ Y/Z]^T = [x \ y]^T$ são as coordenadas normalizadas (projeção *pinhole*) no plano da imagem. Ao se considerar a distorção, as novas coordenadas normalizadas serão:

$$x' = (1 + kc(1).r^2 + kc(2).r^4 + kc(5).r^6).x + 2.kc(3).x.y + kc(4).(r^2 + 2.x^2), \quad (2)$$

$$y' = (1 + kc(1).r^2 + kc(2).r^4 + kc(5).r^6).y + kc(3).(r^2 + 2.y^2) + 2.kc(4).x.y, \quad (3)$$

onde $r^2 = x^2 + y^2$ e kc é o vetor contendo os coeficientes de distorção radial e tangencial. As coordenadas em pixels seriam então obtidas por:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}. \quad (4)$$

Diversos métodos para obter este conjunto de parâmetros podem ser encontrados na literatura [Zhang, 1999, Heikkilä and Silvén, 1997, Tsai, 1987, Faugeras, 1987]. Neste trabalho foi utilizado o método descrito em [Zhang, 1999]. Este método foi escolhido por considerar a distorção causada pelas lentes e por dispensar a utilização de arranjos ou procedimentos experimentais complexos.

Através do uso deste método, é possível realizar a calibração a partir da observação, pela câmera, de um gabarito de calibração plano a partir de diversos (pelo menos dois) pontos de observação diferentes. O gabarito utilizado pode ser visto na Figura 2.



Figura 2: O gabarito utilizado para a calibração do arranjo estéreo binocular, composto por 12 quadrados na vertical e 16 na horizontal, cada um com 6,5cm de lado.

A única informação que deve ser conhecida com precisão é o tamanho das arestas dos quadrados que compõem o gabarito e o número de quadrados nas direções horizontal e vertical.

Neste método, o conjunto de parâmetros é obtido através da minimização iterativa da distância entre as coordenadas dos pontos do gabarito nas imagens utilizadas na calibração e as coordenadas dos pontos reprojetados nestas mesmas imagens, segundo um conjunto de valores iniciais para os parâmetros, geralmente obtido pela solução analítica de um sistema linear derivado da equação 1, solução que somente é possível se os coeficientes de distorção não forem considerados. Uma implementação deste método, disponível na forma de um *ToolBox* para o MatLab [Bouguet,], foi utilizado neste trabalho.

Os métodos acima referenciados dizem respeito apenas à calibração de uma câmera. No caso da aplicação destes métodos para a calibração de um arranjo estéreo binocular, cada câmera deve ser calibrada separadamente. Isto tem um impacto negativo sobre a precisão da calibração do arranjo estéreo como um todo. Por este motivo, a calibração do arranjo estéreo deve ser feita em duas etapas. Na primeira, as duas câmeras são calibradas separadamente. Em seguida, é realizado um refinamento dos modelos obtidos na primeira etapa através da obtenção dos conjuntos de parâmetros das duas câmeras simultaneamente, como se fossem apenas um. Este procedimento melhora significativamente a precisão do modelo obtido para as câmeras. Esta etapa de refinamento também se encontra implementada no *ToolBox* utilizado.

4.2. Reconstrução Estéreo 3D

Os modelos geométricos obtidos para as câmeras de um arranjo estéreo binocular fazem o mapeamento entre as coordenadas de um ponto no ambiente e as suas coordenadas (em pixels) na imagem. Devido à projeção perspectiva, diversos pontos no ambiente podem ser mapeados em um mesmo ponto no plano de imagem. Portanto, o mapeamento inverso – do plano de imagem para o ambiente – funciona da mesma maneira, sendo um ponto do plano de imagem inversamente mapeado para o ambiente como uma reta (a reta

formada pelo ponto no plano de imagem e o centro de projeção da câmera, ou o *pinhole*) [Faugeras, 1987]. A reconstrução tridimensional através da estereoscopia é baseada neste princípio: o mapeamento inverso das projeções de um mesmo ponto do ambiente nos planos de imagens de duas câmeras resulta em duas retas que se cruzam no ponto do ambiente que gerou aquelas projeções.

Se o modelo linear derivado da equação 1 estiver sendo utilizado, isto é, se a compensação da distorção dada pelas equações 2 e 3 não estiver sendo considerada, existe uma solução analítica para o problema do mapeamento inverso [Faugeras, 1987]. Caso contrário, se o modelo utilizado for o derivado da equação 4, o modelo deixa de ser linear e não existe solução analítica para o problema. Neste caso, soluções iterativas foram propostas [Heikkilä and Silvén, 1997]. Uma destas soluções encontra-se implementada no Toolbox do MatLab [Bouguet,], na forma da função *normalize.m*. Esta função, que recebe como parâmetro as coordenadas do ponto na imagem e os parâmetros intrínsecos da câmera utilizada, retorna as coordenadas normalizadas do ponto no plano de imagem, já citadas anteriormente e designadas por $[X/Z \ Y/Z]^T$. Estas coordenadas, como foi dito, definem uma reta no sistema de coordenadas da câmera.

No entanto, uma importante parte do processo da reconstrução tridimensional não se encontra implementada no toolbox utilizado. Esta parte, uma das contribuições deste trabalho, precisou ser formalizada e implementada como segue abaixo.

As coordenadas normalizadas do ponto nos planos de imagem encontram-se em sistemas de coordenadas diferentes (cada câmera tem o seu próprio sistema). Para que a intersecção das retas definidas por estas coordenadas possa ser computada, é necessário obter as mesmas em um mesmo sistema de coordenadas. No caso do arranjo estéreo, para cada câmera foi obtido, através da calibração, o conjunto de parâmetros extrínsecos correspondentes, mais especificamente a matriz de rotação R e o vetor de translação T . De posse destes dados a recuperação do ponto no ambiente dá-se da seguinte maneira: seja $[X_1/Z_1 \ Y_1/Z_1]^T = [a_1 \ b_1]^T$ e $[X_2/Z_2 \ Y_2/Z_2]^T = [a_2 \ b_2]^T$ as coordenadas normalizadas do ponto de interesse nas câmeras 1 e 2, respectivamente. Seja R_1, T_1, R_2 e T_2 as matrizes de rotação e os vetores de translação destas câmeras. $R = R_1/R_2$ e $T = T_1 - T_2$ são a matriz de rotação e o vetor de translação do sistema de coordenadas da câmera 2 com relação à câmera 1, de maneira que:

$$\begin{bmatrix} X_2/Z_2 \\ Y_2/Z_2 \end{bmatrix} = R \begin{bmatrix} X_1/Z_1 \\ Y_1/Z_1 \end{bmatrix} + T.$$

Se

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{e} \quad T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix},$$

Pode-se provar que [Costa, 2003] :

$$Z_1 = T_x - a_2 \cdot T_z / (a_1 \cdot a_2 \cdot r_{31} + a_2 \cdot b_1 \cdot r_{32} + a_2 \cdot r_{33} - (a_1 \cdot r_{11} + b_1 \cdot r_{12} + r_{13})).$$

Resolvido Z_1 , têm-se que $X_1 = a_1 \cdot Z_1$ e $Y_1 = b_1 \cdot Z_1$.

Obtêm-se, assim, as coordenadas do ponto no sistema de coordenadas da câmera

1. Para se obter as coordenadas no sistema de coordenadas do ambiente, designada aqui por $[X_w Y_w Z_w]^T$, basta aplicar a transformação abaixo, finalizando a reconstrução tridimensional do ponto:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R_1^{-1} \left(\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} - T \right).$$

4.3. O problema da correspondência

Até o momento, supõem-se conhecidas as coordenadas, nas duas imagens, do mesmo ponto de interesse do ambiente para realizar a reconstrução estéreo. No entanto, este é a maior dificuldade encontrada na reconstrução tridimensional. Este problema divide-se em duas partes: como identificar, em uma das imagens, um ponto (ou conjunto de pontos) significativos para a aplicação em questão e, dado este ponto (ou conjunto de pontos), como encontrar, na segunda imagem, o mesmo ponto (ou conjunto de pontos) selecionado na primeira.

Neste trabalho, a seleção dos pontos na primeira imagem é feita pela segmentação das regiões, sendo os pontos selecionados correspondentes ao centróides das regiões segmentadas. Da mesma maneira, os centróides das regiões segmentadas na segunda imagem são selecionados. Como as duas câmeras utilizadas estão paralelas entre si e separadas por uma pequena distância (cerca de 20cm), os centróides da segunda imagem não correspondem exatamente aos centróides da primeira imagem, mas estão bastante próximos dos pontos verdadeiramente correspondentes.

A abordagem adotada neste trabalho utiliza uma técnica de *adequação por modelo* (*template matching* ou *block Matching*), onde os centróides da segunda imagem servirão como pontos iniciais para a busca iterativa dos pontos correspondentes aos centróides da primeira imagem. O modelo corresponde à região em torno do ponto de interesse na primeira imagem. É feita então uma busca na segunda imagem, a partir do ponto inicial (centróide) disponível, por uma região cuja diferença dos valores de intensidade dos pixels em relação ao modelo seja a menor possível. Esta busca visa minimizar o quadrado das diferenças entre os valores dos pixels do modelo e da região sendo procurada. Assim, sejam I_1 e I_2 as imagens capturadas pelas câmeras esquerda e direita, respectivamente. R_1 e R_2 são as regiões (blocos) de dimensão $W \times W$ em torno de um dos centróides na primeira e na segunda imagem, $\overline{R_1}$ e $\overline{R_2}$ são os valores médios das bandas RGB dos blocos selecionados nas duas imagens, (x_2, y_2) são as coordenadas do centróide na segunda imagem, (x_r, y_r) são as coordenadas do ponto resultante da busca e $W \times W$ é a região, centrada em (x_2, y_2) , explorada pela busca. O algoritmo desenvolvido neste trabalho, *GBM* (*Global Block Matching*), ilustrado abaixo, determina o melhor ponto candidato, correspondente ao centróide da primeira imagem. A subtração dos valores médios das bandas RGB dos blocos de imagem é feita para diminuir a influência da diferença de iluminação entre as duas imagens.

GBM ($x_2, y_2, R_1, R_2, \overline{R_1}, \overline{R_2}$)

$$(x_r, y_r) = (x_2, y_2)$$

$$R_1 = R_1 - \overline{R_1}$$

```

 $R_2 = R_2 - \overline{R_2}$ 
 $mindist = \sum (R_2 - R_1)^2$ 
for  $i = x_2 - Wd$  to  $x_2 + Wd$ 
    for  $j = y_2 - Wd$  to  $y_2 + Wd$ 
         $R_2 = I_2[i - W : i + W, j - W : j + W]$ 
         $R_2 = R_2 - \overline{R_2}$ 
         $dist = \sum (R_2 - R_1)^2$ 
        if  $dist < mindist$ 
            then
                 $mindist = dist$ 
                 $(x_r, y_r) = (i, j)$ 
return  $(x_r, y_r)$ 

```

5. Resultados Experimentais

Para demonstrar o funcionamento do sistema de visão desenvolvido, foi conduzido um experimento no qual as posições de um alvo e de um obstáculo foram mensuradas de três maneiras diferentes: com uma trena manual (para gerar valores de referência), com os sonares e com o sistema de visão.

O arranjo estéreo binocular foi calibrado na faixa de 1 a 4 metros, através de 35 pares de imagens (352x288 pixels, 24bit de cores) correspondentes à observação – de diferentes pontos de vista – do gabarito de calibração mostrado na Figura 2. Os parâmetros utilizados para a aplicação do *Block Matching* foram $W = 40$ e $Wd = 20$. Duas câmeras *Koddo DSP 21X*, alinhadas paralelamente e separadas por uma distância de 20cm, foram utilizadas para a aquisição das imagens. O robô utilizado foi o *Pioneer 2DX*, que possui seis sonares frontais e dois laterais. Destes sonares, apenas os dois sonares da parte frontal central do robô foram utilizados, uma vez que o campo visual comum das duas câmeras coincide com a faixa de detecção destes dois sonares. Durante os testes, o alvo e os obstáculos foram movidos de modo que cada um destes fosse detectado por um dos sonares.

O robô foi colocado em oito posições diferentes em relação ao alvo e ao obstáculo, a partir das quais as posições dos mesmos em relação ao robô foram medidas. O conjunto de imagens utilizado para a calibração, os pares de imagens capturados pelo robô nas oito posições e as imagens resultantes da segmentação baseada em cores estão disponíveis no site do projeto [Pacheco, 2003]. A Figura 3 mostra o par de imagens capturado na posição 1.

A Tabela 1 mostra os valores obtidos para as posições utilizando os três métodos de localização, juntamente com o erro dos dois últimos em relação ao primeiro (o erro corresponde à distância euclidiana entre o ponto localizado com a trena manual e o ponto localizado com o método em questão). A última coluna mostra a distância entre o centróide do objeto resultante da segmentação na segunda imagem e o ponto resultante da

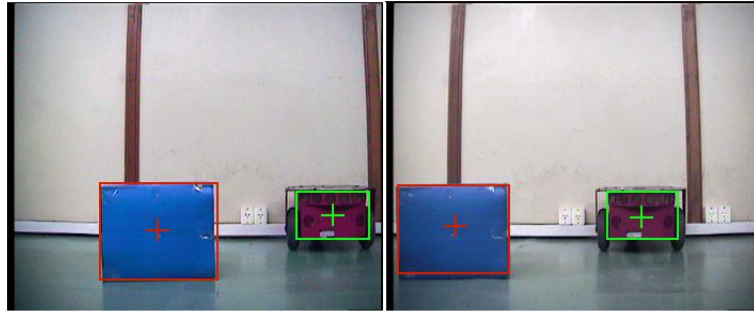


Figura 3: Par de imagens capturados pelo arranjo estéreo a partir da posição 1: um obstáculo (caixa azul, à esquerda) e um alvo (Robô vermelho, à esquerda) na frente do robô. Os retângulos que envolvem o alvo e o obstáculo mostram o resultado da segmentação das regiões.

Tabela 1: Localização do alvo e do obstáculo.

		Trena Manual		Sonares			Estereoscopia			
Posição	Objeto	x (mm)	y (mm)	x (mm)	y (mm)	erro (mm)	x (mm)	y (mm)	erro (mm)	erro (pixels)
posição 1	Alvo	1540	-290	1496	-251	59	1595	-338	73	4
	Obstáculo	1240	270	1178	196	97	1551	206	318	12
posição 1	Alvo	1750	-320	1667	-282	91	1774	-424	107	3
	Obstáculo	1360	300	1311	219	95	1701	291	341	10
posição 1	Alvo	1830	-380	2165	-369	335	1972	-462	164	3
	Obstáculo	1560	370	1508	254	127	2113	407	554	12
posição 1	Alvo	2120	-290	2036	-347	102	2332	-304	212	2
	Obstáculo	1750	320	1682	284	77	1850	382	118	2
posição 1	Alvo	2440	-380	2398	-410	52	2781	-308	349	2
	Obstáculo	2030	420	2000	340	85	2036	487	67	3
posição 1	Alvo	2830	-580	2790	-484	104	3162	-508	340	1
	Obstáculo	2410	430	2329	395	88	2511	542	151	1
posição 1	Alvo	2970	-710	2932	-505	208	3246	-766	282	1
	Obstáculo	2800	430	2709	465	97	3374	482	576	4
posição 1	Alvo	3260	-720	—	—	—	3694	-704	434	3
	Obstáculo	3160	550	—	—	—	3315	594	161	2

busca através de *Block Matching* (coordenadas (x_2, y_2) e (x_r, y_r) do algoritmo, respectivamente). A Tabela 2 mostra o erro médio obtido na localização do alvo e do obstáculo com o uso dos sonares e do sistema de visão.

6. Discussão e Conclusão

Apesar do erro obtido com o sistema de visão ser um pouco maior do que o obtido com os sonares, a precisão deste sistema ainda é aceitável para a tarefa de navegação robótica. Além disso, na Tabela 1 pode-se ver que os sonares não conseguem localizar o alvo e o obstáculo na posição 8 (mais distante), uma vez que o alcance dos sonares utilizados é de 3 metros. O alcance da estereoscopia, por sua vez, é limitado apenas pela região do ambiente utilizada durante a calibração do arranjo estéreo e pela distância entre as câmeras, uma vez que esta distância determina o campo visual comum das duas câmeras e a precisão da estereoscopia (quanto maior a distância, maior a precisão, porém menor o campo visual comum). Outro resultado que chama a atenção é a diferença no erro médio obtido com o sistema de visão para o alvo e o obstáculo. Como se pode ver na Figura 3, o obstáculo (caixa a esquerda) apresenta uma composição de cor mais homogênea do que o alvo (robô a direita). Isto faz com que a eficiência da utilização do *Block Matching*

Tabela 2: Erros Médios da Localização

	Sonares	Estereoscopia
	Erro Médio (mm)	Erro Médio (mm)
Alvo	136	245
Obstáculo	95	286

seja menor no obstáculo do que no alvo, acarretando a obtenção de um erro maior para o primeiro. Assim, na última coluna da Tabela 1 pode-se ver que os erros em pixels associados ao obstáculo são sempre maiores do que os associados ao alvo. Por outro lado, o erro obtido com os sonares foi maior na detecção do alvo, pois a geometria do alvo (robô) é curva (e a do obstáculo, plana).

Por último, e mais importante, o sistema de visão consegue distinguir o alvo do obstáculo – o que não é possível quando se utiliza apenas os sonares –, dando maior flexibilidade sensorial ao robô.

Referências

- Arkin, R. C. (1999). *Behavior-Based Robotics*. MIT Press.
- Bouguet, J.-Y. Camera calibration toolbox for matlab. Disponível em: http://vision.caltech.edu/bouguetj/calib_doc/index.html. Acesso em: 06/02/03.
- Bruce, J., Balch, T., and Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS'00*, volume 3, pages 2061–2066.
- Costa, A. H. R. (2003). Robótica móvel inteligente: Progressos e desafios. tese de livre docência.
- Faugeras, O. (1987). *Three-Dimensional Computer Vision – A Geometric ViewPoint*. MIT Press.
- Heikkilä, J. and Silvén, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition – CVPR'97*, pages 1106–1112.
- Pacheco, R. N. (2003). Website do projeto react. Disponível em: <http://www.lti.pcs.usp.br/robotics/react>. Acesso em: 28/03/03.
- Pacheco, R. N. and Costa, A. H. R. (2002). Navegação de robôs móveis utilizando o método de campos potenciais. In Sakude, M. T. S. and de A. Castro Cesar, C., editors, *Workshop de Computação – WORKCOMP'2002*, pages 125–130, Instituto Tecnológico de Aeronáutica – ITA, São José dos Campos, SP.
- Tsai, R. Y. (1987). A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, RA-3(4):323–344.
- Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the IEEE International Conference on Computer Vision – ICCV'99*, pages 666–673.