

LOCALIZAÇÃO DE ROBÔS MÓVEIS USANDO VISÃO MONOCULAR E RETAS VERTICAIS

ROBERTO J. G. BARRA*, ANNA HELENA REALI COSTA*

**Laboratório de Técnicas Inteligentes – LTI
Escola Politécnica da Universidade de São Paulo – EPUSP
Av. Prof. Luciano Gualberto, trav.3, nr.158
05508-900 São Paulo - SP, Brazil*

Emails: roberto.barra@poli.usp.br, anna.reali@poli.usp.br

Abstract— A key component of a mobile robot system is the ability to localize itself accurately, which involves estimating its pose with respect to some global representation of space using sensor data corrupted by noise. In this work we propose a system that estimates the localization of a mobile robot equipped with odometers and a video camera that operates successfully in stationary and structured indoor environments. To reduce computational complexity, the system observes environment features that generate vertical lines as projections. Promising results were achieved during experiments using a real robot.

Keywords— Mobile Robots, Localization, Computer Vision, Matching

Resumo— Um componente fundamental no sistema de um robô móvel consiste na habilidade de localizar-se acuradamente, o que envolve estimar sua postura em relação a uma representação global do espaço usando dados sensoriais corrompidos por ruído. É proposto um sistema que estima a localização de um robô móvel equipado com odômetros e uma câmera de vídeo, o qual opera, com sucesso, em um ambiente interno, estruturado e estático. Para reduzir a complexidade computacional, o sistema observa elementos do ambiente que geram retas verticais como projeções. Resultados promissores foram obtidos durante experimentos usando um robô real.

Palavras-chave— Robôs Móveis, Localização, Visão Computacional, Correspondência

1 Introdução

Para que um robô móvel consiga executar determinadas tarefas fundamentais, como navegar entre dois pontos pré-determinados, por exemplo, é necessário ter o conhecimento de sua postura (posição e orientação) no ambiente.

Neste trabalho é apresentado um método que combina informações provenientes de visão monocular e odômetros para determinar a postura de um robô, com o objetivo de obter um sistema de localização acurado e de baixo custo. Para combinar as observações e obter a estimativa da postura é usado o Filtro de Kalman Unscented (Wan and van der Merwe, 2000), um estimador não-linear gaussiano.

Uma das dificuldades encontradas em usar visão numa aplicação que precisa fornecer resultados num curto intervalo de tempo é o grande fluxo de informação produzido pela câmera. O foco principal do tratamento da visão nesse trabalho está em minimizar o custo computacional dos algoritmos que lidam com o fluxo de informação da câmera. Duas abordagens foram seguidas: a primeira implica em reduzir a complexidade computacional dos algoritmos adotados, de modo a minimizar o tempo de retardo entre a captura de um quadro pela câmera e a geração de uma estimativa da postura do robô baseada nesse quadro. A segunda implica em aumentar o tempo decorrido entre dois quadros tratados, deixando assim a unidade de processamento livre para que a plataforma

posse ser compartilhada por outros módulos, responsáveis por outras funcionalidades, como navegação e planejamento, por exemplo.

O algoritmo para o tratamento da visão monocular proposto opera detectando nos quadros capturados elementos do ambiente que geram projeções na forma de segmentos de retas verticais, e associando-as a elementos previamente conhecidos, denominados de marcos.

A escolha pelo uso de projeções em forma de retas verticais deve-se aos seguintes motivos:

- estão presentes em quase todos ambientes internos, como escolas, hospitais e escritórios, e em alguns ambientes externos, como postes e prédios;
- têm facilidade de identificação e extração a partir dos quadros capturados;
- apresentam boa robustez a ruídos e oclusão;
- permitem uma representação matemática simplificada do ambiente: uma vez detectadas, ignora-se o eixo vertical na representação de sua localização, reduzindo a representação do espaço de três para duas dimensões.

Diversos artigos utilizam visão monocular para determinar a localização de um robô móvel, mas a maioria deles usa a visão combinada com algum outro sensor de profundidade. Borges et al. (2005), por exemplo, utilizam visão monocular e um sensor de profundidade baseado em laser para

resolver o problema da localização e mapeamento local.

Alguns trabalhos optam por não modelar elementos específicos do ambiente e usam os quadros inteiros para identificar certas regiões conhecidas do ambiente. Saedan et al. (2006) seguem essa abordagem num sistema de localização usando visão omni-direcional. A desvantagem dessa abordagem é o custo computacional elevado de comparações entre quadros inteiros.

Outros trabalhos apresentam soluções similares à solução adotada neste trabalho, porém específicas a determinadas situações. Xia et al. (2006) usam visão monocular para corrigir a localização de um veículo autônomo observando faixas de delimitação em ruas. As principais diferenças entre esse trabalho e o apresentado neste artigo estão tanto na forma da detecção e extração das projeções dos quadros quanto na forma de se associar projeções aos marcos.

As contribuições deste artigo consistem na proposição do algoritmo para associar as projeções extraídas dos quadros aos marcos, e no uso desse algoritmo num estimador para determinar a localização de um robô móvel. O sistema apresentado neste trabalho tem seu uso restrito a ambientes estruturados internos, onde podem ser encontrados elementos que produzam projeções em forma de retas verticais. Os resultados experimentais reportados neste trabalho são bastante motivadores, encorajando desenvolvimentos futuros no algoritmo de associação proposto.

O restante deste artigo está organizado da seguinte forma: na seção 2, detalha-se a representação adotada para as projeções e elementos do ambiente. Na seção 3, o algoritmo proposto é detalhado. Na seção 4, o estimador de localização é descrito. Na seção 5, os resultados de alguns experimentos são reportados e, finalmente, na seção 6 são apresentadas as conclusões.

2 Representação do ambiente

Do ponto de vista do algoritmo que trata as observações da visão monocular, o ambiente é composto pelo conjunto dos marcos que produzem projeções na câmera em forma de reta vertical, sendo que para esses marcos algumas características são conhecidas. Devido ao modo como a câmera se encontra alinhada em relação ao ambiente (ver figura 1) marcos que geram projeções de retas verticais podem ser representados no ambiente também por retas verticais. Esses marcos em geral são formados por intersecções de superfícies, ou mudança de cor ou quebra de continuidade de uma superfície.

Tanto os marcos como suas projeções são representados por sua posição (no ambiente, para

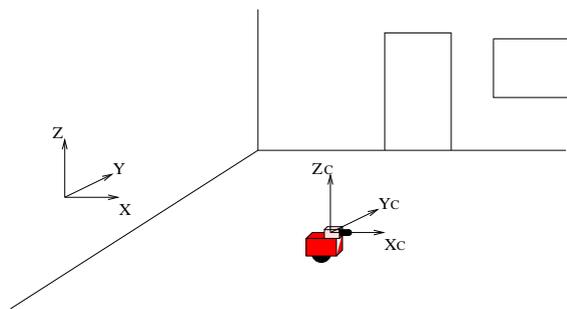


Figura 1: Sistemas de coordenadas adotado para o ambiente (XYZ) e para a câmera ($XcYcZc$). O eixo Zc está alinhado ao eixo Z .

os marcos, e no quadro de vídeo, para as projeções) e por um conjunto de características que captura a distribuição de cores na vizinhança de cada elemento (marco ou projeção). Uma vez realizada as associações, apenas a informação sobre a posição dos marcos é utilizada pelo estimador para estimar a postura do robô. Além do conhecimento da posição dos marcos, o mapa armazena também a posição das paredes principais do ambiente. Esse conhecimento é usado na etapa de associação para determinar quais marcos estão sendo observados.

2.1 Características espaciais

Como neste trabalho todos os elementos de interesse são paralelos à vertical, o mapa do ambiente elimina o eixo Z (segundo a figura 1) e é representado por um plano paralelo ao plano de deslocamento do robô. Os marcos são representados por pontos contidos nesse plano. O plano de projeção da câmera, onde são produzidos os quadros de vídeo, também tem uma dimensão reduzida, sendo representado por uma reta paralela ao eixo Yc , e as retas verticais projetadas são representadas por pontos nessa reta.

2.2 Características fotométricas

Para cada projeção extraída é determinado um conjunto de características baseado na distribuição de cores ao seu redor. Esse conjunto é usado para identificar, em quadros distintos, projeções originadas por um mesmo elemento do ambiente, e também para ajudar a restringir as possibilidades de associação entre projeções e marcos.

Esse conjunto de características é denominado Perfil Médio de Cor (PMC) e foi descrito anteriormente por Barra and Costa (2005). O primeiro passo para se determinar o PMC é a identificação e extração das retas verticais do quadro capturado. Após serem extraídas, o algoritmo considera, para cada projeção

identificada, uma região do quadro capturado centrada na seqüência de pixels que a originou. A região se estende por uma faixa de n pixels para a direita e para a esquerda de cada pixel da seqüência central, ao longo da mesma.

A partir dessa região três vetores (um para cada bande de cor: vermelho, verde e azul) de $2n + 1$ posições são criados. Eles têm o mesmo tamanho da largura da região considerada e cada um é criado do seguinte modo: a posição central do vetor (a posição $n + 1$) recebe o valor médio dos pixels contidos na seqüência central, e cada posição distante i do centro (em ambos os sentidos) recebe o valor médio dos pixels que estão distantes i pixels da seqüência central.

Para os marcos é proposto um conjunto de características similar aos *PMCs* mas menos informativo, porém menos sensível a variações da iluminação do ambiente. Foi criado então a Média de Cores (*MCs*), que armazena, para cada banda de cor, apenas as médias da intensidade à esquerda e à direita das projeções, considerando a mesma vizinhança usada no cálculo dos *PMCs*. Os valores ficam armazenados nas seguintes variáveis: R_{dir} , R_{esq} , G_{dir} , G_{esq} , B_{dir} e B_{esq} , onde R , G e B indicam valores correspondentes às bandas vermelha, verde e azul, respectivamente.

3 Descrição do Algoritmo

O objetivo do algoritmo que trata a visão monocular é identificar projeções de marcos nos quadros capturados, para que, a partir de associações entre os marcos e suas projeções, corrija-se a postura do robô.

A operação do algoritmo se dá em duas fases: a fase de associação e a de validação. A fase de associação associa projeções extraídas do quadro atual a marcos sendo observados. Para isso é usado uma estimativa inicial da postura do robô fornecida pelo estimador (o *UKF*). Em seguida se dá a fase de validação. Essa fase visa minimizar erros de associação gerados pela ambigüidade que aparece na fase de associação devido a incapacidade da visão monocular em determinar diretamente a profundidade que os obstáculos se encontram da câmera. Nessa fase estabelece-se que o mesmo marco seja associado por duas iterações consecutivas a projeções que foram identificadas, por um processo que independe das informações do mapa, como sendo originadas por um mesmo elemento do ambiente. Esse outro processo associa as projeções extraídas no quadro atual com as projeções extraídas no quadro tratado na iteração anterior. O algoritmo pode ser dividido nos seguintes passos:

1. Extração das projeções de retas verticais do quadro sendo tratado na iteração atual. Esse processo é o mesmo adotado por Barra and Costa (2005).
2. Cálculo dos *PMCs* das projeções extraídas.
3. Determinação das associações, com base nos *PMCs*, entre as projeções extraídas nessa iteração e das projeções extraídas na iteração anterior.
4. Determinação das associações entre as projeções e os marcos.
5. Validação das associações entre projeções e marcos. Nesse passo, é verificado se as associações obedecem a condição estabelecida anteriormente, isto é, se o par (projeção, marco) da iteração anterior se mantém na iteração atual. Caso a associação seja validada, ela é usada na correção da postura do robô.

É importante frisar que o algoritmo deve ser rigoroso em relação à geração de associações inválidas entre projeções e marcos, pois é muito mais danoso para a estimativa da postura usar uma associação inválida do que deixar de usar uma válida. Contudo, a etapa de associação entre marcos e projeções (passo 4) não precisa ser rigorosa, bastando fornecer um conjunto de associações com boa probabilidade de serem válidas. A etapa de validação (passo 5) se encarrega eliminar as associações erradas.

3.1 Associação entre projeções de quadros consecutivos

Após extraídas as projeções do quadro de vídeo e calculados os *PMCs* (Barra and Costa, 2005), tenta-se associá-las às projeções obtidas no quadro anterior. Esse problema é conhecido na literatura como o problema da correspondência, sendo muito estudado na literatura de visão estereoscópica. É fato conhecido que o problema da correspondência é tanto mais simples quanto menor for a disparidade entre os quadros (a distância entre as posições onde os quadros foram capturados). Como dito anteriormente, um dos objetivos que se deseja alcançar no tratamento da visão é a possibilidade de aumentar o tempo decorrido entre dois quadros processados para minimizar o uso do processador. Isso implica em não processar todos os quadros fornecidos pela câmera, e também no aumento da disparidade entre quadros consecutivos, o que aumenta a complexidade do problema da correspondência.

Com uma disparidade entre 30 a 90 *cm*, obtida experimentalmente, e sem o conhecimento prévio da posição no ambiente dos elementos sendo observados, não se pode prever como as posições das projeções irão se deslocar entre quadros consecutivos. Esse problema é agravado quando projeções se encontram próximas umas das outras num quadro de vídeo. É interessante notar, porém, que apesar de

aumentar a dificuldade da correspondência, uma maior disparidade torna a etapa de validação mais eficaz, pois quanto maior a disparidade entre quadros tratados, menor a chance que a etapa de associação entre marcos e projeções realize uma associação errada entre um mesmo marco e projeções originadas por um mesmo elemento do ambiente.

A solução adotada neste trabalho faz uso dos *PMCs* para identificar as correspondências, usando a estimativa do deslocamento do robô entre os momentos em que os quadros foram capturados (estimava da disparidade) como uma forma de reduzir o número de possibilidades de associações ente as projeções dos quadros. Barra and Costa (2005) descrevem o método em maiores detalhes.

3.2 Associação entre projeções e marcos

Para realizar a associação entre marcos e projeções é necessário obter o subconjunto dos marcos que potencialmente estão sendo observados pela câmera. Esse conjunto é determinado usando uma estimativa da postura do robô, fornecida pelo estimador (obtida com auxílio das observações anteriores e dos odômetros), e aplicando uma série de restrições ao conjunto dos marcos conhecidos. Primeiro eliminam-se os marcos muito distantes (mais de 50 metros) da estimativa da postura do robô. Essa restrição deve ser calibrada conforme as características da câmera usada. Em seguida são selecionados os marcos que, caso não existissem obstáculos no ambiente, seriam capazes de gerar projeções num plano de projeção da câmera aumentado lateralmente. Essa seleção é feita determinando as retas que unem cada marco à posição estimada do centro de projeção da câmera. Em seguida determinam-se quais dessas retas intersectam um segmento de reta posicionado sobre o plano de projeção da câmera (segundo a estimativa da postura do robô). O centro desse segmento coincide com o centro do plano de projeção, porém tem o dobro de seu tamanho. A última etapa da seleção é a eliminação de marcos cujas retas que os unem ao centro de projeção intersectem alguma parede presente no mapa do ambiente.

Para o conjunto de marcos selecionados determina-se a posição esperada para sua projeção no quadro. Isso é feito calculando a intersecção entre as retas que unem os marcos à estimativa do centro de projeção e a estimativa do plano de projeção.

O próximo passo do algoritmo é criar um conjunto inicial contendo todos os pares possíveis de associações entre marcos e projeções que apresentem similaridade entre si. Essa similaridade é determinada através das características fotométricas, representadas pelos

PMCs e *MCs*. Primeiro os *PMCs* das projeções são convertidos em *MCs*. Essa conversão é realizada descartando o valor central dos vetores que compõem o *PMC* e calculando o valor médio da cada lado dos vetores. A partir dos *MCs*, são calculados $diff_{banda}^{(ij)}$ e $diff_{média}^{(ij)}$ para cada projeção i e marco j , sendo:

$$diff_{média}^{(ij)} = \left| M^{(i)} - M^{(j)} \right| e$$

$$diff_{banda}^{(ij)} = \frac{\left| R_{diff}^{(i)} - R_{diff}^{(j)} \right|}{3} + \frac{\left| G_{diff}^{(i)} - G_{diff}^{(j)} \right|}{3} + \frac{\left| B_{diff}^{(i)} - B_{diff}^{(j)} \right|}{3},$$

com: $M^{(i)} = \frac{R_{dir}^{(i)} + R_{esq}^{(i)} + G_{dir}^{(i)} + G_{esq}^{(i)} + B_{dir}^{(i)} + B_{esq}^{(i)}}{6}$,
 $R_{diff}^{(i)} = R_{dir}^{(i)} - R_{esq}^{(i)}$, $G_{diff}^{(i)} = G_{dir}^{(i)} - G_{esq}^{(i)}$ e
 $B_{diff}^{(i)} = B_{dir}^{(i)} - B_{esq}^{(i)}$. Para $M^{(j)}$, $R_{diff}^{(j)}$, $G_{diff}^{(j)}$ e $B_{diff}^{(j)}$ as equações são equivalentes.

As associações são rejeitadas caso $(30 - diff_{banda}^{(ij)}) < 0$ ou $(100 - diff_{média}^{(ij)}) < 0$, sendo 30 e 100 valores definidos experimentalmente. Caso sejam aceitas, é calculado o parâmetro C'_{ij} , usado na próxima etapa:

$$C'_{ij} = \frac{0,7(30 - diff_{banda}^{(ij)}) + 0,3(100 - diff_{média}^{(ij)})}{0,7 \cdot 30 + 0,3 \cdot 100}.$$

Esse parâmetro foi obtido empiricamente, de modo a balancear a importância tanto da intensidade média da imagem ao redor das retas verticais quanto da variação de cor responsável pela projeção da reta vertical. A partir de agora, (ij) será usado para representar a associação entre a projeção i e o marco j . Esse filtro em geral reduz consideravelmente o número de associações possíveis de serem realizadas, o que permite que as demais etapas adotem algoritmos de maior complexidade computacional.

A próxima etapa consiste em obter uma série de conjuntos de (ij) , sendo que, no fim do algoritmo, as associações entre marcos e projeções detectadas virão de um desses conjuntos. Esses conjuntos são gerados levando em conta a distância entre as projeções extraídas das projeções estimadas para os marcos. Considerando que a postura estimada do robô não seja muito distante da postura real, é esperado que o conjunto dos valores formados pelas distâncias entre cada projeção extraída e a projeção esperada do marco que a gerou apresente valores próximos.

Para cada (ij) é calculado o parâmetro $u_{shift}^{(ij)}$, que é a diferença entre a posição da projeção extraída i e da projeção estimada do marco j . É criado um vetor contendo todos os (ij) ordenados pelo seu valor de $u_{shift}^{(ij)}$. Em seguida, o vetor é

percorrido diversas vezes por uma janela móvel de tamanho variável. A cada passo, o começo da janela avança sobre uma posição do vetor, e o fim da janela é determinado pela diferença entre o valor de $u_{shift}^{(ij)}$ da primeira e da última posição que ela abrange. Para cada passagem da janela, adota-se um valor diferente da diferença máxima de $u_{shift}^{(ij)}$ permitida dentro da janela. Foram adotados 4 valores máximos para a janela: 3, 6, 12 e 24. Para cada conjunto de (ij) delimitado pelas janelas, determina-se um conjunto de associações respeitando a seguinte regra: de cada janela é gerado um conjunto com o maior número de (ij) possível sem que uma mesma projeção ou marco se repita entre os (ij) do conjunto.

O critério adotado para comparar os conjuntos obtidos e determinar qual será escolhido como resposta do algoritmo considera as características fotométricas, através do valor de C'_{ij} , e as distâncias entre projeções extraídas e estimadas. O critério principal derivado a partir dos valores de $u_{shift}^{(ij)}$ é a variância: para cada conjunto k , determina-se a variância dos valores de $u_{shift}^{(ij)}$, $Var(u_{shift}^{(k)})$, e o conjunto com menor $Var(u_{shift}^{(k)})$ tem maior probabilidade de conter o maior número de associações corretas.

Notou-se, porém, que quanto maior o valor médio dos $u_{shift}^{(ij)}$ e quanto mais espaçadas entre si se encontrem as projeções contidas num conjunto, maior o valor que $Var(u_{shift}^{(k)})$ do conjunto tende a assumir. Para equilibrar esse efeito o valor médio de $u_{shift}^{(ij)}$ no conjunto, $\bar{u}_{shift}^{(k)}$ e o espaçamento máximo entre as projeções contidas num conjunto, d_{max} são usados para amortecer a contribuição de $Var(u_{shift}^{(k)})$. O número de associações (ij) contidos num conjunto também tende a influenciar no valor de $Var(u_{shift}^{(k)})$, sendo que quanto maior o número, menor o valor de $Var(u_{shift}^{(k)})$. Levando em conta esses fatores determinou-se empiricamente um parâmetro para auxiliar na seleção do conjunto com maior probabilidade de conter associações corretas:

$$F_{shift}^{(k)} = \frac{\sqrt{n_k}}{Var(u_{shift}^{(k)}) + \frac{\bar{u}_{shift}^{(k)} d_{max}}{7}}, \quad (1)$$

sendo n_k o número de (ij) no conjunto k . A partir de $F_{shift}^{(k)}$ e da média dos C'_{ij} de cada conjunto, $\bar{C}_{ij}^{(k)}$, obtém-se o parâmetro Sc_k :

$$Sc_k = \bar{C}_{ij}^{(k)} + 100.F_{shift}^{(k)},$$

onde a multiplicação por 100 compatibiliza os dois termos e determina a importância de um sobre o outro. Esse valor deve ser controlado conforme a qualidade dos valores de MC disponíveis para os marcos. Como resposta dessa etapa, são escolhidos como associações entre projeções e

marcos os pares contidos no conjunto com maior Sc_k .

4 Estimador

Para determinar a estimativa da postura combinando as observações dos odômetros e da visão monocular foi adotado o Filtro de Kalman Unscented (Julier et al., 1995), que é um estimador bayesiano recursivo, e opera de forma similar a seus semelhantes, como o Filtro de Kalman e o Filtro de Kalman Estendido: um modelo de dinâmica estima a evolução do estado entre duas iterações, e um modelo de observação gera observações esperadas que são usadas, juntamente com as observações medidas pelos sensores, quando essas estão disponíveis, para corrigir a estimativa gerada pelo modelo de dinâmica.

Neste trabalho, o estado é a postura do robô, e, assim como feito por Leonard and Durrant-Whyte (1992), o modelo de dinâmica utiliza as observações dos odômetros para determinar a predição da postura. As projeções extraídas dos quadros são consideradas como observações do sensor de visão, e as projeções estimadas dos marcos, obtidas durante a associação entre projeções e marcos, são as observações esperadas fornecidas pelo modelo de observação.

5 Experimentos

Foram realizados experimentos para determinar o parâmetro Sc_k do algoritmo de tratamento da visão monocular e a distância percorrida pelo robô entre quadros consecutivos tratados, d_{it} . Primeiro calibrou-se Sc_k , visando minimizar o número de associações incorretas, e em seguida variou-se d_{it} , visando encontrar seu valor ótimo. Notou-se que variando d_{it} entre 10 cm e 40 cm os resultados obtidos variavam pouco, porém, a partir dos 40 cm até 90 cm, os resultados se deterioravam. A partir desses resultados fixou-se d_{it} em 37,5 cm.

A figura 2 mostra o gráfico de erro das estimativas obtidas em outro experimento, onde o robô percorreu um trajeto de aproximadamente 21 metros num corredor de uma faculdade, no qual a visão monocular observou em média 8 marcos por quadro tratado, dos quais 3, em média, foram usados para corrigir a postura. A trajetória real percorrida foi obtida através da medição manual de 37 pontos por onde o robô passou, com incerteza de até 3 cm nessas medições. No gráfico a linha pontilhada mostra a curva de erro das estimativas obtidas usando apenas odômetros, enquanto que a linha cheia mostra o erro usando a visão monocular. O erro médio usando visão monocular ficou próximo a 4 cm, valor que pode ser considerado baixo ao se comparar a outros trabalhos publicados, mesmo levando em conta

o pequeno tamanho do percurso (21 metros). A variação observada na curva de erro usando apenas odômetros é justificada pelo fato do robô ter percorrido um trajetória num circuito semi-fechado, na forma de um 'U'. Nesse tipo de trajeto parte do erro acumulado se afastando da posição inicial tende a ser cancelado durante a aproximação à mesma.

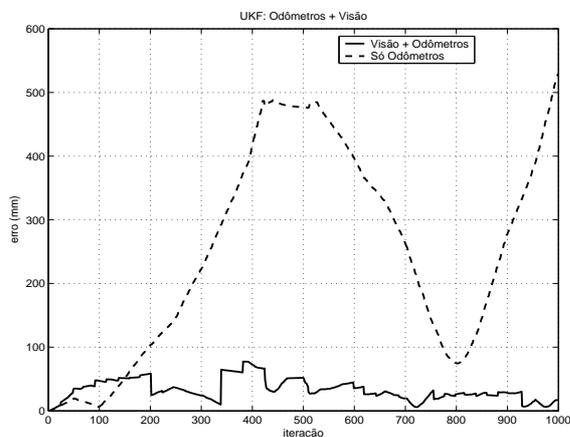


Figura 2: Curvas de erro da estimativa da posição usando apenas odômetros e odômetros e visão.

Nesses experimentos também foi mensurado o tempo de execução de uma iteração do estimador usando e não usando a visão monocular. Os resultados estão na tabela 1. Além dos valores absolutos, uma coluna adicional exibe os tempos de execução normalizados por uma operação padrão de processamento de imagem. Essa coluna foi adicionada para desvincular os resultados da plataforma de testes e da linguagem de programação adotadas nos experimentos. A operação adotada para a normalização foi a convolução de um quadro de vídeo por uma matriz com 3 colunas e 3 linhas. O tempo de execução médio obtido para essa operação foi de 30 ms.

Tabela 1: Tempos de Execução

Sensores / Estimador	Tempo	Norm.
odômetros	4 ms	0,13
odômetros + visão	88 ms	2,93

6 Conclusões

Foi apresentado um algoritmo para determinar a estimativa da postura de um robô usando visão monocular e odômetros. O algoritmo localiza no ambiente elementos cujas projeções geradas são retas verticais e as associa a marcos conhecidos. Os experimentos mostraram bons resultados, indicando que o algoritmo atende aos objetivos propostos, tanto em acurácia como em tempo de execução.

O próximo passo a ser realizado é o estudo da viabilidade do uso de técnicas alternativas para o algoritmo de associação entre marcos e projeções, como o JPDA (Bar-Shalom and Li, 1995) e o MHT (Reid, 1979).

Referências

- Bar-Shalom, Y. and Li, X. R. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs, CT.
- Barra, R. J. G. and Costa, A. H. R. (2005). Correspondência rápida de retas verticais entre imagens para localização de robôs móveis, *Proc. of the VII Sim. Brasileiro de Automação Inteligente*, Sociedade Brasileira de Automática, São Luís, MA, Brasil.
- Borges, G. A., Aldon, M. J., Alcalde, V. H. and Suarez, L. R. (2005). Local map building for mobile robots by fusing laser range finder and monocular video images, *Proc. of the VII Sim. Brasileiro de Automação Inteligente*, Sociedade Brasileira de Automática, São Luís, MA, Brasil.
- Julier, S. J., Uhlmann, J. K. and Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems, *Proc. of the IEEE American Control Conference*, Vol. 3, IEEE Computer Society Press, Seattle, Washington, pp. 1628–1632.
- Leonard, J. J. and Durrant-Whyte, H. F. (1992). *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic publishers, Norwell, MA, EUA.
- Reid, D. (1979). An algorithm for tracking multiple targets, *IEEE Transactions on Automatic Control* **AC-24**: 843–854.
- Saedan, M., Lim, C. W. and Ang, M. H. (2006). Vision-based localization using a central catadioptric vision system, *Proc. of the 10th Int. Sym. on Experimental Robotics*, Rio de Janeiro, Brazil.
- Wan, E. A. and van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation.
- Xia, T. K., Yang, M. and Yang, R. Q. (2006). Vision based global localization for intelligent vehicles, *Proc. of the IEEE Intelligent Vehicles Int. Sym.*, IEEE Computer Society Press, Tokyo, Japan, pp. 571–576.