

UMA ABORDAGEM BASEADA EM RETAS PARA SLAM EM AMBIENTES PLANOS USANDO VISÃO MONOCULAR

ANDRÉ M. SANTANA*, ADELARDO A. D. MEDEIROS†

**Departamento de Informática e Estatística - DIE
Universidade Federal do Piauí - UFPI
Teresina, PI, Brasil*

*†Departamento de Engenharia de Computação e Automação - DCA
Universidade Federal do Rio Grande do Norte - UFRN
Natal, RN, Brasil*

Emails: andremacedo@ufpi.edu.br, adelardo@dca.ufrn.br

Abstract— This work purposes a line-based approach to the SLAM problem using planar information. The solution presented is based on Extended Kalman Filter, in Hough transform and in use of a homographic matrix to perform the image-world mapping. The proposed model allows the use of line features directly in the Kalman equations without additional intermediate calculations and startup delayed initialization, same images being obtained from a single camera. Results with a real robot are presented to validate the proposal.

Keywords— SLAM, Kalman Filter, Hough Transform, Monocular Vision.

Resumo— A proposta deste trabalho consiste em uma abordagem baseada em retas, usando informação planar, para o problema de SLAM (Simultaneous Localization and Mapping). A solução apresentada é baseada no Filtro de Kalman Estendido, na utilização da Transformada de Hough para identificar linhas nas imagens e no uso de uma matriz de homografia para realizar o mapeamento imagem-mundo. A modelagem proposta permite utilizar as características das retas diretamente nas equações de Kalman sem a necessidade de cálculos intermediários e de inicialização atrasada, mesmo as imagens sendo obtidas a partir de uma única câmera. Resultados com um robô real são apresentados a fim de validar a proposta.

Keywords— SLAM, Filtro de Kalman, Transformada de Hough, Visão Monocular.

1 Introdução

Um atributo essencial para autonomia de robôs é uma percepção confiável do mundo. Além da confiabilidade, para a aceitação generalizada de aplicações, é necessário que as tecnologias utilizadas proporcionem uma solução com preço acessível, ou seja, que os componentes utilizados sejam de baixo custo. Uma solução é utilizar sensores óticos nos robôs para resolver os problemas de percepção.

Devido à utilização maciça de câmeras digitais pessoais, câmeras em computadores e celulares, o preço do sensor de imagem diminuiu significativamente e os fez muito atraentes. Além disso, as câmeras podem ser usadas para resolver uma série de problemas-chave na robótica e em outras operações automatizadas, pois fornecem uma variedade muito grande de informação do ambiente, consomem pouca energia e são facilmente integradas ao hardware do robô. O objetivo principal é tirar proveito deste poderoso e barato sensor e criar algoritmos confiáveis e eficazes para resolver problemas na robótica.

Em SLAM, um robô móvel usa seus sensores para explorar o ambiente, ganha conhecimento sobre ele, interpreta o cenário, constrói um mapa adequado e, em seguida, calcula a sua posição relativa usando o mapa que está sendo criado. Quando câmeras são utilizadas como sensor principal dos robôs para tarefas de SLAM o termo visual SLAM é usado para designar este processo.

Os principais desafios em SLAM visual são: a) como detectar características em imagens; b) como reconhecer que uma característica detectada é ou não a mesma que uma detectada previamente; c) como decidir se uma nova característica detectada será ou não adotada como uma nova marca; d) como calcular a posição 3D de marcas a partir de imagens 2D; e e) como estimar a incerteza associada com os valores calculados.

Em geral, todos esses aspectos devem ser resolvidos. No entanto, em situações especiais, é possível desenvolver estratégias específicas para superar todos os esses problemas. O sistema proposto mostra uma solução para monocular SLAM em ambientes planos e com linhas no chão. Esta não é uma condição muito limitante, pois muitos ambientes, como universidades, shoppings, museus, hospitais, casas e aeroportos, por exemplo, possuem linhas como componentes do piso.

O algoritmo usado neste trabalho é baseado no Filtro de Kalman Estendido (FKE), de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de visão monocular. O processamento de imagem identifica linhas no chão usando a Transformada de Hough e suas características são utilizadas como marcos do ambiente. A fase de predição do filtro é feita usando o modelo geométrico do robô e a fase de atualização usa os parâmetros das linhas detectadas por Hough diretamente nas equações de Kalman, sem qualquer etapa de cálculo intermédio.

Nossa abordagem considera que usando linhas existentes como marcas, a complexidade total do SLAM é reduzida: a) linhas podem ser facilmente detectadas em imagens; b) linhas no chão são geralmente igualmente bem espaçadas, então a possibilidade de confusão é reduzida; c) como o número de linhas na imagem não é tão grande, cada nova linha detectada pode ser definida como uma nova marca; d) um chão plano é uma superfície 2D e assim existe uma matriz de conversão constante e fácil de calcular (uma homografia) entre o plano da imagem e o plano do chão, sem incertezas a respeito da informação 3D de profundidade dos pontos; e e) depois do processamento, o número de pixels na imagem que pertencem à linha é uma boa medida de confiança da marca.

A literatura mostra trabalhos um pouco mais antigos (Dao et al., 2003; Smith et al., 2006; Kitano et al., 2007) e mais recentes (Kim and Oh, 2008; Wu et al., 2009) utilizando retas.

Gee et al. (2008) mostram mecanismos para descobrir e, em seguida, incorporar alto nível de estrutura, tais como linhas e superfícies, no mapa do visual SLAM. Lemaire and Lacroix (2008) propõem a utilização de linhas 3D como marcos. Eles comentam que existem algumas vantagens em utilizar linhas: essas primitivas são muito numerosas nos ambientes indoor e, ao contrário do mapa de pontos esparsos, que só são úteis para fins localização, um mapa de segmentos relevantes dá informações sobre a estrutura do ambiente. Ahn et al. (2008) mostram a construção de mapa híbrido para SLAM utilizando pontos 3D e linhas.

Wongphati et al. (2009) implementam fast-SLAM usando retas verticais identificadas por um sistema de visão omnidirecional. Amarasinghe et al. (2009) mostram uma proposta de SLAM baseada em filtro de Kalman estendido integrando laser e câmera onde os marcos identificados são bordas salientes de paredes, mesas e cadeiras. Eade and Drummond (2009) também usam linhas como marcos para um SLAM monocular.

Nossa abordagem difere dos últimos trabalhos apresentados por usar linhas 2D do ambiente como marcos. As linhas extraídas são mapeadas para o plano do robô através do uso de uma homografia e isto permite a utilização das linhas diretamente nas equações de Kalman sem necessitar de uma inicialização atrasada.

2 Modelagem

2.1 Filtro de Kalman Estendido - FKE

O Filtro de Kalman Estendido trabalha com um modelo segundo o Sistema (1) onde $p(\cdot)$ é o modelo não linear do sistema, \mathbf{s}_t é o vetor de estados, \mathbf{u}_{t-1} representa os sinais de entrada e $h(\cdot)$ é o modelo não linear do sensor. Os sinais ε_t e δ_t são ruídos gaussianos de média zero.

$$\begin{cases} \mathbf{s}_t = p(\mathbf{s}_{t-1}, \mathbf{u}_{t-1}, \varepsilon_{t-1}) \\ \mathbf{z}_t = h(\mathbf{s}_t) + \delta_t \end{cases} \quad (1)$$

A cada período de amostragem, o FKE calcula a melhor estimativa do vetor de estados em duas fases: a **predição** que usa o Sistema (2) para prever o estado corrente baseado no estado anterior e nos sinais de entrada aplicados e a **atualização** que usa o Sistema (3) para corrigir a predição do estado pela verificação de sua compatibilidade com as medidas atuais dos sensores.

$$\begin{cases} \bar{\mu}_t = p(\mu_{t-1}, \mathbf{u}_{t-1}, 0) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \end{cases} \quad (2)$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t)) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \quad (3)$$

onde:

$$\mathbf{G}_t = \left. \frac{\partial p(\mathbf{s}, \mathbf{u}, \varepsilon)}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mu_{t-1}, \mathbf{u}=\mathbf{u}_{t-1}, \varepsilon=0} \quad (4)$$

$$\mathbf{V}_t = \left. \frac{\partial p(\mathbf{s}, \mathbf{u}, \varepsilon)}{\partial \varepsilon} \right|_{\mathbf{s}=\mu_{t-1}, \mathbf{u}=\mathbf{u}_{t-1}, \varepsilon=0} \quad (5)$$

$$\mathbf{H}_t = \left. \frac{\partial h(\mathbf{s})}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mu_{t-1}} \quad (6)$$

As matrizes \mathbf{G}_t e \mathbf{V}_t são os jacobianos que linearizam o modelo de processo $p(\cdot)$ e o ruído de processo ε_t respectivamente. \mathbf{K}_t é matriz ganho de Kalman, \mathbf{H}_t é a matriz jacobiana que lineariza o modelo do sensor $h(\cdot)$ e \mathbf{Q}_t é a matriz de covariância do ruído de medição δ_t . Os elementos μ_t e Σ_t são respectivamente a média e a matriz de covariância do vetor de estados \mathbf{s}_t .

2.2 Fase de Predição: Modelo de Processo

Considere um robô com acionamento diferencial, como mostra a Figura 1, em que $\Delta\theta_R$ e $\Delta\theta_L$ são os deslocamentos angulares direito e esquerdo das rodas respectivamente. Assumindo que a velocidade pode ser considerada constante durante um período de amostragem, pode-se determinar o modelo geométrico cinemático do movimento do robô pelo Sistema (7):

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta\theta} [\sin(\theta_{t-1} + \Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \frac{\Delta L}{\Delta\theta} [\cos(\theta_{t-1} + \Delta\theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \quad (7)$$

no qual:

$$\begin{cases} \Delta L = (\Delta\theta_R r_R + \Delta\theta_L r_L)/2 \\ \Delta\theta = (\Delta\theta_R r_R - \Delta\theta_L r_L)/b \end{cases} \quad (8)$$

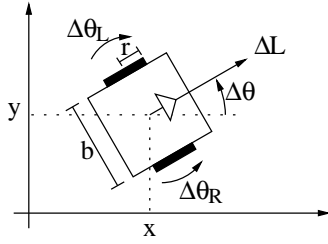


Figura 1: Variáveis do modelo cinemático.

ΔL e $\Delta \theta$ são os deslocamentos linear e angular do robô; b é a distância entre as rodas e r_R e r_L são os raios da roda direita e esquerda, respectivamente. Quando $\Delta \theta \rightarrow 0$, outro sistema, obtido a partir do limite do Sistema (7), deve ser usado.

Foi adotada a abordagem defendida por Thrun et al. (2005) que considera a informação de odometria como sinais de entrada a serem incorporados ao modelo do robô ao invés das medições sensoriais.

As diferenças entre o deslocamento angular real das rodas ($\Delta \theta_R$ e $\Delta \theta_L$) e os deslocamentos medidos pelos encoders ($\Delta \tilde{\theta}_R$ e $\Delta \tilde{\theta}_L$) são modelados por um ruído branco gaussiano: $\Delta \theta_R = \Delta \tilde{\theta}_R + \varepsilon_R$ e $\Delta \theta_L = \Delta \tilde{\theta}_L + \varepsilon_L$. As medidas $\Delta \tilde{L}$ e $\Delta \tilde{\theta}$ são definidas substituindo-se ($\Delta \theta_R$ and $\Delta \theta_L$) por ($\Delta \tilde{\theta}_R$ and $\Delta \tilde{\theta}_L$) no Sistema (8).

Usando os Sistemas (7) e (8) para calcular o modelo de estado $p(\cdot)$, é possível calcular, por derivação do modelo, as matrizes \mathbf{G} e \mathbf{V} (Equações 4 e 5) usadas na fase de predição.

Sabe-se que a odometria introduz erros acumulativos. Portanto, o desvio padrão dos ruídos ε_R e ε_L é assumido ser proporcional ao módulo do deslocamento angular de cada roda. Isto leva à definição da matriz \mathbf{M} dada pela Equação (9).

$$\mathbf{M} = \begin{pmatrix} (M_R |\Delta \tilde{\theta}_R|)^2 & 0 \\ 0 & (M_L |\Delta \tilde{\theta}_L|)^2 \end{pmatrix} \quad (9)$$

2.3 Fase de Atualização: Modelo do Sensor

Os marcos utilizados nesse trabalho são linhas formadas pela junção do piso do ambiente onde o robô navega. O sistema é baseado em um robô que possui uma câmera fixa acoplada à sua estrutura, como mostra a Figura 2.

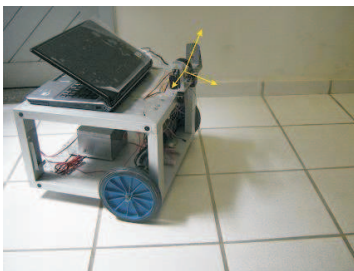


Figura 2: Robô real.

Os marcos são detectados processando as imagens através da transformada de Hough. As linhas de Hough são descritas pelos parâmetros (ρ e α) e a relação entre elas é expressa na Equação (10) onde (ρ) é o módulo e (α) é o ângulo do menor vetor que conecta a origem do sistema de coordenadas à linha (Gonzalez and Woodes, 2007).

$$\rho = x \cos(\alpha) + y \sin(\alpha) \quad (10)$$

Definiu-se um sistema de coordenadas fixo (F) e um móvel (M), anexado ao robô, ambos ilustrados na Figura 5. A origem do sistema móvel tem coordenadas (x_M^F, y_M^F) no sistema fixo. θ_M^F representa a rotação do sistema móvel com respeito ao sistema fixo. Deve-se notar que existe uma relação entre (x_M^F, y_M^F, θ_M^F) e a pose do robô (x_t, y_t, θ_t), que é dada pela Equação (11).

$$x_t = x_M^F \quad y_t = y_M^F \quad \theta_t = \theta_M^F + \pi/2 \quad (11)$$

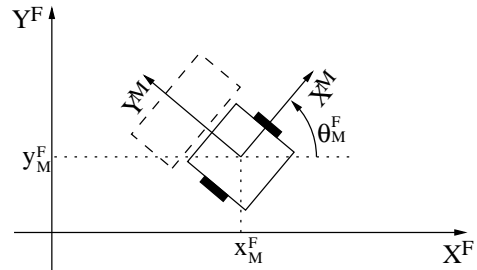


Figura 3: Sistemas de Coordenadas.

Cada linha no chão é descrita por dois parâmetros estáticos (ρ^F, α^F). O mapa a ser produzido pelo processo de SLAM é composto de um conjunto desses pares de parâmetros. Então, o i -ésimo vetor de coordenadas do i -ésimo marco é dado pela Equação (12).

$${}^i c = \begin{bmatrix} {}^i \rho^F \\ {}^i \alpha^F \end{bmatrix} \quad (12)$$

A cada passo o robô captura uma imagem e identifica os parâmetros ($\tilde{\rho}, \tilde{\alpha}$)¹ das linhas detectadas. Os parâmetros da imagem são então convertidos para os seus correspondentes ($\tilde{\rho}^M, \tilde{\alpha}^M$) no sistema de coordenadas móvel (M) anexado ao robô, usando os parâmetros da câmera. O vetor de medições \mathbf{z}_t a ser usado na fase de atualização do FKE (Sistema 3) é definido pela Equação (13).

$$\mathbf{z}_t = \begin{bmatrix} \tilde{\rho}^M \\ \tilde{\alpha}^M \end{bmatrix} \quad (13)$$

Para simplificar a notação, estamos assumindo que há exatamente uma linha por imagem. Porém, na verdade, podemos ter nenhuma, uma ou mais de uma linha por imagem.

¹Usamos ($\tilde{\cdot}$) sobre a variável para indicar os valores medidos ao invés dos calculados.

Para usar a informação diretamente obtida pelo processamento de imagem ($\tilde{\rho}^M, \tilde{\alpha}^M$) na fase de atualização do EKF-SLAM, deve-se deduzir o modelo do sensor $h(\cdot)$, que é o valor esperado desses parâmetros em função das variáveis de estado. Usou-se a relação entre as coordenadas dos sistemas (M) e (F) (Sistema 14) e a Equação (10) em ambos os sistemas de coordenada (Equações 15 e 16).

$$\begin{cases} x^F = \cos(\theta_F^M)x^M - \sin(\theta_F^M)y^M + x_M^F \\ y^F = \sin(\theta_F^M)x^M + \cos(\theta_F^M)y^M + y_M^F \end{cases} \quad (14)$$

$${}^i\rho^F = x^F \cos({}^i\alpha^F) + y^F \sin({}^i\alpha^F) \quad (15)$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \quad (16)$$

Substituindo-se o Sistema (14) em (15), fazendo as equivalências necessárias com a Equação (16) e substituindo algumas variáveis da Equações (11), obtém-se os Sistemas (17) e (18), que representam dois possíveis modelos de sensor $h(\cdot)$ a ser usado no filtro. Para decidir qual modelo usar, calcula-se ambos os valores de α^M e usa-se o modelo que gerar o valor mais próximo de $\tilde{\alpha}^M$.

$$\begin{cases} \rho^M = {}^i\rho^F - x_t \cos({}^i\alpha^F) - y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t + \pi/2 \end{cases} \quad (17)$$

$$\begin{cases} \rho^M = -{}^i\rho^F + x_t \cos({}^i\alpha^F) + y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t - \pi/2 \end{cases} \quad (18)$$

O modelo do sensor é incorporado no FKE através da matriz \mathbf{H} que pode ser obtida a partir da Equação (6).

3 Processamento de Imagem

3.1 Detecção de Linhas

Devido à escolha das linhas do chão como marcos, a técnica adotada para identificá-las foi a transformada de Hough. O propósito dessa técnica é encontrar instâncias imperfeitas de objetos dentro de certa classe de formas por processo de votação efetuado no espaço de parâmetros, no qual objetos candidatos são obtidos como máximo local em uma matriz de acumulação.

Na nossa aplicação as imagens são capturadas em tons de cinza e convertidas para preto e branco usando detector Canny e, posteriormente, aplica-se a transformada de Hough à imagem (Figura 4).

3.2 Conversão para o Mundo

É assumido que o chão é plano e que a câmera é fixa. Então, existe uma relação constante (uma homografia \mathbf{A}) entre os pontos no plano do chão (x, y) e os pontos no plano da imagem (u, v):

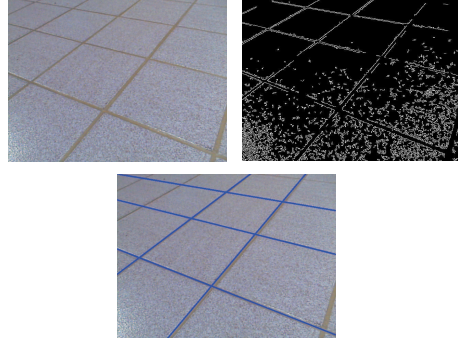


Figura 4: Imagens do piso do ambiente.

O fator de escala s é determinado para cada ponto de tal forma que o valor do terceiro elemento do vetor seja sempre 1.

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (19)$$

A homografia pode ser calculada off-line usando um padrão contendo quatro ou mais pontos notáveis com coordenadas conhecidas (Figura 5). Depois de detectar os pontos notáveis na imagem, têm-se diversas correspondências entre coordenadas de pontos no plano do chão e na imagem. Substituindo-se esses pontos na Equação 19, obtém-se um sistema no qual se pode determinar os oito elementos da matriz de homografia \mathbf{A} .

Uma vez calculada a homografia, para cada linha detectada faz-se o que segue: a) usando os valores de $(\tilde{\rho}, \tilde{\alpha})$ obtidos pela transformada de Hough, calcular dois pontos pertencentes à linha na imagem, b) converter as coordenadas desses dois pontos para o plano do chão usando a homografia \mathbf{A} e, c) determinar $(\tilde{\rho}^M, \tilde{\alpha}^M)$ da linha que passa por esses dois pontos.

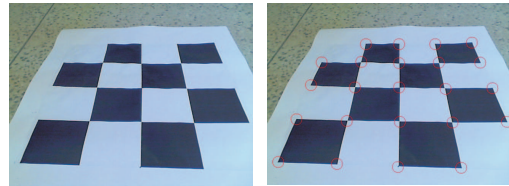


Figura 5: Padrão utilizado.

O erro médio de re-projeção foi de $1.5cm$ e é apresentado como o raio das circunferências cujo centro é o ponto do mundo mapeado na imagem utilizando a inversa da homografia (Figura 5).

3.3 Ruído do Sensor

Como é apresentado na Figura 2, a posição da câmera é de tal forma que o plano da imagem não é paralelo ao plano de chão. O resultando do efeito da inclinação pode ser visto nas Figuras 4 e 5.

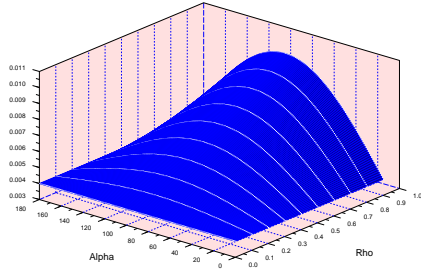


Figura 6: Função variância do sensor.

Por experimentação, observou-se que a informação existente no topo da imagem sofre maior ruído quando comparada com a área inferior, isso nos fez considerar que a variação do ruído deve ser proporcional à distância (ρ) da reta na imagem. Além disso, notou-se que a detecção das linhas é melhor que as verticais, o que nos permitiu inferir que a variância do ruído também é relacionado com o ângulo (α) da linha na imagem. Assim, a variância do ruído do sensor adotada neste trabalho é regida pela Equação (20).

$$\sigma(\rho, \alpha) = a + b \cdot \sin(\alpha) \cdot (\exp(\rho/c) - 1) \quad (20)$$

O termo $[\exp(\rho/c) - 1]$ representa a proporcionalidade à distância, e o termo $[\sin(\alpha)]$, a influência do ângulo. As constantes a , b e c foram calculadas por experimentação e seus valores são: $a = 0.004$, $b = 0.3$ e $c = 45$. A Figura 6 mostra o comportamento da função descrita.

4 Resultados

Os experimentos foram realizados utilizando um robô cujas rodas são acionadas por motores DC com acionamento diferencial. Cada motor tem um encoder óptico e uma placa dedicada baseada em microcontrolador PIC que controla a velocidade local. Além disso, o robô possui, acoplado à sua estrutura, uma webcam colorida e um notebook para processamento das informações (Figura 2).

O local² por onde o robô navegou possui piso formado por cerâmicas quadradas com 25cm de lado (Figura 4). As interseções destas cerâmicas formam linhas horizontais e verticais que são utilizadas como marcos no algoritmo de SLAM. A Figura 7 ilustra as características do ambiente.

Durante o movimento do robô 1962 imagens 640×480 foram processadas em $180ms$, na média. A câmera existente no robô foi posicionada de modo que o ângulo de visão seja da ordem de duas vezes o tamanho do robô.

²Departamento de Física - DF da Universidade Federal do Rio Grande do Norte - UFRN. Este prédio pode ser visto em sistemas de mapeamento online utilizando as coordenadas $5^\circ 50' 29.50''S$ $35^\circ 11' 49.48''W$

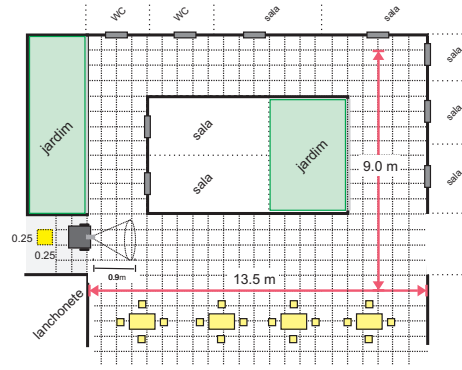


Figura 7: DF-UFRN.

A Equação (21) mostra a matriz de homografia calculada.

$$\mathbf{A} = \begin{pmatrix} 0.1417 & 0.0009 & -49.2065 \\ 0.0073 & -0.0761 & 98.6323 \\ 0.0001 & 0.0029 & 1 \end{pmatrix} \quad (21)$$

Foi comandado ao robô a execução de uma trajetória retangular de circuito fechado e trajetória calculada é apresentada na Figura 8. Os pontos em azul correspondem à trajetória calculada apenas por odometria e os pontos em vermelho à trajetória calculada por SLAM. A distância entre a posição inicial e final, calculadas assumindo uma pose inicial $(0, 0, 0^\circ)$, foi de $0.89m$, usando somente odometria, e $0.02m$, usando SLAM; a distância final real, medida *in loco*, foi muito próxima da calculada por SLAM.

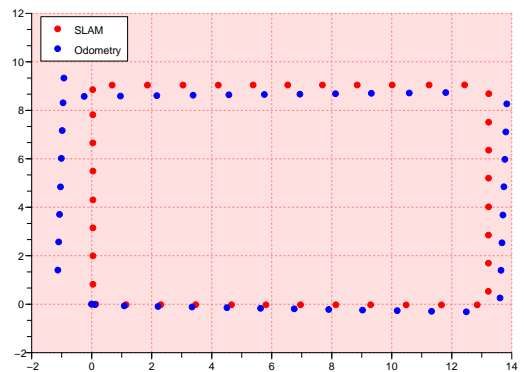


Figura 8: Trajetória executada pelo robô.

Tendo em vista o movimento total realizado pelo robô, a correspondência de linhas detectadas foi correta em 95% dos casos e cada linha foi observada em 15 imagens consecutivas, em média. Em 98% das imagens o sistema detectou linhas: três linhas em 61%, quatro linhas em 26% e cinco linhas em 11% das imagens. Os 2% restantes foram em erros de processamento de imagem.

Ainda sobre trajetória é interessante comentar o comportamento da componente (y_t) da pose do robô no instante do fechamento do laço. Percebeu-se que houve uma mudança no padrão de atualização no instante em que a marca característica do fechamento do circuito é detectada. Em valores, a atualização média que era de 4cm passou para 8cm. Isto acontece porque, pela primeira vez, o robô re-observa uma marca depois de ter se movido por um local onde ela não existe.

Sobre a variância em ρ da linha que caracteriza o fechamento do seu valor antes da re-observação era de 4.6×10^{-5} ($\sigma = 0,68\text{cm}$) e foi reduzido para 9.2×10^{-6} ($\sigma = 0,30\text{cm}$). Esta linha foi observada pela primeira vez no passo 23, e re-observada pela primeira vez no passo 1591.

5 Conclusões e Perspectivas

A principal contribuição deste trabalho é a modelagem do sensor óptico de modo a permitir o uso de parâmetros obtidos a partir do algoritmo de processamento de imagem diretamente nas equações do filtro de Kalman sem fases intermediárias de cálculo de posição ou distância. Além disso, a representação do ambiente utilizando linhas do chão reduz o número de características de referência utilizadas no processo de SLAM. No nosso caso, 194 marcas são usadas para descrever um ambiente com $121,5\text{m}^2$.

Outro ponto importante da nossa abordagem é a utilização, além da odometria, de um sistema de visão monocular e nenhum outro sensor auxiliar como lasers ou sonares. Vale ressaltar que mesmo com visão monocular as retas podem ser usadas no filtro de Kalman no instante da detecção, pois suas características são mapeadas para coordenadas do mundo utilizando a matriz de homografia.

A abordagem proposta não tem nenhuma pretensão de ser geral, pois exige um piso plano e com linhas. No entanto, nos casos onde ela pode ser usada, quando comparada com outras abordagens de SLAM visual, é eficiente tanto em custo computacional, devido ao número reduzido de elementos para representar o ambiente, como em precisão, devido a facilidade na determinação da informação 3D das características. Mesmo em um ambiente de laço fechado o sistema funcionou corretamente reconhecendo linhas previamente detectadas sem a necessidade de procedimentos especiais para lidar com esse problema.

Como trabalho futuro pretende-se: melhorar as propriedades de tempo real do algoritmo de processamento de imagem através da adoção de alguma das variantes da transformada de Hough; lidar com segmentos de reta com comprimento finito e incorporar as informações do ponto inicial e final do segmento como características a ser utilizadas no SLAM; e, por fim, testar a nossa abordagem utilizando outros filtros estatísticos.

Referências

- Ahn, S., Chung, W. and Oh, S. (2008). Construction of hybrid visual map for indoor SLAM, *IEEE Int. Conference on Robotics and Automation*, pp. 2791-2796 .
- Amarasinghe, D., Mann, G. and Gosine, R. (2009). Landmark detection and localization for mobile robot applications: a multisensor approach, *Robotica Cambridge* .
- Dao, N., You, B., Oh, S. and Hwangbo, M. (2003). Visual self-localization for indoor mobile robots using natural lines, *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 1252-1257, .
- Eade, E. and Drummond, T. (2009). Edge landmarks in monocular slam, *Image and Vision Computing*, 27, pp. 588-596 .
- Gee, A., Chekhlov, D., Calway, A. and Cuevas, W. (2008). Discovering higher level structure in visual SLAM, *IEEE Trans. on Robotics*, Vol. 24, No. 5, pp. 980-990 .
- Gonzalez, R. and Woodes, R. (2007). Digital image processing, *Prentice Hall* .
- Kim, S. and Oh, S. (2008). SLAM in indoor environments using omni-directional vertical and horizontal line features, *Journal of Intelligent and Robotic Systems*, 51, pp.31-43 .
- Kitanov, A., Bisevac, S. and Petrovi, I. (2007). Mobile robot self-localization in complex indoor environments using monocular vision and 3D model, *IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics* .
- Lemaire, T. and Lacroix, S. (2008). Monocular SLAM as a graph of coalesced observations, *IEEE Int. Conference on Robotics and Automation*, pp. 2791-2796 .
- Smith, P., Ian, R. and Davison, A. (2006). Real-time monocular slam with straight lines, *British Machine Vision Conference* .
- Thrun, S., Burgard, W., and Fox, D. (2005). Probabilistic robotics, *MIT Press* .
- Wongphati, M., Niparnan, N. and Sudsang, A. (2009). Bearing only fastSLAM using vertical line information from an omnidirectional camera, *IEEE Int. Conference on Robotics and Biomimetics*, pp. 1188-1193 .
- Wu, M., Huang, F., Wang, L. and Sun, J. (2009). Cooperative multi-robot monocular SLAM using salient landmarks, *Int. Asia Conference on Informatics in Control, Automation and Robotics*, pp. 151-155 .