

# A Line-Based Approach to Slam Using Monocular Vision

A. M. Santana and A. A. D. Medeiros

**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.

## I. INTRODUÇÃO

TERMO SLAM é indicado como uma abreviatura para Localização e Mapeamento Simultâneo e foi originalmente proposto por Leonard & Durrant-Whyte [27] com base em trabalhos anteriores de Smith et al. [31].

No problema de 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. Os mapas podem ser representados de várias maneiras, tais como grades de ocupação e mapas de características. Estamos interessados na segunda forma de representação. Maiores detalhes teóricos sobre SLAM pode ser encontrado em Durrant & Bailey [12], [13].

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 do ambiente.

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 que possam extrair as informações necessárias para a resolução de problemas na robótica. Quando câmeras são usadas como sensor principal de um robô para resolver problemas de SLAM, a literatura usa o termo Visual SLAM, ou VSLAM, para designar este processo. Seus principais desafios 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 que será apresentado mostra uma técnica de visual SLAM para 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 processamento de imagem. O processamento de imagem identifica linhas no chão usando 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.

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;

A. M. Santana, Universidade Federal do Piauí, Teresina, PI, Brasil, andremacedo@ufpi.edu.br

A. A. D. Medeiros, Universidade Federal do Rio Grande do Norte, Natal, RN, Brasil, adelardo@dca.ufrn.br

- 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 detectada.

## II. TRABALHOS RELACIONADOS

Recentes extensões ao problema geral de SLAM mostram a possibilidade de usar câmera em vez de sonar ou laser. Exemplos são os trabalhos de Davison et al. [11], Jung [23] e Herath et al. [20], utilizando visão estéreo, bem como os estudos realizados por Davison et al. [10] e Kwok et al. [24], utilizando uma única câmara.

Mansinghka [28] apresenta um visual SLAM para ambientes dinâmicos utilizando transformada SIFT e fluxo ótico. Estrada et al. [14] propõem um método de mapeamento hierárquico que permite obter mapas métricos precisos de grandes ambientes em tempo real. O nível inferior do mapa é composto por um conjunto de mapas locais que são estatisticamente independentes. A parte superior do mapa é um grafo de adjacência cujos arcos são rotulados com a relação entre a localização dos mapas locais e uma estimativa relativa destes mapas é mantida nesse nível em uma relação estocástica. A principal vantagem desta abordagem é ser bastante precisa para o problema de *loop-close*.

Uma solução utilizando informações geométricas do ambiente é apresentada por Chen [4]. Ele comenta que a redundância em SLAM pode reforçar a confiabilidade e precisão das características observadas e por isso as primitivas geométricas comuns em ambientes internos, linhas e quadrados, por exemplo, são incorporadas em um filtro de Kalman estendido para elevar o nível de conhecimento da característica observada.

Frintrop et al. [15] introduzem um novo método para detectar marcos que consiste em um sistema de atenção biologicamente inspirado para detectar regiões contrastantes na imagem. Essa abordagem faz com que regiões sejam facilmente re-detectadas e, por conseguinte, apresente maior facilidade de correspondência.

Dailey & Parnichkun [9] utilizam visão estéreo para um visual SLAM baseado em filtro de partículas. Choi et al. [6] mostram uma abordagem fundido informações de sonares com visão estéreo usando filtro de Kalman estendido.

Reconhecimento automático e registro de objetos como marcos visuais é proposto por Lee & Song [25]. Neste, transformada SIFT e algoritmos de contorno são utilizados para distinguir objetos do fundo da imagem. Quando objetos são detectados e são considerados adequados para a navegação do robô, eles são armazenados e posteriormente podem ser utilizados para corrigir sua pose.

Clemente et al. [8] demonstraram pela primeira vez que SLAM com uma única câmera fornecendo dados de entrada pode atingir larga escala ao ar livre funcionando em tempo real. Eles utilizam inversão de profundidade e mapas hierárquicos no experimento.

Jing Wu & Zhang [22] apresentam um trabalho sobre modelagem de câmera para visual SLAM. O foco deste trabalho é sobre como modelar a incerteza do sensor ótico e como construir os componentes probabilísticos do modelo. Tendo o componente determinístico do processo de calibração da câmera, parâmetros intrínsecos, eles são utilizados para re-projeção do erro. Os erros são então encontrados de acordo com uma distribuição Gaussiana bivariada e a medição da covariância pode ser calculada quando as características estão a diferentes distâncias da câmera.

Estudos recentes em VSLAM mostram como obter informação de profundidade a partir de uma única câmera. Como exemplo tem-se o trabalho de Civera et al. [7]. Hafez et al. [21] também apresentam uma proposta para estimar profundidade e funcionar em tempo real baseada na otimização da técnica de triangulação.

Gee et al. [18] mostram mecanismos para descobrir e, em seguida, incorporar alto nível de estrutura, tais como linhas e superfícies, no mapa do visual SLAM.

Lemaire & Lacroix [26] propõem a utilização de linhas 3D como marcos. Eles comentam que existem algumas vantagens em utilizar linhas 3D: 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.

Também utilizando linhas, no caso verticais, Fu et al. [16] apresentam um trabalho de fusão de informações de laser e câmera em um filtro de Kalman estendido. Neste trabalho as linhas são extraídas da imagem utilizando detector Canny. Ahn et al. [1] mostram uma estratégia de construção de mapa híbrido para SLAM *indoor* utilizando pontos 3D e linhas.

A literatura mostra trabalhos recentes utilizando retas [2], [29], [34]. Wongphati et al. [33] implementam *fast*-SLAM usando retas verticais identificadas por um sistema de visão omnidirecional. Fu & Yang [17] também apresentam uma abordagem utilizando retas para SLAM *indoor*.

Nossa abordagem difere dos últimos trabalhos apresentados nesta seção por usar linhas 2D do ambiente como marcos. As linhas são extraídas da imagem pela Transformada de Hough e são mapeadas para o plano do robô através da utilização de uma matriz de homografia. Os parâmetros de Hough,  $\rho$  e  $\alpha$ , que representam as características de cada reta são usadas nas equações Kalman sem nenhum cálculo intermediário de distância ou posição.

É importante comentar que o uso de linhas para ambientes *indoor* é vantajoso, pois uma pequena quantidade de marcos são necessários para descrever o ambiente e, além disso, utilizar o princípio de que elas estão no mesmo plano fornece informação rica e precisa sobre suas características.

### III. SISTEMA PROPOSTO

O sistema proposto neste trabalho apresenta uma técnica de SLAM visual adequada para ambientes planos e fechados com linhas presentes no chão. O algoritmo utilizado é baseado no Filtro de Kalman Estendido (FKE) de modo a permitir que um robô navegue em um ambiente plano fundindo informações de odometria e de processamento de imagem. As linhas pré-existent no chão são utilizadas como marcos e são identificadas usando a transformada de Hough. A fase de predição do FKE é feita usando o modelo geométrico do robô e a fase de atualização usa diretamente os parâmetros das linhas detectadas por Hough diretamente nas equações de Kalman sem cálculos adicionais intermediários. A Fig. 1 apresenta o resumo da proposta.

#### a) Filtro de Kalman Estendido - FKE

O Filtro de Kalman Estendido trabalha com um modelo segundo o Sistema (1), cujas variáveis são descritas na Tabela I. Os sinais  $\epsilon_t$  e  $\delta_t$  são ruídos gaussianos de média zero.

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

A cada período de amostragem, o FKE calcula a melhor estimativa do vetor de estados em duas fases:

- **fase de predição:** usa o Sistema (2) para prever o estado corrente baseado no estado anterior e nos sinais de entrada aplicados;

- **fase de atualização:** 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,

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

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

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

TABELA I  
DESCRIÇÃO DOS SÍMBOLOS DOS SISTEMAS (1), (2) E (3)

$\mathbf{s}_t$	vetor de estados (ordem $n$ ), instante $t$ ;
$p(\cdot)$	modelo não linear do sistema;
$\mathbf{u}_{t-1}$	sinais de entrada (ordem $l$ ), no instante $t-1$ ;
$\epsilon_{t-1}$	ruído do processo (ordem $q$ ), no instante $t-1$ ;
$\mathbf{z}_t$	vetor de medições (ordem $m$ ), retornadas pelo sensor
$h(\cdot)$	modelo não linear do sensor;
$\delta_t$	ruído de medição;
$\bar{\mu}_t, \mu_t$	média (ordem $n$ ) do vetor de estados $\mathbf{s}_t$ , antes e depois da fase de atualização;
$\bar{\Sigma}_t, \Sigma_t$	matriz de covariância ( $n \times n$ ) do vetor de estados $\mathbf{s}_t$ ;
$\mathbf{G}_t$	matriz jacobiana ( $n \times x$ ) que lineariza o modelo do processo $p(\cdot)$ ;
$\mathbf{V}_t$	matriz jacobiana ( $n \times q$ ) que lineariza o ruído de processo $\epsilon_{t-1}$ ;
$\mathbf{K}_t$	matriz ganho de Kalman ( $n \times m$ );
$\mathbf{H}_t$	matriz jacobiana ( $m \times n$ ) que lineariza o modelo do sensor $h(\cdot)$ ;
$\mathbf{Q}_t$	matriz de covariância ( $m \times m$ ) do ruído de medição $\delta_t$ ;

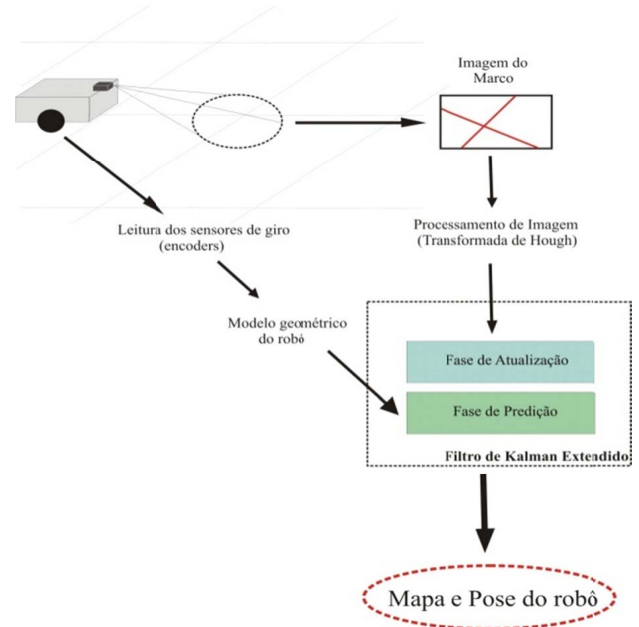


Figura 1. Sistema Proposto.

Na tarefa de SLAM, além de estimar a pose do robô, também são estimadas as características de todos os marcos encontradas pelo caminho. Assim, torna-se necessário incluir tais características no vetor de estado. Se  $i^c$  é o vetor de características do  $i$ -ésimo marco e existem  $k$  marcos, então o vetor de estados é dado pela Equação (7).

$$\mathbf{s}_t = [x_t \ y_t \ \theta_t \ {}^1c_t^T \ \dots \ {}^k c_t^T] \quad (7)$$

Quando o número de marcos ( $k$ ) é conhecido *a priori*, a dimensão do vetor de estados é estática. Caso contrário, ele cresce quando um novo marco é encontrado.

#### b) Fase de Predição: Modelo de Processo

Considere um robô com acionamento diferencial em que  $\Delta\theta_R$  e  $\Delta\theta_L$  são os deslocamentos angulares direito e esquerdo das rodas respectivamente (Fig. 2).

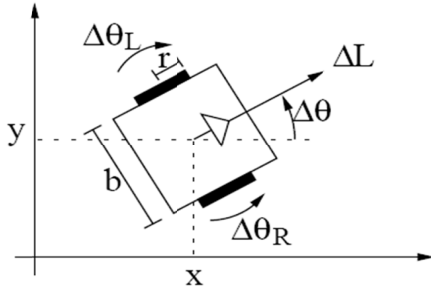


Figura 2. Variáveis do modelo cinemático.

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 (8):

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta\theta} [\sin(\theta_{t-1} + \Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} + \frac{\Delta L}{\Delta\theta} [\cos(\theta_{t-1} + \Delta\theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \quad (8)$$

no qual:

$$\begin{cases} \Delta L = (\Delta\theta_R r_R + \Delta\theta_L r_L)/2 \\ \Delta\theta = (\Delta\theta_R r_R - \Delta\theta_L r_L)/b \end{cases} \quad (9)$$

onde  $\Delta L$  e  $\Delta\theta$  são os deslocamentos linear e angular do robô;  $b$  representa 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 (8), deve ser usado.

Foi adotada a abordagem defendida por Thrun et al. [32], 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 ( $\Delta\theta_R$  e  $\Delta\theta_L$ ) por ( $\Delta\tilde{\theta}_R$  e  $\Delta\tilde{\theta}_L$ ) no Sistema (9). Usando os sistemas (8) e (9) para calcular o modelo de estado  $p(\cdot)$  (Sistema 2), é possível calcular, por derivação do modelo, as matrizes  $\mathbf{G}$  e  $\mathbf{V}$  usadas na fase de predição do filtro (Equações 4 e 5).

Sabe-se que a odometria introduz erros acumulativos. Portanto, o desvio padrão dos ruídos  $\varepsilon_R$  e  $\varepsilon_L$  é assumido ser proporcional ao módulo do deslocamento angular de cada roda. Essas considerações levam à definição da matriz  $\mathbf{M}$  dada pela Equação (10).

$$\mathbf{M} = \begin{pmatrix} (M_R |\Delta\tilde{\theta}_R|)^2 & 0 \\ 0 & (M_L |\Delta\tilde{\theta}_L|)^2 \end{pmatrix} \quad (10)$$

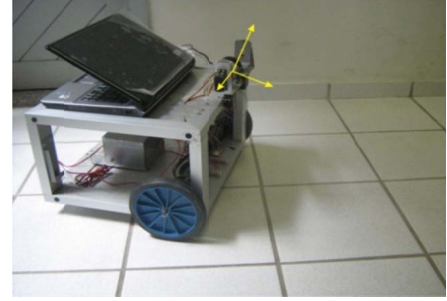


Figura 3. Sistema Robótico.

### c) Fase de Atualização: Modelo do Sensor

Os marcos adotados nesse trabalho são linhas formadas pela junção do piso do ambiente onde o robô navega. O sistema é baseado em um robô com acionamento diferencial que possui uma câmera fixa acoplada à sua estrutura, como mostra a Fig. 3.

Os marcos são detectados processando as imagens através da transformada de Hough. As linhas detectadas são descritas por parâmetros  $\rho$  e  $\alpha$  da Equação (11). A Fig. 4 mostra a representação geométrica desses parâmetros:  $\rho$  é o módulo e  $\alpha$  é o ângulo do menor vetor que conecta a origem do sistema de coordenadas à linha [19].

$$\rho = x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \quad (11)$$

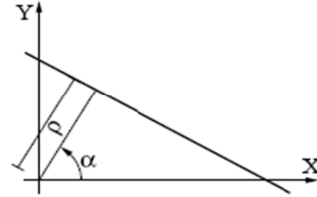


Figura 4. Parâmetros da linhas  $\rho$  e  $\alpha$ .

Definiu-se um sistema de coordenadas fixo ( $F$ ) e um móvel ( $M$ ), anexado ao robô, ambos ilustrados na Fig. 5. A origem do sistema móvel tem coordenadas  $(x_M^F, y_M^F)$  no sistema fixo.  $\theta_M^F$  representa a rotação do sistema móvel com respeito ao sistema fixo. Deve-se notar que existe uma relação estreita entre essas variáveis  $(x_M^F, y_M^F, \theta_M^F)$  e a pose do robô  $(x_t, y_t, \theta_t)$ , que é dada pela Equação (12).

$$x_t = x_M^F \quad y_t = y_M^F \quad \theta_t = \theta_M^F + \pi/2 \quad (12)$$

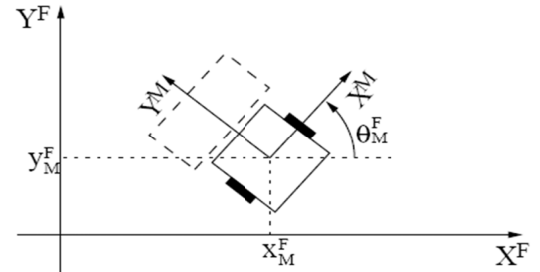


Figura 5. Sistemas de coordenadas Fixo ( $F$ ) e Móvel ( $M$ ).

Cada linha no chão é descrita por dois parâmetros estáticos ( $\rho^F, \alpha^F$ ). O mapa a ser produzido pelo processo de SLAM é composto de um conjunto desses pares de parâmetros. Então, o  $i^c$  vetor de coordenadas do  $i$ -ésimo marco que aparece na Equação (7) é dado pela Equação (13).

$${}^i c = \begin{bmatrix} {}^i \rho^F \\ {}^i \alpha^F \end{bmatrix} \quad (13)$$

A cada passo o robô captura uma imagem e identifica os parâmetros ( $\tilde{\rho}, \tilde{\alpha}$ ) das linhas detectadas. Usamos ( $\sim$ ) sobre a variável para indicar os valores medidos ao invés dos calculados.

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  $z$ , a ser usado na fase de atualização do algoritmo do FKE (Sistema 3) é definido pela Equação (14). 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. A cada passo o FKE calcula sua fase de atualização tantas vezes quanto forem o número de linhas detectadas na imagem.

$$z_t = \begin{bmatrix} \tilde{\rho}^M \\ \tilde{\alpha}^M \end{bmatrix} \quad (14)$$

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 15) e a Equação (11) em ambos os sistemas de coordenada (Equações 16 e 17).

$$\begin{cases} x^F = \cos(\theta_M^F) x^M - \sin(\theta_M^F) y^M + x_M^F \\ y^F = \sin(\theta_M^F) x^M + \cos(\theta_M^F) y^M + y_M^F \end{cases} \quad (15)$$

$${}^i \rho^F = x^F \cos({}^i \alpha^F) + y^F \sin({}^i \alpha^F) \quad (16)$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \quad (17)$$

Substituindo-se o Sistema (15) na Equação (16), fazendo as equivalências necessárias com a Equação (17) e substituindo algumas variáveis usando as Equações (12), obtêm-se os Sistemas (18) e (19), 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  $\alpha^M$  e usa-se o modelo que gerar o valor mais próximo ao valor medido  $\tilde{\alpha}^M$ .

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

$$\begin{cases} \rho^M = -{}^i \rho^F + x_t \cos({}^i \alpha^F) + y_t \sin({}^i \alpha^F) \\ \alpha^M = {}^i \alpha^F - \theta_t - \frac{\pi}{2} \end{cases} \quad (19)$$

O modelo do sensor é incorporado no FKE através da matriz  $H$  (Equação 6), dada pela Equação (20). As colunas finais de  $H$  são quase todas nulas, exceto para as colunas correspondentes ao marco no vetor de estados após o *matching* com a linha detectada na imagem.

$$H = \begin{pmatrix} -\cos({}^i \alpha^F) & -\sin({}^i \alpha^F) & 0 & \cdots & 1 & 0 & \cdots \\ 0 & 0 & -1 & \cdots & 0 & 1 & \cdots \end{pmatrix} \quad (20)$$

#### d) Correspondência ou Matching

Um aspecto crucial do algoritmo de SLAM é estabelecer uma correspondência entre a linha detectada na imagem e um dos marcos representado no vetor de estado. Para escolher o marco correto, primeiramente calculam-se os valores preditos de ( $\rho^F, \alpha^F$ ) usando os valores medidos de ( $\tilde{\rho}^M, \tilde{\alpha}^M$ ) e o modelo da Equação (18), se  $\tilde{\alpha}^M > 0$ , ou da Equação (19), se  $\tilde{\alpha}^M < 0$ . Então, esses valores preditos são comparados com cada um dos valores ( ${}^i \rho^F, {}^i \alpha^F$ ) no vetor de estado. Se a diferença entre o valor predito e o melhor ( ${}^i \rho^F, {}^i \alpha^F$ ) é suficientemente pequena, a correspondência foi encontrada. Se não, considera-se que a nova marca foi detectada e o tamanho do vetor de estado é aumentado.

### IV. PROCESSAMENTO DE IMAGEM

#### a) Detecção de Linhas

Devido à escolha das linhas do chão como marcos, a técnica adotada para identificá-las foi a transformada de Hough [30]. O propósito dessa técnica é encontrar instâncias imperfeitas de objetos dentro de certa classe de formas por processo de votação. Esse processo de votação é efetuado no espaço de parâmetros, no qual objetos candidatos são obtidos como máximo local em uma matriz de acumulação que é construída pelo algoritmo para a computação da transformada de Hough [3].

No nosso caso, a “forma” a ser detectada são linhas descritas pela Equação (11) e o espaço de parâmetros tem coordenadas ( $\rho, \alpha$ ). As imagens são capturadas em tons de cinza e convertidas para preto e branco usando detector de bordas Canny [5]. A Fig. 6.a mostra uma imagem típica do chão do ambiente, a Fig. 6.b mostra a imagem após a aplicação do detector Canny e a Fig. 6.c apresenta as linhas detectadas pela transformada de Hough.

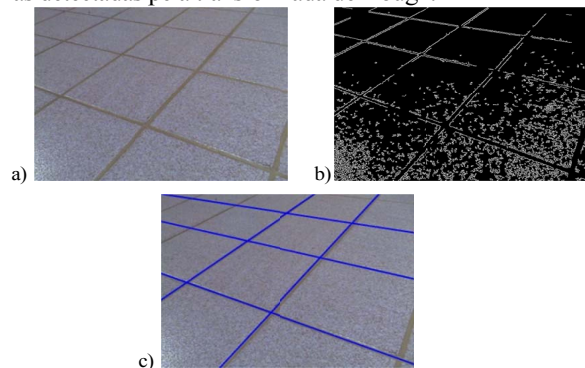


Figura 6. Processamento de Imagem.



### b) 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 ( $u, v$ ):

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (21)$$

O fator de escala  $s$  é determinado para cada ponto de tal forma que o valor do terceiro elemento do vetor seja sempre 1. A homografia pode ser calculada *off-line* usando um padrão contendo quatro ou mais pontos notáveis com coordenadas conhecidas (Fig. 7.a). 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 (21), obtém-se um sistema linear 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.

A fim de verificar a corretude da homografia encontrada, calculou-se o erro de re-projeção utilizando os pontos detectados na imagem e seus correspondentes no mundo. O erro médio calculado foi de  $e = 1.5 \text{ cm}$ . Para facilitar a interpretação deste valor, desenhou-se, sobre o padrão utilizado, circunferências cujo centro é o ponto do mundo mapeado na imagem utilizando a inversa da homografia calculada e o raio é o valor do erro médio (Fig. 7.b).

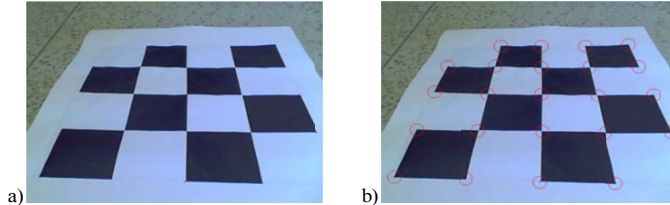


Figura 7. Padrão utilizado no cálculo da homografia.

### c) Ruído do Sensor

Como é apresentado na Fig. 3, a posição da câmera é de tal forma que o plano da imagem não é paralelo ao plano de chão. O resultando do efeito causado pela inclinação da câmara pode ser visto nas Figs 6 e 7. 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 ( $\rho$ ) da reta na imagem. Além disso, notou-se que a qualidade das linhas horizontais que aparecem na imagem é melhor que as verticais, o que nos permitiu inferir que a variância do ruído também é relacionada com o ângulo ( $\alpha$ ) da linha na imagem.

Levando em consideração o comentado anteriormente, a variância do ruído do sensor adotada neste trabalho é regida pela Equação (22). Os valores das constantes  $a$ ,  $b$  e  $c$  foram calculadas por experimentação e seus valores são:  $a = 0.004$ ,  $b = 0.3$  e  $c = 45$ .

$$\sigma(\rho, \alpha) = a + b \cdot \sin(\alpha) \cdot (\exp^{\frac{\rho}{c}} - 1) \quad (22)$$

Nesta equação, o termo  $[\exp^{\frac{\rho}{c}} - 1]$  representa a proporcionalidade à distância, e o termo  $[\sin(\alpha)]$ , a influência do ângulo. A Fig. 8 mostra o comportamento da função descrita pela da Equação (22) utilizando os valores de  $a$ ,  $b$  e  $c$  já apresentados e, considerando  $\rho$  em metros e  $\alpha$  em graus.

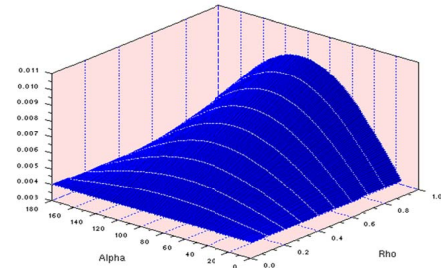


Figura 8. Comportamento da variancia do ruído do sensor.

## V. 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 (Fig. 3).

O local por onde o robô navegou durante a experimentação foi o Departamento de Física - DF da Universidade Federal do Rio Grande do Norte - UFRN. Este ambiente possui piso formado por cerâmicas quadradas com  $25\text{cm}$  de lado (Fig. 6). As interseções destas cerâmicas formam linhas horizontais e verticais que são utilizadas como marcos no algoritmo de SLAM. A Fig. 9 ilustra as características do DF/UFRN.

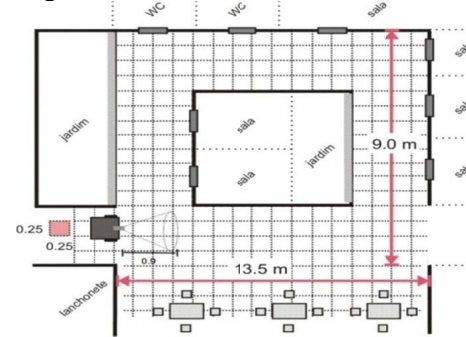


Figura 9. DF-UFRN (Este prédio pode ser visto em sistemas de mapeamento *online* utilizando coordenadas  $5^\circ 50' 29.50'' \text{ S } 35^\circ 11' 49.48'' \text{ W}$  ).

O robô executou uma trajetória quase retangular de circuito fechado dentro deste edifício e durante seu movimento 1962 imagens foram processadas. A câmera utilizada captura imagens  $640 \times 480$  (Fig. 6) e cada imagem é processada, na média, em  $180 \text{ ms}$ . A Fig. 10 mostra os gráficos do tempo de aquisição da imagem, tempo de processamento e o tempo total do sistema incluindo aquisição, processamento e cálculos do algoritmo de SLAM. Neste gráfico é possível observar nitidamente dois picos. Estes picos se deram devido à mudança de iluminação existente no ambiente quando o robô se aproximou das áreas de jardim. O tempo médio de aquisição foi de  $50 \text{ ms}$ , o tempo médio de processamento foi de  $125 \text{ ms}$  e o tempo médio total do sistema foi de  $180 \text{ ms}$ .

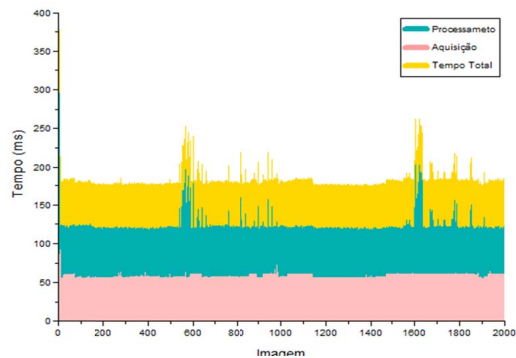


Figura 10. Tempos dos Sistema de Visão.

No tocante à homografia, a Fig. 7.a mostra o padrão que foi utilizado no início do experimento para calculá-la. A câmera foi posicionada de modo que foi possível ter um ângulo de visão da ordem de duas vezes o tamanho do robô. É importante lembrar que a posição da câmera é de tal forma que o plano da imagem não é paralelo ao plano de chão. A Equação (23) mostra a matriz de homografia utilizada.

$$\mathbf{A} = \begin{pmatrix} 0.1417 & 0.0009 & -49.2065 \\ 0.0073 & -0.0761 & 98.6323 \\ 0.0001 & 0.0029 & 1 \end{pmatrix} \quad (23)$$

A Fig. 11 mostra a trajetória calculada. Os pontos em azul correspondem à trajetória calculada apenas por odometria e os pontos em vermelho à trajetória calculada por SLAM.

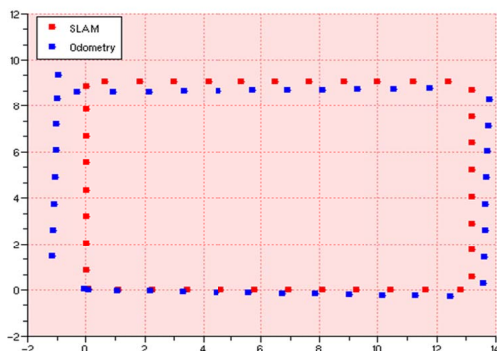


Figura 11. Trajetória percorrida pelo robô no DF-UFRN.

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 do o algoritmo de processamento de imagem.

Mesmo com 2% de erro, o sistema de visão se comportou bem no tocante a problemas de iluminação (Fig. 12) e também sobre problemas de sujeira no chão (Fig. 13).

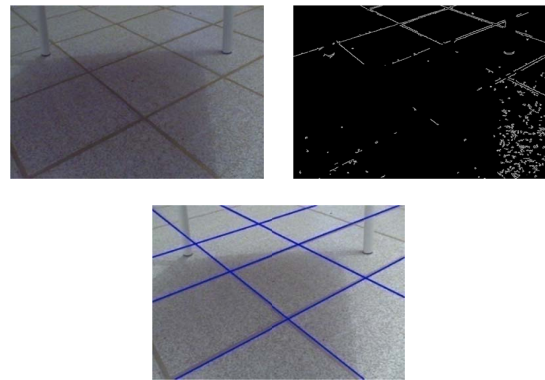


Figura 12. Imagem com mudança de iluminação.

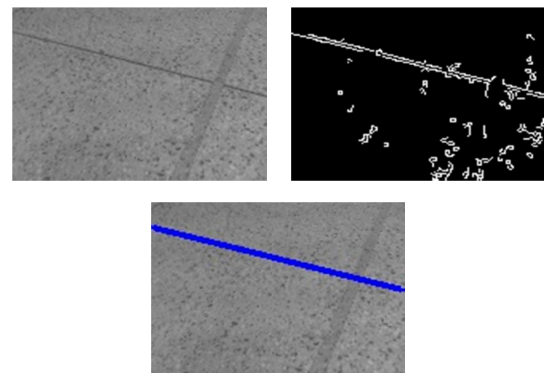


Figura 13. Imagem com uma falsa linha no piso do ambiente.

Voltando a comentar a trajetória realizada pelo robô, a distância entre a posição inicial e final, calculadas assumindo uma pose inicial  $(0, 0, 0^\circ)$ , foi de  $0.89\text{m}$ , usando somente odometria, e  $0.02\text{m}$ , usando SLAM; a distância final real, medida *in loco*, foi muito próxima da calculada por SLAM. A Fig. 14 mostra o instante do fechamento do laço.

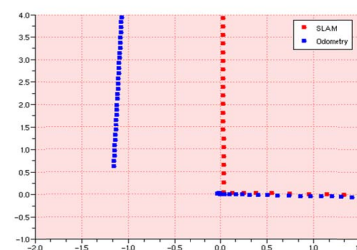


Figura 14. Trajetória: Zoom na região onde há o fechamento do circuito.

Ainda sobre trajetória é interessante apresentar o comportamento da componente  $y$  da pose do robô no instante do fechamento do circuito (Fig. 15). Percebe-se que há 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  $4\text{cm}$  passou para  $8\text{cm}$ . Isto acontece porque, pela primeira vez, o robô re-observa uma marca depois de ter se locomovido por um local onde ela não existe.

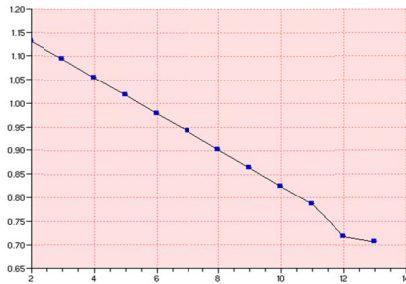


Figura 15. Pose do Robô: coordenada  $y$ .

A Fig. 16 mostra a variância ( $\sigma^2$ ) em  $\rho$  quando da primeira re-observação da linha que caracteriza o fechamento do circuito. Esta linha foi observada pela primeira vez no passo 23, e re-observada pela primeira vez no passo 1591. É possível verificar a atualização da variância apenas quando a linha é observada e re-observada. Antes da primeira re-observação (circuito fechado), o valor da variância de  $\rho$  era de  $4.6 \times 10^{-5}$  ( $\sigma = 0.68\text{ cm}$ ) e após a re-observação este valor foi reduzido para  $9.2 \times 10^{-6}$  ( $\sigma = 0.30\text{ cm}$ ). Finalmente, as variâncias da pose do robô são exibidas na Fig. 17.

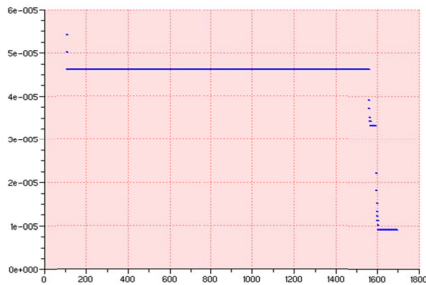


Figura 16. Variância em  $\rho$  da linha que caracteriza o fechamento de circuito.

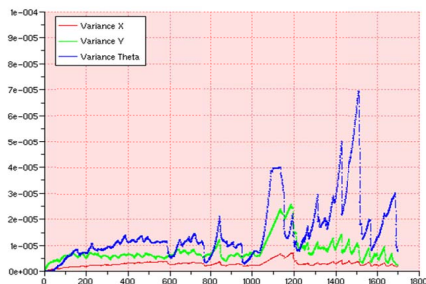


Figura 17. Comportamento das variâncias em  $x$ ,  $y$ ,  $\theta$ .

## VI. 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 mapear 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 diretamente 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 (shoppings, museus, hospitais, casas, aeroportos, etc.). 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 à baixa taxa de erro de correspondência e precisão na determinação da informação 3D das características.

Mesmo em um ambiente de laço fechado, como o apresentado nos resultados (Seção V), 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

- [1] S. Ahn, W. K. Chung, and S. Oh, "Construction of hybrid visual map for indoor SLAM", *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2007.
- [2] D. Amarasinghe, G. K. I. Mann and R. G. Gosine, "Landmark detection and localization for mobile robot applications: a multisensor approach", *Robotica Cambridge*, 2009.
- [3] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
- [4] Z. Chen, "A visual SLAM solution based on high level geometry knowledge and kalman filtering", *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1283-1286, 2006.
- [5] J. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 679 -698, 1986.
- [6] J. Choi, S. Ahn, M. Choi, and W. Chung, "Metric SLAM in home environment with visual objects and sonar features", *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2006.



- [7] J. Civera, A. J. Davison and J. M. Montiel, "Inverse depth parametrization for monocular SLAM", *IEEE Trans. on Robotics*, Vol. 24, pp. 932-945, 2008.
- [8] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos, "Mapping large loops with a single handheld camera" *Robotics: Science and Systems*, 2007.
- [9] M. N. Dailey and M. Parnichkun, "Simultaneous Localization and Mapping with stereo vision", *Int. Conference on Control, Automation, Robotics and Vision*, 2006.
- [10] A. J. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision", *15th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.
- [11] A. J. Davison and D. W. Murray, "Simultaneous Localization and Map-building using active vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 865-880, 2002.
- [12] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I", *IEEE Trans. on Robotics and Automation*, pp. 99-108, 2006.
- [13] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part II", *IEEE Trans. on Robotics and Automation*, pp. 108-117, 2006.
- [14] C. Estrada, J. Neira, and J. D. Tards, "Hierarchical SLAM: Realtime accurate mapping of large environments", *IEEE Trans. on Robotics*, pp. 588-596, 2005.
- [15] S. Frintrop, P. Jensfelt, and H. I. Christensen, "Attentional landmark selection for visual SLAM", *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2006.
- [16] S. Fu, C. Liu, L. Gao, and Y. Gai, "SLAM for mobile robots using laser range finder and monocular vision", *IEEE Int. Conference on Robotics and Automation*, 2007.
- [17] S. Fu and G. Yang, "Uncalibrated Monocular based Simultaneous Localization and Mapping for Indoor Autonomous Mobile Robot Navigation", *IEEE Int. Conference on Networking Sensing and Control*, pp. 663-668, 2009.
- [18] A. Gee, A., D. Chekhlov, A. Calway and W. Cuevas, "Discovering higher level structure in visual SLAM", *IEEE Trans. on Robotics*, Vol. 24, No. 5, pp. 980-990, 2008.
- [19] R. C. Gonzalez and R. E. Woodes, *Digital Image Processing*. Prentice Hall, 2007.
- [20] A. Hafez, S. Bhuvanagiri, M. Krishna and C. Jawahar, "On-line convex optimization based solution for mapping in vSLAM", *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2008.
- [21] D. C. Herath, K. R. S. Kodagoda, and G. Dissanayake, "Stereo vision based SLAM issues and solutions", *IEEE Int. Conference on Robotics and Automation*, 2007.
- [22] J. Jing Wu and H. Zhang, "Camera sensor model for visual SLAM", *Int. Conference on Computer and Robot Vision*, 2007.
- [23] I. K. Jung, "Simultaneous localization and mapping in 3D environments with stereovision" PhD thesis, Institut National Polytechnique - Toulouse - France, 2004.
- [24] N. M. Kwok, G. Dissanayake, and Q. P. Ha, "Bearing-only SLAM using a sprt based gaussian sum filter", *IEEE Int. Conference on Robotics and Automation*, 2005.
- [25] Y. Lee and J. Song, "Autonomous selection, registration, and recognition of objects for visual SLAM in indoor environments", *Int. Conference on Control, Automation and Systems*, 2007.
- [26] T. Lemaire and S. Lacroix, "Monocular SLAM as a graph of coalesced observations", *IEEE Int. Conference on Robotics and Automation*, pp. 2791-2796, 2007.
- [27] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons", *IEEE Trans. on Robotics and Automation*, Vol. 1, No. 3, pp. 376-382, 1991.
- [28] V. K. Mansinghka. *Towards visual SLAM in dynamic environments*. Springer, 2004.
- [29] D. Marzorati, M. Matteucci, D. Migliore and D. G. Sorrenti, "On the Use of Inverse Scaling in Monocular SLAM", *IEEE Int. Conference on Robotics and Automation*, pp. 2030-2036, 2009.
- [30] Hough, P. V. C, "Method and means for recognizing complex patterns", US Patent 3069654, Dec. 18, 1962.
- [31] R. Smith and P. Cheesman, "On the representation of spatial uncertainty", *The Int. Journal of Robotics Research*, Vol. 5, No. 4, pp. 56-68, 1987.
- [32] S. Thrun, W. Burgard, and Fox. D. *Probabilistic Robotics*. MIT Press, 2005.
- [33] M. Wongphati, N. Niparnan and A. Sudsang, "Bearing Only FastSLAM Using Vertical Line Information from an Omnidirectional Camera" *IEEE Int. Conference on Robotics and Biomimetics*, pp. 1188-1193, 2009.
- [34] M. Wu, F. Huang, L. Wang and J. Sun, "Cooperative Multi-Robot Monocular SLAM using Salient Landmarks", *Int. Asia Conference on Informatics in Control, Automation and Robotics*, pp. 151-155, 2009.



#### André Macêdo Santana:

Doutorado em Engenharia da Computação pela Universidade Federal do Rio Grande do Norte-UFRN, Natal, RN, Brasil (2011), Mestrado em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte-UFRN, Natal, RN, Brasil (2007), graduação em Ciência da Computação pela Universidade Federal do Piauí - UFPI, Teresina, PI, Brasil (2004). Atualmente é Professor Adjunto do Departamento de Informática e Estatística - DIE da UFPI. Atua principalmente nos seguintes temas: robótica móvel, visão computacional, filtragem estocástica e inteligência artificial.



#### Adelardo Adelino Dantas de Medeiros:

Doutorado em Robótica realizado no LAAS/CNRS pela Universidade Paul Sabatier em Toulouse, França (1997), mestrado em Engenharia Eletrônica e Computação pelo ITA - Instituto Tecnológico de Aeronáutica - em São José dos Campos, SP (1991) e graduação em Engenharia Elétrica pela UFRN - Universidade Federal do Rio Grande do Norte - em Natal RN (1988). Atualmente é professor associado da UFRN, ligado ao Departamento de Engenharia de Computação e Automação (DCA). Atua principalmente nos seguintes temas: robótica móvel, sistemas em tempo real, arquitetura de sistemas e automação industrial.