



# An Algorithm to Detect Lanemarks for Mobile Robots Navigation

Juliano Cesar Pimentel e Elder Moreira Hemerly\*

Instituto Tecnológico de Aeronáutica  
Departamento de Sistemas e Controle  
Divisão de Engenharia Eletrônica  
12228-901, São José dos Campos – SP – Brasil  
julianop@ele.ita.cta.br

\* Instituto Tecnológico de Aeronáutica  
Departamento de Sistemas e Controle  
Divisão de Engenharia Eletrônica  
12228-901, São José dos Campos – SP – Brasil  
hemerly@ele.ita.cta.br

**Abstract** — This paper presents an algorithm to detect lanemarks for mobile robots navigation. In contrast to the majority of other methods, it does not require human intervention neither an initial model for feature extraction and lanemarks detection. A recursive method for estimating parabola's coefficients is proposed, which model the way to be followed. Experiments with real images, both indoor and outdoor, even in the presence of non-ideal conditions as low luminosity, shadows and obstacle validate the procedure which presents good performance and robustness.

## I. INTRODUÇÃO

Muitos trabalhos têm surgido na literatura versando sobre sistemas de navegação autônoma baseados em visão. A informação visual eleva a autonomia e possibilita navegação em ambientes não-estruturados.

Algoritmos para extração de características e detecção de guias para navegação são necessários para manter o veículo no trajeto desejado sem desvio, informar sobre a presença de obstáculos e outros, além de mapear as informações do “mundo” em 3D para 2D, o que é de interesse para fins de controle de posição e velocidade do veículo.

Dificuldades surgem entretanto no processamento de imagens providas de ambientes não estruturados, devido à complexidade de uma modelagem eficaz das guias, que podem ser retas, curvas, contínuas ou tracejadas, estar sempre presentes ou eventualmente ausentes. Para estes casos são necessários algoritmos robustos que possam lidar com incertezas nas medidas.

O uso da informação visual tem sido amplamente explorada objetivando-se navegação e controle. Uma aplicação *indoor* é realizada em [1]: a navegação, com guiamento baseado em visão, requer o aprendizado de um modelo do ambiente. Para tanto, o veículo é guiado manualmente através de um ambiente e as imagens e dados para a situação de controle são gravados automaticamente. Então um procedimento *off-line* é executado para criar um modelo inicial do ambiente, e no estágio de navegação este modelo vai sendo atualizado a cada novo passeio pelo circuito. Linhas verticais são definidas no percurso e são tratadas pelo algoritmo de aprendizado como pontos-padrão. Na extração de características e identificação das linhas verticais é utilizada uma modificação da transformada generalizada de Hough.

Em [2], um método para detecção de guias utiliza um conjunto de características no domínio da frequência, calculado particionando-se a imagem em blocos de pixels de dimensão 8×8. Estas características refletem a quantidade de energia de borda diagonalmente dominante

que é contida em cada bloco, sendo combinadas via método bayesiano para se detectar as guias de interesse.

Em [3], apresenta-se a necessidade de um processamento *off-line* para extração de características e detecção de bordas, além da presença humana para guiar o veículo por um corredor. O sistema trabalha em duas fases: na primeira, o veículo é guiado por alguém através de um corredor. Durante este movimento, o sistema de visão extrai as características indesejadas, tais como portas ou escadas. A categoria e localização destas características são gravadas em um mapa. Na segunda fase, o sistema examina o mapa e seleciona uma sequência de guias que orientem o veículo ao destino no menor tempo.

Em [4], propõe-se compensação para dois importantes tipos de incerteza na detecção de guias: mudanças na direção das guias e incertezas nas medidas devido a cada nova imagem. Há, no entanto, a necessidade de modelos iniciais aproximados, antes que o rastreamento seja efetivamente inicializado. A correspondência entre o modelo e as guias é estabelecida manualmente por um agente humano (teleoperador), que posiciona os modelos para alinhá-los com a aparência das guias na imagem.

Em [5], apresenta-se um procedimento baseado em modelos para reconhecimento de formas em vias horizontais, não se detendo apenas na presença das guias, mas principalmente nas formas das vias. Entretanto, utiliza-se processamento estático das imagens para que haja classificação dos trechos da via em quatro modelos básicos, *straight*, *curve*, *curve-in*, *curve-out*. Devido ao processamento de imagens estáticas, requer-se o uso de câmeras de longo alcance, na faixa de 20m a 100m (ou mais, dependendo da velocidade do veículo).

Neste artigo propõe-se um algoritmo de processamento de imagens para detecção de guias para navegação de robôs móveis. Utilizando parábolas como modelo geométrico de aproximação para cada guia, e aplicando-se o método recursivo do filtro de Kalman na forma de informação (*KIF*) para atualizar a estimativa dos coeficientes destas parábolas, obteve-se um procedimento robusto e eficiente, bastante veloz na aquisição e processamento de várias imagens por segundo.

Para este procedimento, não há intervenção humana em nenhuma fase do processo, e o algoritmo realiza processamento de imagens *on-line*, sem a necessidade de modelos previamente estabelecidos. Obtiveram-se bons resultados em presença de situações desfavoráveis, tais como baixa luminosidade no ambiente, guias pouco visíveis, presença de objetos na região de navegação, tais como mesas e extintores nas aplicações *indoor* e sombras,

rachaduras no pavimento e carros nas aplicações externas.

Os pontos agrupados para a detecção das guias vêm diretamente do gradiente direcional aplicado à imagem em *gray-level*, sem a necessidade de comparação com nenhum modelo, ao contrário de [4].

Na seção 2, apresenta-se a estrutura do algoritmo. Na seção 3, uma visão mais detalhada do *KIF*, com uma expressão para atualização da matriz de informação que assegura sua propriedade positiva definida e, na seção 4, uma comparação entre o *KIF* e o Algoritmo de Mínimos Quadrados Recursivos Ponderados com Decaimento Exponencial (*WRLS*). Finalmente, na seção 5, resultados experimentais e discussões.

## II. ESTRUTURA DO ALGORITMO PARA DETECÇÃO DE GUIAS PARA NAVEGAÇÃO

O algoritmo identifica as guias através da atualização das estimativas dos coeficientes da parábola. O fluxograma com a estrutura e ordenação do algoritmo utilizado consta na figura 1, e há três estágios principais para obtenção do resultado, que é repetido para cada nova imagem adquirida:

- I. Gradiente Direcional da Imagem – remove da imagem a maioria das características indesejadas.
- II. Aplicação da Transformada de Hough [6], [7] e Interpolação por Mínimos Quadrados [8], [9] – possibilita a determinação da primeira estimativa para os coeficientes da parábola, e em casos de poucos pontos coletados, para as estimativas posteriores.
- III. Recursão – *KIF* [10] para atualizar recursivamente a estimativa dos coeficientes das parábolas.

No estágio (I), a imagem processada é aquela em níveis de cinza (*gray level* – 256 níveis de resolução). O gradiente direcional tira da imagem processada as características indesejadas, tais como linhas próximas da horizontal e aquelas próximas da vertical em função de que, na imagem real, as guias nunca estarão nesta situação. O *threshold* do gradiente pode ser alterado dependendo do ambiente, por exemplo, imagens *indoor* permitem o uso de um *threshold* menor, ao passo que imagens externas, dado o grande contraste entre luz, pavimento e objetos requerem um *threshold* maior. Há ainda a possibilidade de se utilizar *threshold* adaptativo, o que porém eleva o tempo de processamento.

No estágio (II), supõe-se que as guias sejam aproximadas por polinômios de segunda ordem, conforme a equação

$$x = a_1 y^2 + a_2 y + a_3 \quad (1)$$

onde  $x$  e  $y$  são as coordenadas horizontais e verticais, respectivamente, e  $a_1$ ,  $a_2$ ,  $a_3$  são os coeficientes da parábola. Os pontos finais de cada parábola são dados em função dos limites da janela de interesse. Um modelo de segunda ordem foi escolhido por fornecer, com boa aproximação, uma representação da forma real das guias, considerando as necessidades do problema para navegação de veículos autônomos. A transformada de Hough é utilizada a fim de se determinar, para cada guia,

a melhor reta que aproxima a nuvem de pixels vindo do gradiente direcional, e o critério de mínimos quadrados interpolará uma parábola na região definida por pixels vizinhos à reta. Esta estimativa é, em geral, muito boa, e serve para inicializar o algoritmo de recursão. Este estágio só é aplicado para a primeira imagem adquirida na observação ou para a situação onde o número de pixels na imagem se tornar muito pequeno (isto pode ocorrer, por exemplo, pelo fato de o robô se mover em direção a uma parede, onde não há guias a serem detectadas).

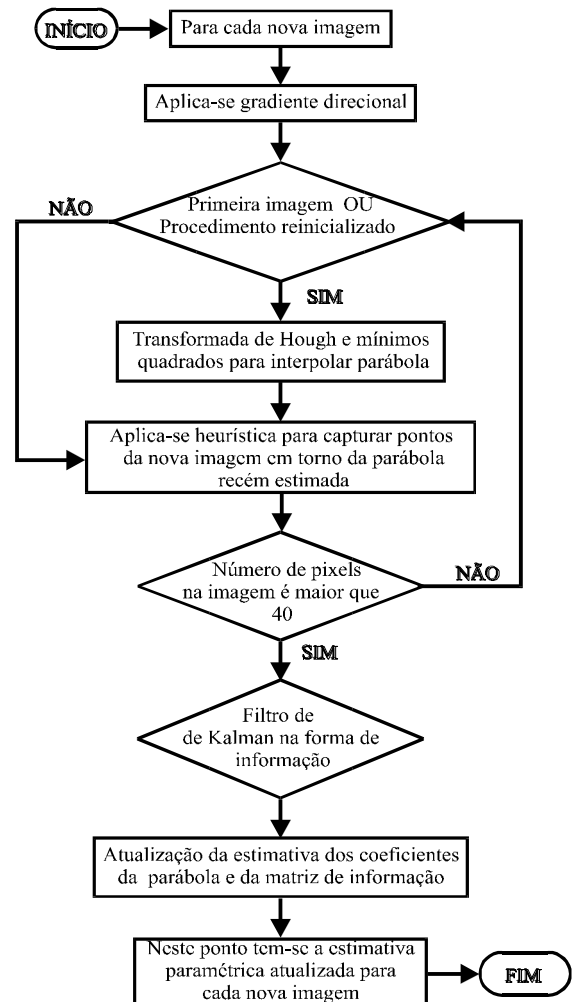


Fig.1. Fluxograma do algoritmo de processamento de imagens.

No estágio (III), a aplicação do algoritmo recursivo atualiza a estimativa dos coeficientes da parábola interpolada, a cada nova imagem. Para o *KIF*, apenas o conhecimento da imagem no instante atual e a estimativa imediatamente anterior são suficientes para determinar a estimativa atual desses coeficientes.

A figura 2 mostra o procedimento descrito nos estágios (I)-(III). A figura 2(a) mostra uma imagem de dimensão 320×240 pixels em *gray-level*. A figura 2(b) mostra o gradiente direcional de 2(a). Perceba-se que o cenário de 2(b) é bastante favorável para obtenção das guias. A figura 2(c) mostra a reta obtida através da transformada de Hough e 2(d) exibe a parábola interpolada pelo método dos mínimos quadrados, para a nuvem de pontos à direita. O resultado obtido pelo *KIF*, a partir desta estimativa inicial, será amplamente discutido a seguir.

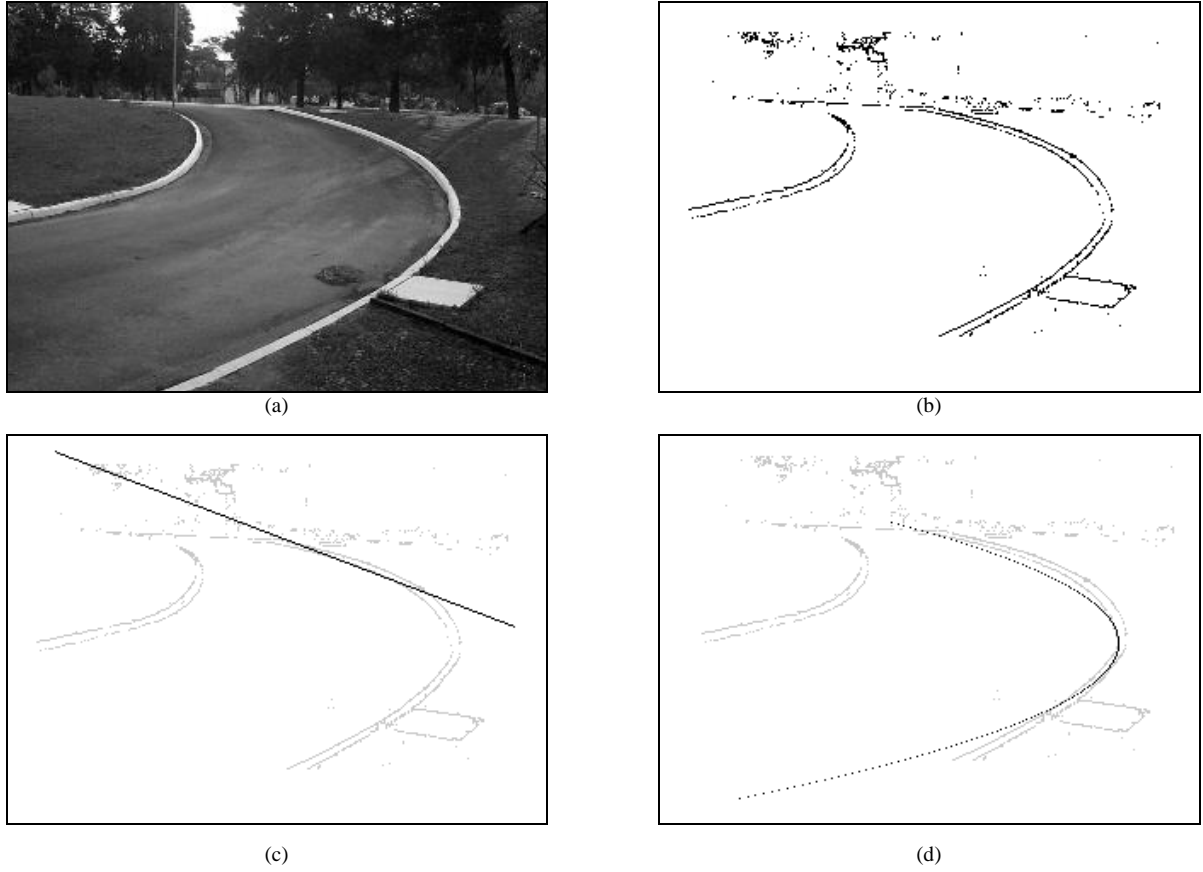


Fig. 2. Procedimento para geração da primeira estimativa dos coeficientes da parábola. A imagem original em *gray-level* está em (a). Em (b), os pixels resultantes do gradiente direcional. A reta resultante da transformada de Hough está em (c), e em (d), a parábola interpolada por mínimos quadrados.

### III. O KIF PARA ESTIMAÇÃO DOS COEFICIENTES QUE MODELAM AS GUIAS

Para o processamento de imagens *on-line* é necessário se recalculam  $a_1$ ,  $a_2$ ,  $a_3$  para cada nova imagem. Esta solução pode ser recursivamente formulada, como segue no caso do KIF, que corresponde ao estágio (III) na estrutura do algoritmo. Para cada conjunto de dados provenientes de uma nova imagem, o algoritmo estima novos coeficientes para as parábolas.

As equações do KIF são apresentadas a seguir, bem como uma forma positiva definida para a atualização da matriz de informação  $P_{t|t}^{-1}$ .

O sistema é descrito pelas equações

$$\hat{\theta}_{t+1} = \hat{\theta}_t + w_t \quad (2)$$

$$x_t = \phi_t^T \hat{\theta}_t + v_t \quad (3)$$

onde  $\hat{\theta}_t \in R^{3 \times 1}$  são os coeficientes da parábola,  $x_t \in R^{N \times 1}$ , e  $\phi_t \in R^{3 \times N}$  tais que

$$\hat{\theta}_{t-1} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad x_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \dots \\ x_{t,N} \end{bmatrix}, \quad \phi_t^T = \begin{bmatrix} y_{t,1}^2 & y_{t,1} & 1 \\ y_{t,2}^2 & y_{t,2} & 1 \\ \dots & \dots & \dots \\ y_{t,N}^2 & y_{t,N} & 1 \end{bmatrix}$$

e  $w_t$  e  $v_t$  são os ruídos de estado e medida, respectivamente, modelando as incertezas das estimativas dos coeficientes das parábolas e também da captura de cada nova imagem, com covariâncias  $Q_t$  e  $R_t$  dadas por

$$Q_t = E[w_t w_{t-k}^T] \quad (4)$$

$$R_t = E[v_t v_{t-k}^T] \quad (5)$$

Aplicando-se o lema da matriz inversa às bem conhecidas equações do filtro de Kalman discreto usual, origina-se o algoritmo recursivo alternativo conhecido como KIF. As equações decorrentes desta transformação usadas no filtro são

$$P_{t|t}^{-1} = P_{t|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (6)$$

$$P_{t+1|t}^{-1} = (I - B_t) A_t \quad (7)$$

onde

$$B_t = A_t (A_t + Q_t^{-1})^{-1} \quad (8)$$

e

$$A_t = P_{t|t}^{-1} \quad (9)$$

A matriz de ganho do filtro pode ser expressa em função de  $P_{t|t}^{-1}$ , isto é,

$$K_t = P_{t|t}^{-1} \phi_t R_t^{-1} \quad (10)$$

e as estimativas paramétricas são obtidas através de relações indiretas, segundo as equações

$$\hat{\theta}_{t+1|t} = (I - B_t) \hat{\theta}_{t|t} \quad (11)$$

e

$$\hat{\theta}_{t|t} = \hat{\theta}_{t|t-1} + \phi_t R_t^{-1} x_t \quad (12)$$

O conjunto de equações de (6) a (12) constituem as equações do *KIF*, que são algebricamente equivalentes às equações do filtro de Kalman usual. Contudo, nesta aplicação, a dimensão do vetor de leitura é elevada, e o filtro de Kalman usual necessitaria inverter uma matriz desta dimensão. Por outro lado, o *KIF* requer a inversão de matriz  $3 \times 3$ . Logo, apresenta grande simplicidade computacional e robustez à propagação de erro.

Uma melhoria adicional consiste na obtenção de uma expressão para a atualizar a matriz de covariância associada ao *KIF*, que preserva a propriedade positiva definida.

Apresenta-se, a seguir, uma maneira de atualizar a matriz  $P_{t+1|t}^{-1}$ , garantindo que ela seja positiva definida.

Seja a equação (6), e considere-se as equações (7), (8) e (9) defasadas de um instante no tempo. Tem-se portanto,

$$P_{t|t-1}^{-1} = (I - B_{t-1}) A_{t-1} \quad (13)$$

$$B_{t-1} = A_{t-1} (A_{t-1} + Q_{t-1}^{-1})^{-1} \quad (14)$$

$$A_{t-1} = P_{t-1|t-1}^{-1} \quad (15)$$

Substituindo-se (15) em (14) e o resultado em (13) segue-se

$$P_{t|t-1}^{-1} = P_{t-1|t-1}^{-1} - P_{t-1|t-1}^{-1} \times (P_{t-1|t-1}^{-1} + Q_{t-1}^{-1})^{-1} P_{t-1|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (16)$$

Seja agora

$$K_t = P_{t-1|t-1}^{-1} (P_{t-1|t-1}^{-1} + Q_{t-1}^{-1})^{-1} \quad (17)$$

Logo, de (16) tem-se

$$P_{t|t-1}^{-1} = P_{t-1|t-1}^{-1} - K_t P_{t-1|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (18)$$

Para um passo à frente em (18), obtém-se a recursão para  $P_{t+1|t}^{-1}$ .

Analisando-se (18), vê-se que a parcela  $\phi_t R_t^{-1} \phi_t^T$  é positiva definida, bastando então encontrar uma relação que garanta que  $P_{t-1|t-1}^{-1} - K_t P_{t-1|t-1}^{-1}$  seja positiva definida.

Para tanto, escreve-se

$$P_{t-1|t-1}^{-1} - K_t P_{t-1|t-1}^{-1} = (I - K_t) P_{t-1|t-1}^{-1} (I - K_t)^T + M_t \quad (19)$$

onde  $M_t \in R^{3 \times 3}$  é uma matriz a ser determinada. Desenvolvendo-se o lado esquerdo de (19), verifica-se que para que a igualdade seja verdadeira, deve-se ter

$$-P_{t-1|t-1}^{-1} K_t^T + K_t P_{t-1|t-1}^{-1} K_t^T + M_t = 0 \quad (20)$$

A equação (20) pode ser reescrita como

$$K_t [-K_t^{-1} P_{t-1|t-1}^{-1} + P_{t-1|t-1}^{-1} + K_t^{-1} M_t (K_t^{-1})^T] K_t^T = 0 \quad (21)$$

e de (17) resulta

$$K_t^{-1} = (P_{t-1|t-1}^{-1} + Q_{t-1}^{-1}) P_{t-1|t-1} \quad (22)$$

Agora, portanto, substituindo-se (22) em (21), verifica-se que a igualdade é satisfeita para

$$M_t = K_t Q_{t-1}^{-1} K_t^T \quad (23)$$

donde finalmente a expressão positiva definida para atualização da matriz de covariância no *KIF*, isto é,

$$P_{t|t}^{-1} = (I - K_t) P_{t-1|t-1}^{-1} (I - K_t)^T + K_t Q_{t-1}^{-1} K_t^T + \phi_t R_t^{-1} \phi_t^T \quad (24)$$

#### IV. COMPARAÇÃO ENTRE O *KIF* E O *WRLS*

O *WRLS* efetua um compromisso entre a importância de novos dados coletados em relação a dados passados na estimativa. O algoritmo preserva todas as informações passadas, representadas pela estimativa dos coeficientes e pela covariância estimada do estado anterior. Uma ponderação exponencial é usada para enfatizar o efeito dos dados atuais em relação aos dados passados.

O estimador de mínimos quadrados recursivo com ponderação exponencial satisfaz às equações

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_{t+1} (y_t - x_{t+1}^T \hat{\theta}_t) \quad (25)$$

onde

$$K_{t+1} = P_t \phi_{t+1} (\lambda I + \phi_{t+1}^T P_t \phi_{t+1})^{-1} \quad (26)$$

e  $P_t$  é computado recursivamente conforme

$$P_{t+1} = \lambda^{-1} (I - K_{t+1} x_{t+1}^T) P_t (I - K_{t+1} x_{t+1}^T)^T + \lambda K_{t+1} K_{t+1}^T \quad (27)$$

onde  $0 < \lambda < 1$ . Note-se que, se  $\lambda = I$ , resulta o método mínimos quadrados recursivo convencional.

As equações (25), (26) e (27) representam uma forma alternativa àquelas apresentadas em [4] e [13] e, dado que garantem que a matriz de covariância  $P_{t+1}$  seja sempre positiva definida, serão utilizadas neste trabalho para efeito de comparação com o *KIF*.

Embora o *WRLS* apresente resultados satisfatórios, a maior simplicidade computacional e robustez à propagação de erro do *KIF* motivaram sua proposição como alternativa viável ao filtro de Kalman usual e até mesmo ao *WRLS*. Outro fator positivo para o uso do *KIF*

em detrimento dos outros métodos está no fato de que não há a necessidade de inversão de matrizes de grande dimensão, o que reduz o risco de mal-condicionamento de matrizes e erro numérico, além de ser um procedimento mais rápido e apresentar bons resultados experimentais. Já no caso da implementação do *WRLS*, em sua forma original, há a necessidade de inversão de matrizes de ordens elevadas, sendo que a ordem da matriz varia de acordo com a imagem obtida a cada instante. Isto pode ser contornado pelo uso do *Square Root Filter (SRF)* [10], [11] e [12], que evita a inversão matricial, tem bom desempenho e é matematicamente equivalente a (25), (26) e (27). Contudo, o *SRF* possui codificação mais complexa e necessita tratar todas as leituras individualmente.

No caso do *WRLS*, o único parâmetro a ser ajustado é  $\lambda$ . Valores experimentais mostraram um  $\lambda$  adequado igual a 0.70.

Para o *KIF*, apenas as covariâncias dos ruídos,  $Q_i$  e  $R_i$ , dadas pelas equações (4) e (5), devem ser ajustadas inicialmente. Testes demonstraram a robustez a estas covariâncias, pois variou-se suas inicializações da ordem de dez vezes, e não houve degradação na estimativa dos coeficientes das parábolas.

## V. RESULTADOS EXPERIMENTAIS E DISCUSSÕES

A placa de aquisição de imagens utilizada no experimento é a Intel® Smart Video Recorder III, com processador Brooktree BT848. Foi utilizado um conjunto câmera CCD e receptor modelo TARGA® IX, *wireless*, em sistema NTSC, transmitindo na frequência de 2.4 GHz. O sistema de transmissão e recepção pode atingir até 200m, sem a presença de obstáculos. As imagens adquiridas são de dimensão 320×240 pixels. O algoritmo foi executado num PENTIUM® II, 450 MHz, com 128 Mbytes de memória RAM, em sistema operacional LINUX, distribuição RED HAT 5.2, versão de Kernel 2.0.36. O *driver* utilizado na captura das imagens está disponível em <http://hydra.dis.uniroma1.it/iocchi/bt848>.

A tabela 1 apresenta uma comparação entre os tempos de processamento para os possíveis métodos de recursão utilizados no algoritmo.

Os valores de tempo foram obtidos para uma imagem padrão, e o número de pixels máximo para definição de cada guia foi de 150 pontos. Obviamente, o tempo de processamento varia para cada nova imagem, porém a tabela 1 é bastante ilustrativa para demonstrar o desempenho do algoritmo dado que, para diversas situações, os valores na tabela podem ser considerados médios.

Tabela 1. Comparação entre métodos recursivos.

Método Recursivo	Tempo de Processamento (ms)
Transformada de Hough + Mínimos Quadrados	122.2
<i>KIF</i>	8.2
<i>WRLS</i>	22.2

O processamento mais básico, que é comum a todos os métodos da tabela 1, é o de captura e obtenção da imagem em *gray-level*, e a taxa de aquisição de imagens é de 30.0 *fps* (quadros por segundo), o que resulta em 33.3 *ms* para cada nova imagem em *gray-level*. Em seguida, o gradiente direcional, que também é necessário, requer 11.1 *ms*, baixando para 22.5 *fps* o número de quadros processados por segundo.

A tabela 1 justifica a proposta do algoritmo da figura 1, baseado no *KIF*. Percebe-se também que o processamento envolvendo a transformada de Hough e interpolação por mínimos quadrados apresenta custo computacional muito elevado, apesar de que a estimativa resultante seja muito precisa. Por isto é utilizado apenas nas condições de inicialização do algoritmo na figura 1.

As figuras 3, 4 e 5 apresentam, respectivamente, as imagens em *gray-level*, a imagem binarizada (gradiente direcional) e as guias detectadas. As imagens foram coletadas no corredor da Divisão de Engenharia Eletrônica do ITA.

As figuras 6, 7 e 8 apresentam a mesma sequência para imagens externas, obtidas na estrada de acesso ao prédio da Engenharia Eletrônica do ITA.

Com os testes efetuados tanto para aplicações *indoor* quanto externas, constatou-se a robustez do algoritmo, mesmo em presença de situações desfavoráveis como sombra, baixa luminosidade e irregularidades no pavimento da estrada. As figuras 6, 7 e 8 reforçam esta conclusão. Para aplicações externas, devido à presença de árvores, houve a formação de sombras na imagem, o que pode ser solucionado aplicando-se novos valores de *threshold* para o gradiente direcional.



Fig. 3. Imagem indoor em *gray-level*.

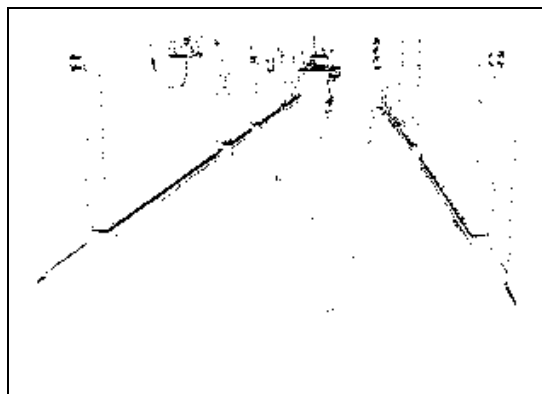


Fig. 4. Pixels resultantes do gradiente direcional.

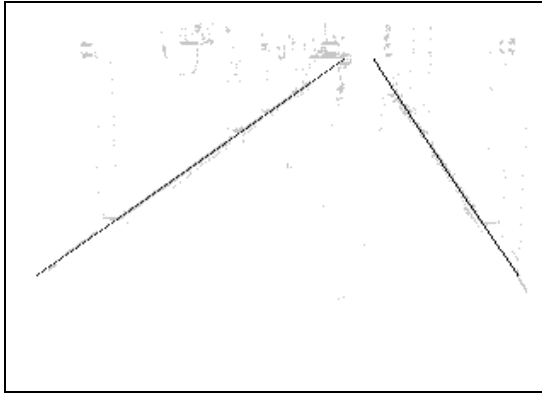


Fig. 5. Guias para navegação detectadas.

## VI. CONCLUSÕES

Um algoritmo foi proposto para se detectar guias para navegação de robôs móveis. O processamento envolveu três importantes estágios: a) extração de características da imagem, b) detecção de retas e interpolação de parábolas de interesse, e c) algoritmo recursivo para atualização destes coeficientes. Resultados experimentais validaram o procedimento, que mesmo em presença de situações desfavoráveis, obteve bons resultados. O sistema atingiu média de atualização dos coeficientes da parábola da ordem de 19 Hz.

Trabalhos futuros apontam na direção da obtenção da informação das coordenadas do mundo físico para o robô móvel através da informação visual, obtida por uma transformação de perspectiva, de modo a se efetuar o controle de posição e velocidade do robô. Seria também conveniente comparar o desempenho do algoritmo com o obtido por técnicas baseadas no fluxo óptico.



Fig. 6. Imagem externa em gray-level.

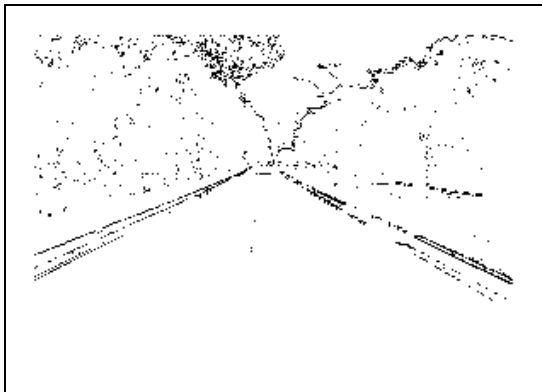


Fig. 7. Pixels resultantes do gradiente direcional.

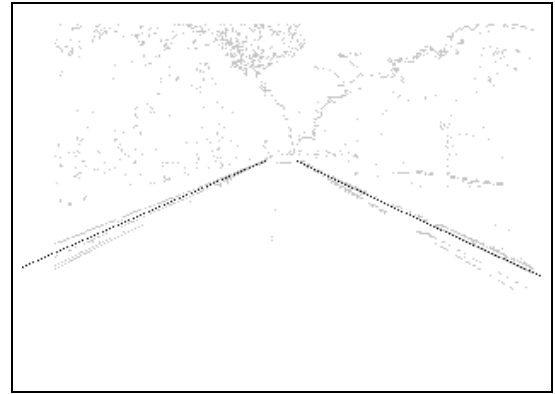


Fig. 8. Guias para navegação detectadas.

## VII. AGRADECIMENTOS

O primeiro autor agradece a CAPES pela concessão da bolsa de mestrado, e o segundo autor a FAPESP pelo auxílio financeiro, sob processo 99/05772-2.

## VIII. REFERÊNCIAS

- [1] G.Y. Chen e W.H. Tsai, "An incremental-learning-by-navigation approach to vision-based autonomous land vehicle guidance in indoor environments using vertical line information and multiweighted generalized Hough transform technique", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol.28, no.5, oct. 1998, pp. 740-748.
- [2] C. Kreucher e S. Lakshmanan, "LANA: a lane extraction algorithm that uses frequency domain features", *IEEE Transactions on Robotics and Automation*, vol.15, no.2, apr. 1999, pp. 343-350.
- [3] T. Kanbara, J. Miura e Y. Shirai, "Selection of efficient landmarks for an autonomous vehicle", in *International Conference on Intelligent Robots and Systems-Part 2*, jul. 1993, pp. 1332-1338.
- [4] H. Schneiderman e M. Nashman, "A discriminating feature tracker for vision-based autonomous driving", *IEEE Transactions on Robotics and Automation*, vol.10, no.6, dec. 1994, pp. 769-775.
- [5] Z. Hu e K. Uchimura, "Action-based road horizontal shape recognition", *SBA Controle & Automação*, vol.10, no.2, 1999, pp. 83-88.
- [6] K.S. Fu, R.C. Gonzalez e C.S.G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*, NY: McGraw-Hill Inc., 1987.
- [7] R. Jain, R. Kasturi e B.G. Schunck, *Machine Vision*, NY: McGraw-Hill Inc., 1995.
- [8] K. Kluge e C. Thorpe, Representation and recovery of road geometry in YARF, in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1990, pp. 114-119.
- [9] G.C. Goodwin e R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, NY: Academic Press, 1977.

- [1] G.C. Goodwin and R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, New York: Academic Press, 1977.
- [2] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [3] G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.
- [4] M. Morf and T. Kailath, "Square Root Algorithms for Least-Squares Estimation", *IEEE Transactions on Automatic Control*, Vol. AC-20, No.4, Aug., pp. 487-497, 1975.
- [5] G.C. Goodwin and K.S. Sin, *Adaptive Filtering Prediction and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [6] A.H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.
- [7] E.M. Hemerly, *Controle por Computador de Sistemas Dinâmicos*, São Paulo, SP: Edgard Blücher Ltda., segunda edição, 2000.
- [8] R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons Inc., 1992.