

DESENVOLVIMENTO DE UM SISTEMA PARA NAVEGAÇÃO DE ROBÔS MÓVEIS POR CAMINHOS EM PLANTAÇÕES

JODAS, Danilo Samuel
Universidade Estadual Paulista “Julio de Mesquita Filho” (UNESP)
danilojodas@gmail.com

MARRANGHELLO, Norian
Universidade Estadual Paulista “Julio de Mesquita Filho” (UNESP)
norian@ibilce.unesp.br

PEREIRA, Aledir Silveira
Universidade Estadual Paulista “Julio de Mesquita Filho” (UNESP)
aledir@ibilce.unesp.br

GUIDO, Rodrigo Capobianco
Universidade Estadual Paulista “Julio de Mesquita Filho” (UNESP)
guido@ieee.org

RESUMO: A utilização de robôs móveis na agricultura mostra-se importante em tarefas de cultivo e na aplicação de agrotóxicos em quantidades mínimas para reduzir a poluição do meio ambiente. Neste artigo apresentamos o desenvolvimento de um sistema para controlar a navegação de um robô móvel autônomo por caminhos em plantações. O controle da direção do robô é realizado com base em imagens das trilhas as quais, após um processamento prévio, para extração de características, são submetidas a uma máquina de vetores de suporte para a definição da rota a ser seguida. O objetivo do projeto no qual este trabalho se insere é o controle do robô em tempo real, para tanto, o sistema será embarcado em *hardware*. Neste trabalho, relata-se a implementação de uma máquina de vetores de suporte a qual apresentou uma precisão em torno de 93% da rota adequada.

PALAVRAS-CHAVE: robótica móvel, processamento de imagens, redes neurais artificiais, máquinas de vetores de suporte.

ABSTRACT: *The use of mobile robots in the agriculture turns out to be interesting in tasks of cultivation and application of pesticides in minute quantities to reduce environmental pollution. In this paper we present the development of a system to control an autonomous mobile robot navigation through tracks in plantations. Track images are used to control robot direction by preprocessing them to extract image features, and then submitting such characteristic features to a support vector machine to find out the most appropriate route. As the overall goal of the project to which this work is connected is the robot control in real time, the system will be embedded onto a hardware platform. However, in this paper we report the software implementation of a support vector machine, which so far presented around 93% accuracy in predicting the appropriate route.*

KEYWORDS: *mobile robotics, image processing, artificial neural network, support vector machine.*

1. INTRODUÇÃO

Nos últimos anos, houve um aumento significativo da utilização de robôs móveis em diversas áreas, tais como exploração espacial e operações de resgate. Esse aumento se deve à execução de atividades em locais de difícil acesso ou em situações que podem ocasionar riscos aos

seres humanos.

O surgimento de algoritmos inteligentes foi um fator que contribuiu significativamente para o avanço da robótica móvel devido a possibilidade de criação de agentes inteligentes que atuem de forma autônoma e confiável na execução de atividades para os quais foram projetados, sem a intervenção de um especialista humano.

Grande parte dos trabalhos desenvolvidos em robótica móvel abrange a utilização de redes neurais artificiais como sistemas inteligentes, as quais determinam uma saída baseando-se nos dados de entrada do ambiente externo, sendo estes capturados por algum tipo de sensor. Esses dados de saída são utilizados para realizar algum tipo de controle do agente móvel.

A combinação de algoritmos inteligentes com visão computacional tem proporcionado bons resultados em aplicações onde robôs móveis necessitam de um mapeamento do ambiente de atuação, com o intuito de evitar colisão com outros objetos ou determinar a direção para um local. Neste caso, as imagens são capturadas por uma câmera de vídeo e são utilizadas por algoritmos de pré-processamento e segmentação para extração das características relevantes da imagem, sendo estas utilizadas como dados de entrada pelos algoritmos inteligentes para a determinação de uma saída.

O surgimento de processadores com maior capacidade computacional também teve importância nesse cenário, já que a execução dessas tarefas necessitam de respostas em tempo real para garantir a viabilidade da implementação em hardware dos algoritmos computacionais de controle do robô móvel. Dispositivos FPGA (*Field Programmable Gate Array*) também são utilizados na implementação de robôs móveis devido à possibilidade de reconfiguração da sua estrutura interna de dispositivos lógicos, podendo isto pode melhorar significativamente o desempenho na execução das ati-

vidades do robô móvel devido a adaptação do hardware para um algoritmo específico.

2. Objetivos

O objetivo deste trabalho é apresentar um sistema de navegação baseado em algoritmos de processamento de imagens e máquinas de vetores de suporte, o qual será posteriormente implementado em *hardware*, para garantir a dirigibilidade de um robô móvel por caminhos de plantações. O sistema de navegação é utilizado para manter o robô na trilha da plantação e controlar sua direção baseando-se em imagens do terreno, as quais são utilizadas por algoritmos de processamento de imagens que são aplicados para a melhoria da qualidade da imagem e na extração das características do caminho. O caminho identificado é utilizado por uma máquina de vetores de suporte para a determinação do ângulo de direção do robô móvel.

O desenvolvimento do sistema de navegação é direcionado para a agricultura. Além desta atividade, outras serão aplicadas neste cenário, tais como a detecção de pragas em plantações e controle da aplicação de agrotóxicos para erradicá-las. Portanto, o controle de navegação é um elemento essencial para o sucesso das outras atividades.

3. Método desenvolvido

Na figura 1 é mostrado o diagrama de funcionamento do sistema de navegação proposto.

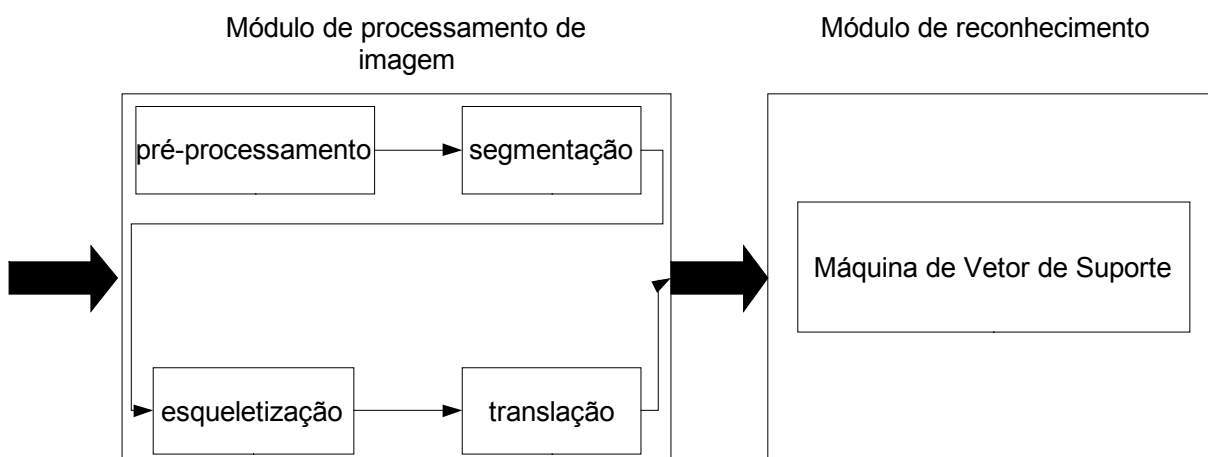


Figura 1: Diagrama de funcionamento do sistema de navegação.

No módulo de processamento de imagens são aplicadas técnicas para melhoria da qualidade da imagem e extração do caminho da plantação. No submódulo de pré-processamento é aplicado um filtro de suavização para a eliminação dos ruídos presentes na imagem. No submódulo de segmentação é feita a separação do caminho da área da plantação e a esqueletização do caminho identificado. No submódulo de translação é feito o deslocamento do caminho identificado para o centro da imagem com o objetivo de padronizar a representação dos dados para o módulo de reconhecimento.

O módulo de reconhecimento é composto por uma máquina de vetores de suporte, a qual recebe os *pixels* da imagem resultante do módulo de processamento de imagens e determina uma saída que representa o ângulo de direção para o padrão de pixels apresentado. O uso de uma máquina de vetores de suporte se deve ao baixo tempo de treinamento e a capacidade de generalização semelhante ao perceptron de múltiplas camadas.

O sistema de navegação foi desenvolvido em *software* para realização de testes iniciais. O desenvolvimento dos algo-

ritmos de processamento das imagens foi realizado na linguagem de programação C em conjunto com a biblioteca OpenCV (*Open Computer Vision Library*). O desenvolvimento do algoritmo da máquina de vetores de suporte também foi realizado na linguagem de programação C.

Para a realização dos testes, formou-se um banco de imagens composto por imagens de plantações de amendoim e soja, as quais foram adquiridas por uma câmera digital Kodak, modelo M531, sob diversas condições de iluminação. O banco é composto por 1186 imagens, sendo 570 imagens de amendoim e 616 imagens de soja. Cada imagem foi redimensionada à uma resolução de 300x225 pixels para diminuir o tempo de execução dos algoritmos de pré-processamento e segmentação.

As imagens do banco foram submetidas a ruídos, procedimento efetuado com as funções da biblioteca OpenCV, para a realização de testes com técnicas de suavização. A filtragem mediana foi utilizada para remover os ruídos presentes na imagem. Na figura 2 (b) é visualizado o resultado da filtragem mediana com uma máscara 7x7 em uma imagem submetida a ruído.



a)



b)

Figura 2: a) imagem submetida a ruídos b) resultado da aplicação da filtragem mediana.

A segmentação consiste em obter uma imagem binária onde a área verde, a qual corresponde à plantação, é representada pela cor branca e a área correspondente ao caminho é representada pela cor preta. O modelo de cor HSV foi utilizado para identificação da área verde correspondente à plantação. No HSV, a imagem é representada por três componentes:

- *Hue* (Matiz): Representando a cor pura;
- *Saturation* (Saturação): Representando o grau de diluição da cor pura por luz branca;
- *Value* (Valor): Representando o

contraste da imagem;

A vantagem em utilizar o modelo de cor HSV é a separação da informação de cor da informação de intensidade. Uma deficiência do uso desta abordagem pode surgir quando uma imagem possui sombras. Uma sombra é uma região escura na imagem formada pelo bloqueio da iluminação por um objeto. Sombras podem prejudicar a segmentação de uma imagem devido à semelhança com regiões escuras pertencentes às áreas de interesse ou por serem processadas como extensão de um objeto presente na imagem. Na figura 3 é ilustrada a incorreta identificação do caminho da plantação em uma imagem com sombras.

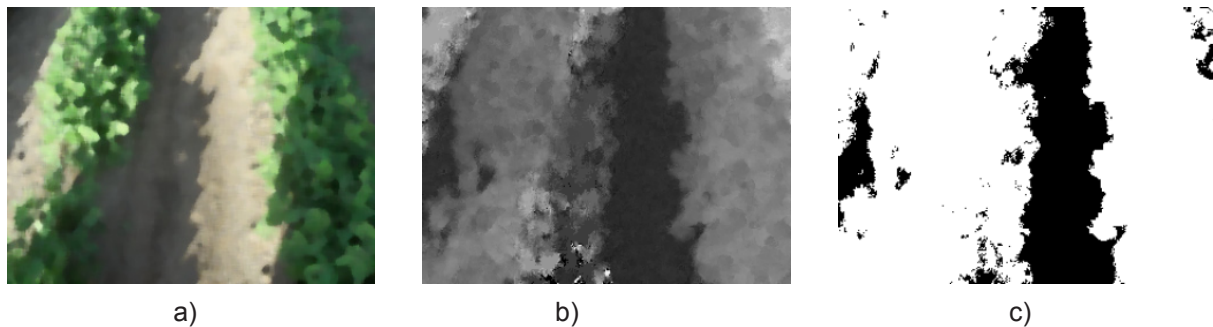


Figura 3: a) imagem com sombras b) componente matiz c) identificação incorreta do caminho da plantação.

É possível notar na figura 3 (c) a identificação parcial do caminho da plantação. Isto se deve ao fato da região da sombra ser processada como parte integrante da área da plantação, como pode ser observado na figura 3 (b), devido a semelhança com a trilha da plantação esquerda. Portanto, a região correspondente à sombra, quando representada no componente matiz, possui os mesmos valores do intervalo considerado para identificação da área da plantação. Diante desta dificuldade, é necessário que sombras sejam identificadas e desconsideradas no procedimento de extração do caminho. A abordagem utilizada neste trabalho é a obtenção de uma imagem com valores de pixels no espaço logarítmico, denominada imagem do espaço log-cromático, a qual é invariante a iluminação

e livre de sombras (FINLAYSON, 2004). O método pode ser escrito conforme a equação 1, a qual é utilizada para obter uma imagem no espaço log-cromático (XU, 2006).

$$inv = \cos(\theta) \cdot \ln\left(\frac{r}{g}\right) + \sin(\theta) \cdot \ln\left(\frac{b}{g}\right) \quad (1)$$

Na equação 1, [r,g,b] são os valores de cores da imagem no modelo RGB e θ é a direção de projeção da sombra. Esta equação segue um princípio que diz que uma imagem no modelo de cor RGB pode ser convertida em uma imagem em nível de cinza a qual representa apenas a propriedade de refletância (FINLAYSON, 2004). Na figura 4 (b) é exibido o resultado da aplicação da equação 1 sobre uma imagem com sombra.

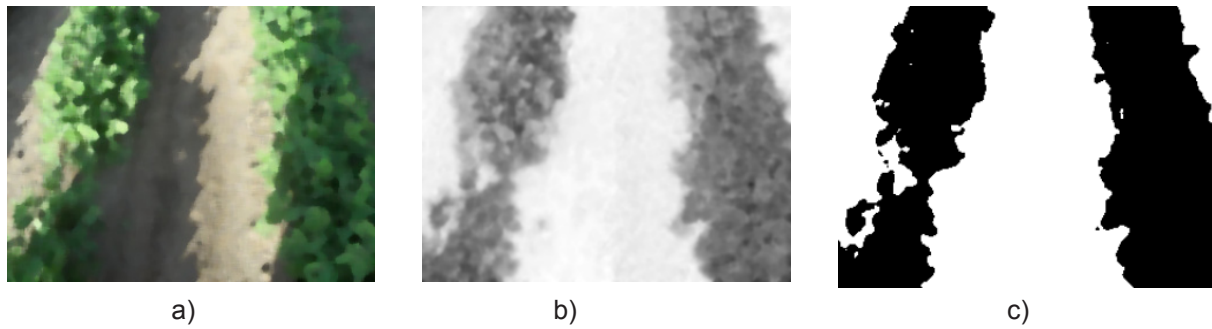


Figura 4: a) imagem com sombra b) imagem invariante a iluminação c) imagem invariante a iluminação binarizada.

É possível perceber na figura 4 (b) que os efeitos da sombra foram minimizados no caminho da plantação. A imagem invariante à sombras foi submetida ao processo de binarização, conforme é mostrado na figura 4 (c), onde foi possível identificar a área da plantação pela cor preta e o caminho pela cor branca. A imagem invariante a iluminação binarizada é utilizada juntamente com o componente matiz para gerar a imagem com o caminho extraído. A condição a ser satisfeita para que o caminho seja identificado corretamente é quando os valores de pixels no componente matiz estiverem entre 60° e 180° e os valores dos pixels correspondentes na imagem invariante a iluminação for igual à 0 (preto). Desta maneira, a região do caminho referente à sombra no componente matiz não é identificada como área da plantação, pois nesta situação a condição mencionada resulta em um valor lógico falso. Como a imagem invariante à sombras foi suficiente para a identificação

do caminho, não foi necessário eliminar as sombras da imagem original.

A operação morfológica de fechamento (GONZALEZ, 2000, p. 373) foi utilizada para suavizar as bordas do caminho identificado, eliminar pequenos buracos na imagem e eliminar pequenos istmos com a trilha adjacente, os quais são gerados devido a falhas na plantação. Este procedimento auxilia na melhor representação do esqueleto do caminho, minimizando os efeitos de proeminências que podem ser gerados devido a irregularidades nas bordas da imagem. Na figura 5 (b) é ilustrada a operação de fechamento sobre o resultado da segmentação do caminho da figura 5 (a) utilizando um elemento estruturante em forma de elipse de dimensões 20×20 . Foram utilizadas 20 iterações do fechamento sobre a imagem segmentada para obter mais suavização das bordas. A definição do elemento estruturante e do número de iterações foi feita empiricamente mediante vários testes.



Figura 5: . a) imagem segmentada b) resultado do fechamento.

Devido ao posicionamento da câmera no terreno, é possível que hajam na imagem capturada caminhos adjacentes a trilha principal. A eliminação desses caminhos reduz a quantidade de informações a ser transmitida para a máquina de vetores de suporte. Para a eliminação desses caminhos, foi utilizado o algoritmo de crescimento de regiões por agregação de *pixels* (GONZALEZ, 2000, p. 326). O algoritmo inicia com a seleção de *pixels* “sementes”, sendo que a condição de escolha são os *pixels* com menor nível de cinza, ou seja, em uma imagem binária são *pixels* que tenham a cor preta. Esse nível de cinza pertence aos caminhos identificados na imagem. Em uma imagem com três caminhos identificados, é possível selecionar três *pixels* sementes, um para cada caminho. O procedimento de crescimento de regiões é rea-

lizado e, simultaneamente a este processo, é feita a contagem de *pixels* pertencentes a cada região. Cada região é rotulada com um valor inteiro e ao final do processo é possível determinar qual região – caminho da imagem – possui a maior quantidade de *pixels*, sendo que este caminho é o que prevalecerá na imagem. Os outros caminhos são eliminados mediante a alteração dos seus valores de níveis de cinza para branco. Na figura 6 é ilustrado o resultado deste processo em uma imagem com dois caminhos. Na imagem mostrada na figura 6 (a), os caminhos são representados pela cor preta, sendo possível notar que o caminho da direita é o que possui a maior quantidade de *pixels* em relação ao caminho adjacente. Na imagem da figura 6 (b) o caminho adjacente foi eliminado.

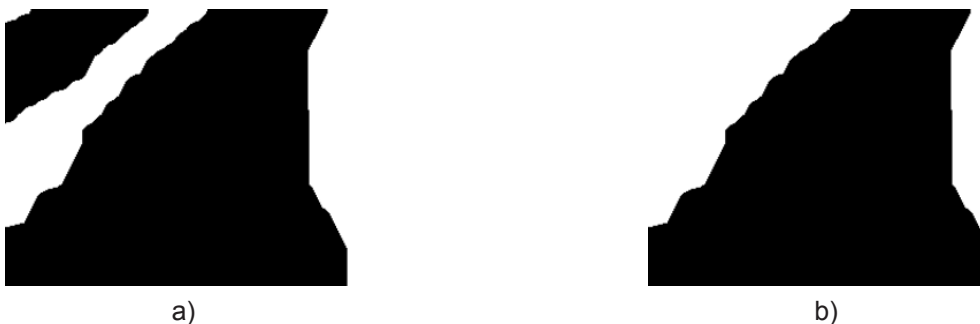


Figura 6: a) imagem com dois caminhos, representados pela cor preta b) Imagem com caminhos adjacentes eliminados

A próxima etapa foi o afinamento do caminho identificado (ZHANG e SUEN, 1984). O objetivo da obtenção do esqueleto da imagem é reduzir a quantidade de informações do caminho identificado a serem passadas para a máquina de vetores de suporte, reduzindo assim a sua complexidade. Na figura 7 (a) é ilustrado o processo de afinamento em uma imagem de plantação já segmentada.

O caminho extraído da imagem de uma plantação pode estar sujeito à deslocamentos devido ao posicionamento da câmera. Isto pode ocasionar dificuldades no reconhecimento da direção, pois cada deslocamento é classificado pela máquina de vetores de suporte em um ângulo diferen-

te, mesmo que tenham ângulos idênticos. Portanto, o reconhecimento da direção do esqueleto deve ser independente do seu deslocamento na imagem. Neste trabalho foi utilizado o algoritmo de invariância à translação (YÜCCER e OFLAZER, 1993), o qual é utilizado para calcular o centro do objeto na imagem e fazer com que ele coincida com o centro da imagem. O centro do objeto é obtido pela média das coordenadas (x,y) que contêm *pixels* brancos e é representada pelas equações 2 e 3.

$$x_{\alpha} = \frac{1}{\sum_{i=1}^M \sum_{j=1}^N f(x_i, y_j)} \sum_{i=1}^M \sum_{j=1}^N f(x_i, y_j) x_i \quad (2)$$

$$y_{\alpha} = \frac{1}{\sum_{i=1}^M \sum_{j=1}^N f(x_i, y_j)} \sum_{i=1}^M \sum_{j=1}^N f(x_i, y_j) y_j \quad (3)$$

onde $f(x, y)$ é a intensidade de cinza nas coordenadas (x, y) , M e N representam as quantidades de linhas e colunas da imagem, respectivamente. A função de mapeamento dos *pixels* do objeto para o centro da imagem é realizada mediante a equação 4.

$$f_i(x_i, y_i) = ((M/2) - X_{av}, (N/2) - Y_{av}) \quad (4)$$

onde M e N representam as quantidades de linhas e colunas da imagem, respectivamente. O procedimento consiste em calcular a diferença entre o centro (X_{av}, Y_{av}) do objeto – neste caso o esqueleto – e o centro (x, y) da imagem e utilizar esta diferença para deslocar os *pixels* do objeto para o centro da imagem. Na figura 7 (b) é ilustrado este procedimento.

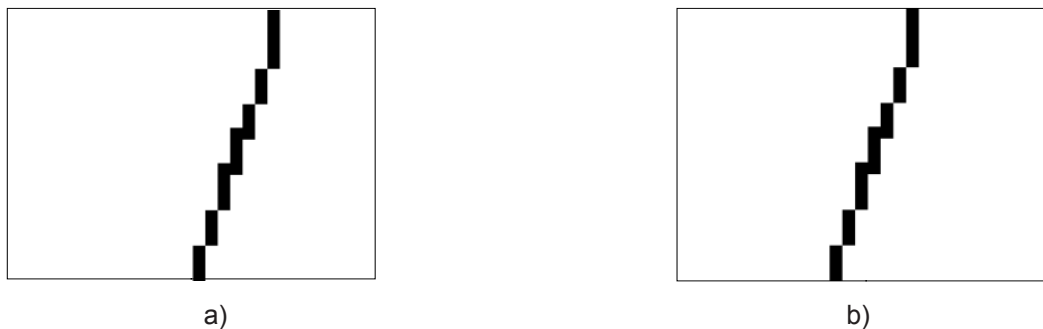


Figura 7: a) imagem com esqueleto deslocado do centro b) esqueleto centralizado

A imagem esqueletizada, com seu esqueleto centralizado por meio das equações 2, 3 e 4, foi transformada em um vetor de 690 elementos, os quais representam os *pixels* da imagem. O vetor de elementos é utilizado como entrada para uma SVM (*Support Vector Machine*), a qual é utilizada para determinar a direção de navegação do robô móvel. Foram estabelecidos 19 ângulos de direção, representados no intervalo de -45° a 45° , com intervalos de 5° de discretização. Máquinas de vetores de suporte permitem a classificação de padrões em duas classes separadas por um hiperplano de decisão. A construção de um hiperplano ótimo

como superfície de decisão é fundamental para que a separação entre os exemplos seja máxima, sendo esta tarefa realizada na etapa de treinamento da SVM. Foi utilizada uma máquina de vetores de suporte para classificar as imagens de entrada em um padrão de direção. A topologia da SVM é composta de 690 elementos de entrada, 71 elementos intermediários e 1 elemento de saída. O número de elementos intermediários foi determinado de acordo com a quantidade de exemplos utilizados no treinamento. Em cada elemento intermediário foram utilizadas duas funções de *kernel*, as quais são apresentadas na tabela 1.

Tabela 1. Núcleos do produto interno utilizados em SVM.

Tipo	Função do núcleo
Núcleo de função de base radial	$\exp\left(\frac{-1}{2\sigma^2} \ x - y\ ^2\right)$
<i>Hardware Friendly Kernel</i>	$2^{-d\ x-y\ _1}$

O primeiro *kernel* apresentado na tabela 1 é uma função gaussiana, onde x é o vetor de entrada, y é o valor do vetor de suporte e γ é o valor que determina a largura da curva gaussiana. A segunda função de *kernel* apresentada na tabela 1, denominada *Hardware Friendly Kernel*, foi desenvolvida para reduzir a complexidade de implementação em *hardware* de funções de *kernel* de SVM (ANGUITA, D. et al, 2006). O uso deste *kernel* é mais apropriado para implementação em *hardware* do que o *kernel* de função de base radial, pois evita o cálculo de divisões e exponenciais. Desta maneira, o *kernel Hardware Friendly* torna-se mais rápido no treinamento e no reconhecimento de padrões. O parâmetro γ é um valor inteiro definido como uma potência de dois, isto é, $\gamma = 2^p$, com $p = 0, 1, 2, \dots, Z$. A norma da distância entre os exemplos de entrada x e os vetores de suporte y é representada como $\|x - y\|$. O uso da norma ao invés da distância euclidiana evita o cálculo da raiz quadrada e da exponencial, tornando a implementação em *hardware* menos complexa.

No elemento de saída é apresentado um valor real o qual é o ângulo de direção classificado pela SVM.

4. Experimentos

Os testes foram realizados com imagens de amendoim e soja. Foram realizados três testes com Máquina de Vetores de Suporte, cada um utilizando uma abordagem diferente. As funções de *kernel* utilizadas nas SVMs foram a de base radial e o *hardware-friendly kernel*. O valor escolhido para o parâmetro γ , do *hardware-friendly kernel*, foi 0.0625, correspondente à 2^{-4} . A escolha do valor de γ foi feita empiricamente por meio de testes. Em cada teste, foram consideradas as seguintes informações:

- Quantidade de acertos exatos
- Quantidade de acertos aproximados
- Erros
- imagens não classificadas
- Percentual de acerto

Os acertos exatos indicam a quan-

tidade de imagens onde os ângulos de direção desejado e calculado pela SVM coincidiram exatamente. Os acertos aproximados indicam a quantidade de imagens onde os ângulos de direção desejado e calculado pela SVM tiveram uma diferença de 5° . Os erros representam a quantidade de imagens que tiveram uma diferença de 10° ou mais entre o ângulo desejado e o calculado pela SVM. O percentual de acerto no reconhecimento foi calculado considerando os acertos exatos e aproximados. Os acertos aproximados foram considerados devido à semelhança dos esqueletos que possuem ângulos com 5° de diferença, fazendo com que a SVM classifique uma imagem com a informação aproximada do ângulo. A tolerância pode ser aceitável se o tempo de processamento for baixo o suficiente para evitar que a próxima saída seja feita após uma longa distância percorrida pelo robô móvel, podendo isto minimizar a diferença de distância fora da rota.

A determinação dos parâmetros de treinamento da máquina de vetores de suporte foi feita utilizando a transformada de Hough (GONZALEZ, 2000, p. 308) e o método dos mínimos quadrados. O primeiro método consiste em transformar as retas da imagem representadas no espaço (x,y) para o espaço de parâmetros (ρ,θ) , onde ρ é a distância da reta em relação à coordenada $(0,0)$ e θ representa o ângulo de inclinação da reta. Portanto, o valor de θ é associado à imagem e utilizado como referência no aprendizado supervisionado da máquina de vetores de suporte. O segundo método consiste em obter os parâmetros da equação da reta que melhor se ajustam a uma amostra de pontos. A equação da reta é representada por $\alpha \cdot x + b$, onde α é o ângulo de inclinação da reta e b representa o deslocamento da reta em relação ao eixo y . Os pontos da amostra são representados pelas coordenadas dos *pixels* que compõem o esqueleto do caminho identificado e o valor obtido para o parâmetro α é utilizado no treinamento da máquina de vetores de suporte. Entretanto, a transformada de Hough e método dos mínimos quadrados foram utilizados apenas na etapa de treinamento e não são técnicas integrantes do sistema de navegação, pois em tempo de execução

a SVM, após ter sido treinada, deve ser capaz de reconhecer o ângulo de direção de novas imagens que não foram utilizadas na etapa de treinamento.

4.1. Primeiro teste

O primeiro teste foi realizado com 1186 imagens, todas pertencentes ao banco de imagens de plantações. Foram utilizadas 800 imagens para treinamento, sendo 400 imagens de amendoim e 400 de soja, e 386 imagens para verificação, sendo 170 imagens de amendoim e 216 imagens de

soja. Foram utilizadas 19 SVMs para classificação, sendo todas compostas por 690 elementos de entrada, 800 elementos intermediários e 1 elemento de saída. O número de elementos intermediários foi determinado de acordo com o número de exemplos utilizados no treinamento. Os exemplos de treinamento foram apresentados para cada SVM, inclusive a saída desejada. O ângulo de direção foi determinado pela SVM que apresentou saída 1. Na tabela 2 são apresentados os resultados obtidos neste primeiro teste para cada *kernel* usado no treinamento e reconhecimento da SVM.

Tabela 2. Estatísticas do reconhecimento do primeiro teste.

	Função de base radial	Hardware Friendly Kernel
Acertos exatos	260	262
Acertos aproximados	20	22
Erros	4	5
Imagens não classificadas	102	97
Percentual de acerto	72,5%	73,6%

Os resultados obtidos dos *kernels* de função de base radial e do *Hardware-Friendly* forem equivalentes. O fator prejudicial neste teste foi a quantidade de imagens que não foram classificadas. Essa situação ocorre quando a saída das 19 SVMs é -1, fato que ocorre devido a imagem apresentada não ter sido reconhecida em nenhum ângulo de direção. A quantidade insuficiente de imagens de treinamento para um determinado ângulo de direção foi um fator que contribuiu para que algumas imagens apresentadas às SVMs não pudessem ser classificadas.

4.2. Segundo teste

A implementação em *hardware* de 19 SVMs pode não ser possível devido a grande quantidade de elementos de processamento que pode exceder a capacidade de armazenamento do *hardware*. Cada SVM possui 690 elementos de entrada, 800 elementos intermediários e 1 elemento de saída, totalizando 1.491 elementos de processamento, sendo que a utilização de 19 SVMs totalizará 28.329 elementos

de processamento a serem implementados em *hardware*. Diante disso, foi estudado a possibilidade de implementação de apenas uma SVM que classifique todos os padrões de direção. Além de diminuir consideravelmente o espaço para síntese da SVM em *hardware*, o resultado da classificação foi melhor em relação ao teste anterior devido às imagens não classificadas terem sido reconhecidas como acertos exatos ou aproximados. Entretanto, a quantidade de erros também pode aumentar devido a saída da SVM estar sujeita a erros.

Neste teste foi utilizada apenas uma SVM com 690 elementos de entrada, 800 elementos intermediários e 1 elemento de saída, o qual possui saída entre -45 a 45, ao invés de 1 ou -1. Os resultados foram melhores em relação ao teste anterior, conforme pode ser visto na tabela 3. Apesar da quantidade de erros ter aumentado boa parte das imagens que não haviam sido classificadas anteriormente foram classificadas com a tolerância de 5° de diferença, ocasionando um aumento na quantidade de acertos aproximados e consequentemente no percentual de acerto.

Tabela 3. Estatísticas do reconhecimento para o segundo teste.

	Função de base radial	<i>Hardware Friendly Kernel</i>
Acertos exatos	238	216
Acertos aproximados	118	141
Erros	30	29
Imagens não classificadas	0	0
Percentual de acerto	92,2%	92,5%

4.3. Terceiro teste

O uso de 800 imagens de treinamento gera 800 vetores de suporte e 800 pesos para compor os parâmetros de reconhecimento da SVM. Sendo cada vetor de suporte representado por uma imagem de treinamento de 690 *pixels* de 1 byte, então são necessários 552.000 bytes ou 539 kbytes de memória para armazenamento de tais parâmetros, os quais podem não ser suportados pela memória interna de determinados modelos de dispositivos de FPGA. Além disso, a utilização de um grande número de imagens de treinamento pode di-

minuir a capacidade de generalização no reconhecimento pela SVM. Diante disso, neste teste foram utilizadas 71 imagens de treinamento e 1057 imagens de verificação, todas pertencentes ao banco de imagens de plantações. Desta maneira, a SVM ficou composta por 690 elementos de entrada, 71 elementos intermediários e 1 elemento de saída. Foram removidas 58 imagens do banco inadequadas para teste. A redução do número de imagens de treinamento permite, além de solucionar o problema de limitação do FPGA, aumentar a capacidade de generalização no reconhecimento da SVM. Os resultados podem ser vistos na tabela 4.

Tabela 4. Estatísticas do reconhecimento para o terceiro teste.

	Função de base radial	Hardware Friendly Kernel
Acertos exatos	776	810
Acertos aproximados	197	175
Erros	84	72
Imagens não classificadas	0	0
Percentual de acerto	92,05%	93,19%

Devido ao aumento no número de imagens de teste, o número de acertos aproximados e erros também aumentou. Entretanto, nota-se que houve uma pequena diferença no percentual de acerto em relação ao teste anterior, principalmente no percentual de acerto do segundo *kernel*. Além disso, os resultados obtidos por meio do *hardware-friendly kernel* neste teste foram melhores em relação ao *kernel* de função de base radial. O armazenamento em *hardware* diminuiu significativamente com a redução do número de imagens de treinamento, pois são necessários apenas 48 kbytes para armazenamento dos vetores de suporte e dos pesos.

4.4. Análise de desempenho

Os testes foram realizados em computador com 4 Gibabytes de memória RAM e um processador Intel Dual Core de 2200 Mhz. O tempo de execução de cada função de *kernel* para o treinamento e reconhecimento da SVM é apresentado no gráfico da figura 8. O *kernel* HFK (*Hardware Friendly Kernel*) apresentou um tempo de execução duas vezes mais rápido que o *kernel* RBF (*Radial Basis Function*) e, além disso, foi o *kernel* que apresentou melhores resultados no último teste. Independentemente da função de *kernel* utilizada, o uso de SVM mostra-se mais vantajoso em

relação ao tempo de treinamento do que redes neurais artificiais do tipo perceptron de múltiplas camadas, já que na SVM não há ciclos de retropropagação de erro como ocorre neste tipo de rede neural artificial.

Isto possibilita a implementação de treinamento on-line para agregar a classificação de novos padrões em diferentes campos de plantações.

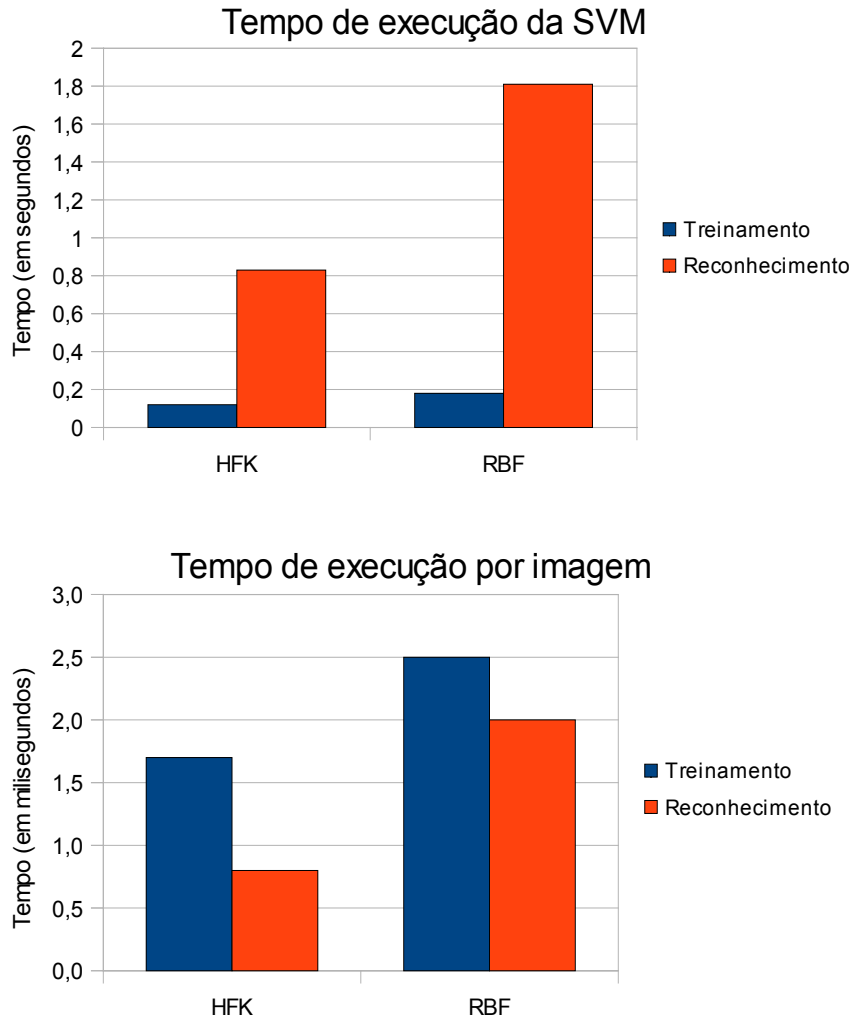


Figura 8: a) tempo de execução médio b) tempo de execução por imagem.

5. CONCLUSÕES

Neste trabalho foi apresentado um sistema de navegação para dirigibilidade de robôs móveis por caminhos em plantações, baseado em visão computacional e máquinas de vetores de suporte, além das etapas do desenvolvimento do sistema e os resultados obtidos por meio dos testes em *software*. O principal objetivo do trabalho foi obter um percentual de acerto superior a 90% utilizando uma pequena quantidade de informações da imagem, isto é, apenas o esqueleto do caminho da plantação.

A maior dificuldade enfrentada

nesta etapa do trabalho foi atingir um percentual de acerto exato no reconhecimento, fato este que ocorre devido a semelhança dos esqueletos que possuem valores próximos de ângulos. Entretanto, verificou-se que o reconhecimento para a maioria das imagens diferiu 5° em relação ao ângulo desejado, sendo que isto permitiu verificar a possibilidade de considerar o acerto aproximado por meio da análise da diferença da rota desejada. Verificou-se que uma diferença de 5° no reconhecimento causa 7 centímetros de desvio em relação à rota correta de navegação. Portanto, o tempo de execução do algoritmo deve ser rápido o

suficiente para que essa aproximação seja considerada e permitir taxas de atualização do ângulo de navegação a distâncias menores.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ANGUITA, D. et al. Feed-Forward Support Vector Machine Without Multipliers. **IEEE Transactions on Neural Networks**, volume 17, nº 5, pp. 1328-1331, setembro de 2006

FINLAYSON, G. D.; DREW, M. S.; CHENG, L. Intrinsic Images by Entropy Minimization. **European Conference on Computer Vision**, pp. 582-595, de 2004

GONZALEZ, R. C. ; WOODS, R. E. **Processamento de imagens digitais**. Edgar Blücher, São Paulo, 2000

HAYKIN, S. **Redes Neurais: Princípios e prática**. Bookman, Porto Alegre, 2001

HEARST, M. Support Vector Machines. **IEEE Intelligent Systems**, pp. 18-21, julho e agosto de 2008

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Brasport, Rio de Janeiro, 1999

XU, L.; QI, F.; JIANG, R. Shadow Removal from a Single Image. **Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications**, volume 2, pp. 1049-1054, outubro de 2006

YÜCEER, C. OFLAZER, K. A rotation, scaling, and translation invariant pattern classification system. **Journal of Pattern Recognition**, volume 26, nº 5, pp. 687-710, maio de 1993

ZHANG, T. Y.; SUEN, C. Y. A fast parallel algorithm for thinning digital patterns. **Communications of the ACM**, volume 27, nº 3, pp. 236-239, Março de 1984

Danilo Samuel Jodas possui graduação em Ciência da Computação pelo Centro Universitário do Norte Paulista e atualmente é aluno do Programa de Pós Graduação em Ciência da Computação do Instituto de Biociências, Letras e Exatas de São José do Rio Preto. Tem experiência na área de Ciência da Computação, com ênfase em Processamento Digital de Imagens, e atua nos seguintes temas: processamento de imagens, reconhecimento de padrões e computação reconfigurável. Atualmente trabalha em pesquisa direcionada ao reconhecimento de padrões para a robótica móvel.

Norian Marranghello possui graduação em Engenharia Eletrônica pela Pontifícia Universidade Católica do Rio Grande do Sul (1982), mestrado em Engenharia Elétrica pela Universidade Estadual de Campinas (1987), doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (1992), é pós-doutorado em Sistemas de Computação pela Universidade de Aarhus na Dinamarca (1998) e livre-docência em Sistemas Digitais pela Universidade Estadual Paulista (1998). Atualmente é Professor Titular da Universidade Estadual Paulista. Tem experiência nas áreas de Engenharia Elétrica e de Ciência da Computação, com ênfase em Sistemas Digitais, atuando principalmente nos seguintes temas: sistemas digitais integráveis, modelagem e simulação de sistemas, arquiteturas reconfiguráveis, redes de Petri e síntese de sistemas digitais.

Aledir Silveira Pereira possui graduação em Engenharia Elétrica pela Fundação Educacional de Barretos (1980), mestrado em Engenharia Elétrica pela Universidade de São Paulo (1987) e doutorado em Física Aplicada Computacional pela Universidade de São Paulo (1995). Atualmente é professor assistente doutor da Universidade Estadual Paulista Júlio de Mesquita Filho. Tem experiência na área de Ciência da Computação, com ênfase em Processamento Digital de Imagens e Sistemas de Controle e Automação, atuando principalmente nos seguintes temas: processamento de imagens, imagens médicas, controle e automação, ferramenta de software e ensino de computação.

Rodrigo Capobianco Guido possui graduação em Ciência da Computação pela UNESP - câmpus de São José do Rio Preto-SP e graduação em Engenharia de Computação pela Fundação Educacional de Votuporanga – FEV. Obteve o grau de Mestre em Engenharia Elétrica pela Faculdade de Engenharia Elétrica e de Computação da UNICAMP, de Doutor em Física Aplicada Computacional pelo Instituto de Física de São Carlos da USP e o título de Livre-docente na área de Processamento Digital de Sinais pelo Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos da USP. Atua na área de processamento de sinais, especialmente com base na transformada wavelet associada com técnicas inteligentes para aprendizado de máquina e reconhecimento de padrões.