

Luciano de Oliveira Neris

MÉTODO DE EXTRAÇÃO DA POSIÇÃO DE MÁQUINAS
AGRÍCOLAS POR VISÃO COMPUTACIONAL BASEADO
EM REDES PULSADAS E PONTO DE FUGA

Tese apresentada à Escola de Engenharia de
São Carlos da Universidade de São Paulo,
como parte dos requisitos para a obtenção do
Título de Doutor em Engenharia Elétrica.

Área de Concentração: Processamento de Si-
nais e Instrumentação

Orientador: Prof. Dr. Valentin Obac Roda

São Carlos
2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

N446m Neris, Luciano de Oliveira
Método de extração da posição de máquinas agrícolas por visão computacional, baseado em redes pulsadas e ponto de fuga / Luciano de Oliveira Neris ; orientador Valentin Obac Roda. -- São Carlos, 2008.

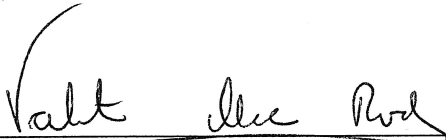
Tese (Doutorado-Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Processamento de Sinais e Instrumentação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2008.

1. Controle automático. 2. GPS - Sistema de Posicionamento Global. 3. Visão computacional. 4. Redes neurais pulsadas. 5. Ponto de fuga. I. Título.

FOLHA DE JULGAMENTO

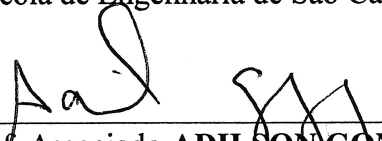
Candidato: Tecnólogo **LUCIANO DE OLIVEIRA NERIS**

Tese defendida e julgada em 07/04/2008 perante a Comissão Julgadora:




Prof. Associado **VALENTIN OBAC RODA (Orientador)**
(Escola de Engenharia de São Carlos/USP)

Aprovado



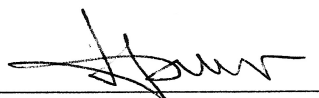
Prof. Associado **ADILSON GONZAGA**
(Escola de Engenharia de São Carlos/USP)

APROVADO



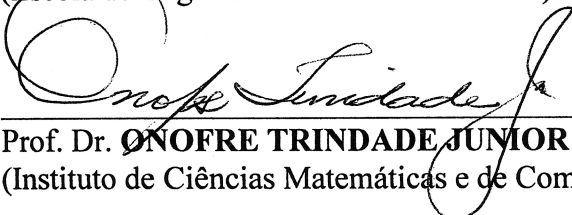
Dr. **ALVARO MACEDO DA SILVA**
(Empresa Brasileira de Pesquisa Agropecuária/EMBRAPA)

aprovado



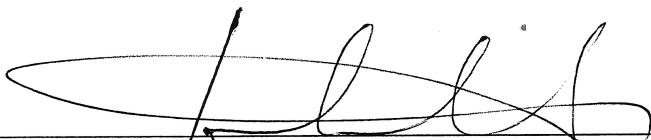
Prof. Dr. **MANOEL LUIS DE AGUIAR**
(Escola de Engenharia de São Carlos/USP)

Aprovado



Prof. Dr. **ONOFRE TRINDADE JUNIOR**
(Instituto de Ciências Matemáticas e de Computação/USP)

APROVADO



Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica e
Presidente da Comissão de Pós-Graduação

*A minha querida esposa, Vânia,
e ao meu eterno mestre Silvio.*

Agradecimentos

Ao meu orientador, Prof. Dr. Valentin Obac Roda, pela orientação, pelo apoio e por todo o conhecimento que me foi transmitido.

À Vânia, minha esposa, pelo carinho e por sua compreensão e apoio durante a elaboração deste trabalho.

Aos meus pais, irmãos e avós pelo incentivo, apoio e carinho.

À empresa AGX Tecnologia Ltda. por autorizar a utilização do sistema AGpilot[®] e aos colegas de trabalho pelo incentivo.

À Universidade de São Paulo pela estrutura e suporte.

A todos os companheiros do laboratório pelas discussões técnicas e amizade.

A todos os professores e funcionários do Departamento de Engenharia Elétrica da EESC/USP e a todos aqueles que, de uma forma ou de outra, contribuíram para a realização deste trabalho.

"Visão é a arte de enxergar coisas invisíveis"

Jonathan Swift

Resumo

NERIS, L. O. (2008). *Método de extração da posição de máquinas agrícolas por visão computacional baseado em redes pulsadas e ponto de fuga*. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

A redução de custos e a melhora do processo produtivo são essenciais para o aumento da rentabilidade e da produtividade das áreas agrícolas. O investimento em tecnologia se torna, portanto, fundamental em um mundo cada vez mais competitivo. Neste trabalho é apresentado o desenvolvimento de um método de extração da posição de máquinas agrícolas, em relação às linhas de cultivo, a partir do processamento de imagens fornecidas por uma câmera de vídeo colorida. A posição extraída é a informação básica utilizada em um sistema de direcionamento automático, permitindo determinar quais ações devam ser tomadas para manter a máquina em sua trajetória. O correto posicionamento da máquina sobre as linhas de cultivo melhora o processo de pulverização, ocasionando a redução de custos e o aumento da produtividade da área. O método proposto está embasado nos conceitos de ponto de fuga e busca antecipada. Essas técnicas permitiram simplificar o processamento das imagens e conseqüentemente a redução do tempo de processamento. Essas características, aliadas ao correto posicionamento da câmera, devem permitir que o método proposto possa ser utilizado no controle de máquinas agrícolas que operam em grandes velocidades como os pulverizadores.

Palavras-chave: controle automático; GPS; visão computacional; redes neurais pulsadas; ponto de fuga

Abstract

NERIS, L. O. (2008). *Method for position extraction of agricultural machine based on pulsed neural networks and vanishing point*. Ph.D. Thesis - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

Cost reduction and productive process improvement are essential to increase yield in agricultural areas. Investments in technology become, therefore, important in a competitive world. This work presents a novel approach for extracting agricultural machine position, with respect to crop rows, processing images captured by a color video camera. The correct machine positioning in crop rows can improve agricultural processes such as spraying, decreasing the costs and increasing the area yield. The proposed method is based on the look-ahead and vanishing points techniques. These techniques allow the reduction of the algorithm complexity and, therefore, the reduction of the processing time. These characteristics added to the camera position may allow the system to control agricultural machines that run at high speeds, such as sprayers.

Keywords: automatic control; GPS; computer vision; pulsed neural networks; vanishing point

Lista de Figuras

2.1	Representação simplificada dos neurônios.	36
2.2	Elementos ou nós de processamento e tipos de funções mais comuns (Adaptado de LIPPMANN (1988)).	37
2.3	Modelo do sistema visual (Adaptado de LINDBLAD e KINSER (2005)).	40
2.4	PCNN aplicada ao processamento de imagens digitais (Adaptado de NAZMY (2004)).	42
2.5	O neurônio PCNN (Adaptado de LINDBLAD e KINSER (2005)).	42
2.6	Imagem original e os resultados obtidos.	46
2.7	Trajetória do veículo.	48
2.8	Modelo de câmera.	49
2.9	Projeções de um objeto no plano de projeção.	50
2.10	Pontos de fuga (cultura de milho).	51
2.11	Projeção perspectiva de um ponto $P(x, y, z)$ em um plano de projeção.	53
3.1	Posicionamento das câmeras (Adaptado de BENSON <i>et al.</i> (2003b)).	57
3.2	Exemplo de imagem coletada (BENSON <i>et al.</i> , 2003b).	58
3.3	Câmera estereoscópica (KISE <i>et al.</i> , 2005).	60
3.4	Cálculo do ponto de navegação (Adaptado de BENSON <i>et al.</i> (2003b)).	61
3.5	Imagens processadas pelo sistema.	62
3.6	Sistema de coordenada Mundo-Câmera (ASTRAND e BAERVELDT, 2005).	63

4.1	Imagem de uma área cultivada (soja).	69
4.2	Diferentes localizações do ponto de fuga.	71
4.3	Imagens de culturas e os gráficos de soma.	72
4.4	Experimento de validação.	73
4.5	Diferentes localizações do ponto de fuga - câmera deslocada.	74
4.6	Diferentes localizações do ponto de fuga - câmera deslocada e rotacionada.	74
4.7	Experimentos realizados.	75
4.8	Imagem original e os resultados em diferentes iterações.	77
4.9	Imagens originais, imagens resultantes e os gráficos de soma - canal H do modelo HSI.	80
4.10	Imagens originais, imagens resultantes e os gráficos de soma - canal S do modelo HSI.	82
4.11	Imagens originais, imagens resultantes e os gráficos de soma - canal I do modelo HSI.	84
4.12	Imagens originais, imagens resultantes e os gráficos de soma - 2*G-R-B.	86
5.1	Plataforma de testes.	93
5.2	Diagrama de blocos da plataforma de testes.	93
5.3	Sensor de imagem da plataforma de testes.	94
5.4	Sensor de ângulo da plataforma de testes.	95
5.5	Motorreductor - direção.	97
5.6	Motorreductor - tração.	97
5.7	Diagrama de blocos do sistema desenvolvido.	98
5.8	Diagrama de blocos do sistema AGpilot®.	101
5.9	Interface do sistema AGpilot®.	102
5.10	Antena GPS.	102

5.11	Bússola.	103
5.12	Sensor de pressão.	104
5.13	Sensor de ângulo.	104
5.14	Bloco hidráulico.	105
6.1	Posicionamento do triciclo.	110
6.2	Interface de configuração do módulo Sensor de Ângulo.	111
6.3	Interface de configuração do módulo Controlador de Ângulo.	112
6.4	Interface de configuração do módulo Controlador de Rota.	113
6.5	Exemplo de quadro processado - teste inicial.	114
6.6	Teste 01 - área uniforme e fita imóvel.	114
6.7	Teste 02 - região esbranquiçada.	115
6.8	Teste 03 - fita esticada e virada.	115
6.9	Teste 04 - região esbranquiçada, buracos, palha e fita em movimento.	116
6.10	Teste 05 - fita em movimento e de dorso.	116
6.11	Exemplos de quadros normais.	118
6.12	Exemplos de quadros com palha e mancha escura.	119
6.13	Exemplos de quadros com buraco e manchas brancas.	120
6.14	Exemplos de quadros com buraco.	121
6.15	Exemplos de quadros com mancha branca.	122
A.1	Interface da ferramenta de teste desenvolvida.	139
C.1	Teste do módulo Controlador de Ângulo ($KP = 2,0$).	158
C.2	Erro observado - Teste 1.	159
C.3	Erro observado - Teste 2.	159
C.4	Erro observado - Teste 3.	160

Lista de Tabelas

2.1	Parâmetros da rede ICM.	46
4.1	Parâmetros da rede ICM e a quantidade de neurônios ativados.	78
4.2	Comparação dos resultados obtidos com a rede ICM.	87
C.1	Parâmetros do módulo Controlador de Rota.	159

Sumário

Resumo	ix
Abstract	xi
1 Introdução	25
1.1 Motivação e Relevância	28
1.2 Objetivos	30
1.3 Organização do Texto	31
2 Conceitos Fundamentais	33
2.1 Redes Neurais	34
2.2 Rede Neural de Pulso Acoplado	37
2.2.1 O Córtex Visual	39
2.2.2 Modelos Matemáticos do Córtex Visual	40
2.2.3 Aplicações	45
2.3 Controle Lateral	47
2.4 Modelo de Câmera e Pontos de Fuga	49
3 Métodos de Extração da Posição e de Orientação de Máquinas Agrícolas	55
3.1 Sistema de Orientação Baseado na Detecção de Borda	56
3.2 Detecção de Linhas Baseado em Visão Estéreooscópica	59

3.3	Detecção de Linhas Baseado na Transformada de Hough	62
3.4	Considerações Finais	64
4	Método de Extração da Posição de Máquinas Agrícolas Baseado no Ponto de Fuga e na Busca Antecipada	67
4.1	Extração da Posição	69
4.2	Posição da Câmera	71
4.3	Pré-Processamento	76
4.4	Conclusões	87
5	Plataforma de Testes	91
5.1	Triciclo	92
5.1.1	Software	98
5.2	AGpilot®	100
5.3	Considerações Finais	106
6	Resultados Experimentais	109
6.1	Materiais e Métodos	110
6.1.1	Calibração	111
6.2	Resultados	112
6.3	Considerações Finais	123
7	Conclusões	125
7.1	Contribuições	126
7.2	Trabalhos Futuros	128
	Referências	129
	Glossário	135

A	Implementação do Modelo ICM	139
B	Funções do Sistema Desenvolvido	145
B.1	Gerador PWM	145
B.2	Controlador PID	147
B.3	Controlador de Ângulo	148
B.4	Filtro	152
B.5	Controlador de Rota	153
C	Testes do Sistema AGpilot[®]	157
C.0.1	Resultados	157

Capítulo 1

Introdução

O homem sempre buscou formas de tornar mais fáceis todas as tarefas que realiza. A agricultura, por exemplo, foi desenvolvida para reduzir o tempo gasto na busca de alimentos realizada pelos antigos nômades. Atualmente, a mecanização agrícola permite aos agricultores obter elevados índices de produtividade e cultivar grandes áreas com menos recursos (BENSON *et al.* , 1998).

Visando a melhoria da qualidade da produção, a automação agrícola tem sido o objeto de pesquisas de várias instituições. A automação é algo aplicado a tarefas repetitivas, se encaixando perfeitamente às tarefas agrícolas. A automação agrícola pode variar desde um simples sistema de auxílio ao operador, como os sistemas de controle automático de altura de barras de pulverização, até a uma operação completamente autônoma (BENSON *et al.* , 2000).

As operações realizadas em campo, combinadas à robustez e segurança necessárias, dificultam a automação agrícola. O ambiente agrícola possui um numeroso conjunto de circunstâncias que difere muito daqueles encontrados em um ambiente controlado. Diversos problemas que dificultam a automação podem ser facilmente detectados. Entre eles podem-se citar (HAGUE *et al.* , 2000; BENSON *et al.* , 2000):

- Grandes áreas de cultivo - as áreas cultivadas atualmente são cada vez maiores exigindo, conseqüentemente, que os equipamentos agrícolas operem continuamente por longos períodos principalmente na época de plantio e de colheita;
- Superfícies irregulares - diversos tipos de relevo podem ser encontrados nas áreas agrícolas. Terraços, valas e até mesmo buracos feitos por animais silvestres podem estar presentes nessas áreas. Algumas formas de plantio também causam irregularidades na superfície do solo;
- Intempéries - as máquinas agrícolas estão sujeitas a diversas condições climáticas que afetam de forma substancial os dispositivos eletromecânicos e os dados de sensores. Tais condições incluem as altas ou baixas temperaturas, a chuva, a umidade e o vento. Além das condições climáticas citadas anteriormente, as máquinas agrícolas também estão sujeitas à lama, à poeira e aos defensivos químicos que, em sua maioria, são corrosivos. Os dispositivos eletrônicos devem ser projetados de forma a suportar essas condições para terem uma vida útil maior.

Um dos grandes desafios a ser superado na automação agrícola é a redução de custos. As novas tecnologias também são difíceis de serem aceitas pelos agricultores. Eles somente aceitarão a tecnologia que oferecer benefícios financeiros a um custo razoável. Desenvolver sistemas que possam ser utilizados em um grande número de aplicações a partir de componentes já existentes diminui em muito o custo e aumenta a sua aceitabilidade (JAHNS, 1997).

Diversos problemas relacionados à qualidade e à segurança na área de cultivo são normalmente causados pela fadiga dos operadores devido aos longos períodos de trabalho a que estão submetidos. Os operadores devem manipular indicadores e/ou monitores de plantio, de colheita, e de pulverização; controlar a velocidade e a direção das máquinas agrícolas; controlar a elevação das barras de pulverização ou das plataformas de colheita;

e manipular muitos outros controles que devem ser comandados em conjunto (BENSON *et al.* , 2000).

Uma das tarefas que pode perfeitamente ser automatizada e, conseqüentemente, trazer um aumento significativo da produtividade das áreas agrícolas é retirar do operador a tarefa de guiar as máquinas agrícolas. Essas máquinas são meios de produção altamente especializados, tornando a supervisão do processo produtivo como a principal tarefa dos operadores. Guiar tais veículos, apesar de não ser a principal função do operador exige grande parte de sua atenção influenciando diretamente na produtividade da área cultivada (CHOI *et al.* , 2002).

Atualmente, existem vários sistemas comerciais de direcionamento automático para máquinas agrícolas. A grande maioria desses sistemas utiliza o receptor GPS (*Global Positioning System*) como o principal sensor de navegação. A principal desvantagem desses sistemas é o alto custo ocasionado pelos elevados preços dos receptores GPS de maior precisão (BUICK, 2006; AGCO, 2008; JOHN DEERE, 2008; HEMISPHERE GPS, 2008).

Existe no mercado atual um grande número de produtos baseados na tecnologia de imageamento. Câmeras e filmadoras digitais, placas de captura e de digitalização de imagens e processadores de vídeo dedicados são alguns dos exemplos. Os preços desses dispositivos estão em declínio o que torna a tecnologia muito atraente para o desenvolvimento de sistemas baseados em visão computacional voltados à agricultura. O direcionamento automático de máquinas agrícolas é uma das aplicações que pode ser desenvolvida utilizando essa tecnologia.

A visão computacional permite obter informações não intrusivas (*contactless*) do mundo real, estendendo a habilidade dos robôs de operar em circunstâncias e ambientes que não podem ser precisamente controlados. O controle baseado em visão computacional tem sido alvo de intensivas pesquisas nos últimos anos. A determinação de atitude, a navegação, o posicionamento e a manipulação robótica são algumas das aplicações exis-

tentes (BARRETO *et al.* , 2002; PINTO e REID, 1998; PINTO *et al.* , 1999; LEEMANS e DESTAIN, 2006; BROGGI *et al.* , 2003).

Para a orientação de veículos existe uma importante vantagem da visão computacional em relação aos outros métodos destinados à mesma tarefa: a visão computacional pode fornecer a posição dos objetos na cena. Também é possível obter alguma informação dos objetos possibilitando, por exemplo, diferenciar a cultura das ervas daninhas (JAHNS, 1997; PERNOMIAN, 2002).

A universalidade e versatilidade dos sistemas baseados em visão computacional, o crescente desenvolvimento dos dispositivos de imagem e a boa relação custo-desempenho, fazem com que esses sistemas sejam aplicáveis em diversas etapas do processo de cultivo como a pulverização e a colheita (JAHNS, 1997; BENSON *et al.* , 2003a).

1.1 Motivação e Relevância

O setor agrícola brasileiro tem se destacado cada vez mais pelos recordes de produtividade. Muitos dos bons resultados são oriundos da mecanização do processo produtivo. Além do aumento da produtividade a mecanização permite reduzir os custos de plantio, os custos de colheita e a quantidade de insumos agrícolas utilizados (KEICHER e SEUFERT, 2000).

Eliminar do operador a tarefa de guiar a máquina agrícola aumenta a qualidade do processo produtivo e traz benefícios imediatos ao produtor. Tornar a tecnologia de direcionamento automático mais barata e confiável surge como uma necessidade e um desafio numa época tão competitiva.

As máquinas agrícolas, em alguns tratos culturais, podem operar em velocidades elevadas, dificultando a dirigibilidade e afetando a produtividade da área cultivada. A pulverização, por exemplo, é normalmente realizada em velocidades superiores a 15 *km/h*

(STOMBAUGH e SHEARER, 2001). Tal característica aliada às grandes dimensões dos modernos pulverizadores e à curta distância entre as linhas de cultivo afeta a qualidade da pulverização devido ao amassamento das plantas, às sobreposições e às falhas de aplicação. A incorreta pulverização reduz a produtividade das áreas cultivadas e eleva os custos de produção. O aumento da eficiência e a redução dos custos podem ser obtidos com a utilização de sistemas de direcionamento automático. No entanto, tais sistemas devem possuir elevada precisão e tempos de resposta pequenos, exigindo a utilização de sensores de posição precisos e capazes de fornecer dados a uma taxa elevada.

Diversas técnicas têm sido propostas ou desenvolvidas para guiar automaticamente as máquinas agrícolas. A popularização e o aumento da precisão dos modernos receptores GPS possibilitaram o desenvolvimento e a comercialização de diversos sistemas de direcionamento automático e assistido. Técnicas como fusão de sensores, redes neurais, e filtros de Kalman têm sido estudadas e aplicadas na solução de problemas relacionados à utilização de receptores GPS como a baixa precisão, a perda do sinal proveniente dos satélites de posicionamento e a detecção de obstáculos (HAGUE *et al.* , 2000; F.REID *et al.* , 2000; ZHANG e QIU, 2004; BENSON *et al.* , 2003a).

O principal problema dos sistemas de direcionamento automático que utilizam receptores GPS como o principal sensor de posição é a precisão do posicionamento. A qualidade dos dados de posição desses receptores depende do número de satélites captados, da existência de multicaminhamento (*multipath*) ocasionado pela proximidade de obstáculos como construções e árvores e é afetada pela atmosfera terrestre por influenciar na propagação do sinal eletromagnético (MONICO, 2000).

A precisão dos receptores GPS pode ser aprimorada com a utilização de métodos de correção da posição. Alguns receptores GPS podem aceitar correções de estações DGPS (*Differential Global Positioning System*), satélites WAAS (*Wide Area Augmentation System*), ou gerar internamente as correções com base em uma posição inicial, mantendo

uma posição relativa consistente. Esses métodos permitem fornecer, em condições ideais, a posição com precisão de 1 *m* CEP (*Circular Error Probable*), tornando-os adequados somente nas operações pré-plantio (BENSON *et al.* , 2003b; HANA *et al.* , 2004; HAGUE *et al.* , 2000; MONICO, 2000).

O direcionamento automático de máquinas agrícolas nas operações pós-plantio exige dados de posição extremamente precisos. Essa precisão, normalmente inferior a 10 *cm* CEP, pode ser obtida com a utilização de receptores de dupla frequência e de estações de correção RTK (*Real-Time Kinematic*). No entanto, esses receptores possuem uma grande desvantagem: o custo elevado (F.REID *et al.* , 2000; MONICO, 2000).

Grande parte das áreas agrícolas é cultivada em linhas, o que permite desenvolver métodos de reconhecimento e de extração da posição de máquinas agrícolas, relativa às linhas de cultivo, de imagens da cultura. Sistemas de direcionamento automático de custos menores podem, portanto, ser desenvolvidos para guiar as máquinas agrícolas em áreas onde as linhas de cultivo são perceptíveis (HANA *et al.* , 2004; JAHNS, 1997; PERNOMIAN, 2002).

1.2 Objetivos

Existem diversos problemas que dificultam a identificação das linhas de cultivo em uma imagem da área cultivada. A presença de sombras, ervas daninhas e falhas de plantio, por exemplo, exigem que o método de extração seja capaz de atuar de forma satisfatória em condições variadas.

Um dos requisitos para o desenvolvimento de métodos de extração de informações para sistemas de controle automático é o reduzido tempo de processamento. Algoritmos velozes fornecem altas taxas de dados e podem ser utilizados em veículos que operam em velocidades elevadas.

O objetivo principal deste trabalho é o desenvolvimento de um método de extração eficiente e robusto capaz de obter a posição de uma máquina agrícola, em relação às linhas de cultivo, de imagens fornecidas por uma câmera de vídeo colorida. A posição extraída é a informação básica utilizada em um sistema de direcionamento automático, permitindo determinar quais ações devem ser tomadas para manter a máquina em sua trajetória sem danificar as plantas da cultura.

1.3 Organização do Texto

Este trabalho encontra-se dividido em sete capítulos. No segundo capítulo são apresentados os conceitos fundamentais que formam a base do método proposto para a extração da posição de máquinas agrícolas das imagens da cultura.

No terceiro capítulo são apresentados os trabalhos relacionados ao trabalho proposto. São apresentados os principais componentes dos sistemas, os métodos de extração das informações e o posicionamento das câmeras.

No quarto capítulo é descrito o método proposto e apresentadas as suas vantagens e desvantagens. Testes preliminares de validação também estão contidos nesse capítulo.

No quinto capítulo é apresentada a plataforma de testes e de validação do método proposto. São descritos os módulos de hardware e software e os seus respectivos componentes. Também é apresentado nesse capítulo o sistema AGpilot®.

A descrição dos testes de validação do sistema está contida no sexto capítulo. Nesse capítulo estão descritos os materiais e métodos utilizados bem como os resultados obtidos.

As conclusões, as contribuições e os trabalhos futuros estão contidos no sétimo capítulo.

Capítulo 2

Conceitos Fundamentais

A habilidade humana de reconhecer, classificar e de discriminar objetos com extrema facilidade é ainda pouco desvendada e objeto de pesquisa em diversas áreas científicas. “Ensinar máquinas a enxergar” sempre se mostrou como uma tarefa desafiadora.

O desejo de reproduzir o sistema visual humano se estende desde a construção de dispositivos bioeletrônicos para deficientes visuais até a movimentação automática de veículos guiados por câmeras de vídeo. Os sistemas de visão computacional também têm atraído o interesse de diversos segmentos da indústria por permitir o aumento da eficiência do processo produtivo e a melhoria da qualidade dos produtos. Muitas das tarefas visuais realizadas pelo homem podem ser executadas por máquinas equipadas com um sistema de visão computacional. Robôs são empregados em tarefas como a de controle de qualidade em linhas de produção, de identificação e classificação de produtos ou ainda na movimentação em ambientes (TRUYENQUE, 2005)

Desenvolver sistemas dotados de uma capacidade visual semelhante à dos seres humanos é uma tarefa extremamente complexa. O sistema de visão humano possui uma base de dados muito rica e uma enorme capacidade de trabalhar sob condições muito variadas (GONZALES e WOODS, 2000; MARQUES FILHO e VIEIRA NETO, 1999).

Um dos problemas existentes no desenvolvimento de sistemas baseados em visão é a quantidade de informação de entrada disponível. Para reduzir o tempo de processamento, somente alguns processos básicos são normalmente implementados. Outro problema dos sistemas baseados em visão é a baixa confiabilidade e a instabilidade, ocasionadas principalmente por mudanças de iluminação, oclusão, movimento e ruído nos equipamentos de captura. O sistema de visão humano integra várias características que são observadas em paralelo, como cor, movimento e contornos, junto ao seu conhecimento do mundo para lidar com esses problemas (TRUYENQUE, 2005; GRASSI JÚNIOR, 2002; GONZALES e WOODS, 2000; MARQUES FILHO e VIEIRA NETO, 1999; GOMES e VELHO, 1994; JAIN *et al.*, 1995).

Devido aos problemas citados anteriormente, é muito complexo construir um sistema de visão computacional de propósito geral que seja capaz de trabalhar com toda classe de objetos e em todos os ambientes. É preciso restringir o campo de ação, mesclar tecnologias e construir sistemas mais específicos (TRUYENQUE, 2005; BERGQUIST, 1999).

As técnicas descritas a seguir são utilizadas direta ou indiretamente neste trabalho com o intuito de tornar mais robusto e eficaz o método proposto de extração da posição. O ponto de fuga e a busca antecipada são utilizados para simplificar a extração da posição e para aumentar a estabilidade dos controladores realimentados. As redes neurais pulsadas foram estudadas e a sua utilização foi direcionada ao pré-processamento das imagens de entrada.

2.1 Redes Neurais

Sistemas conexionistas são modelos computacionais criados com base na estrutura de funcionamento do cérebro humano e caracterizam-se pela reunião de grande quantidade

de elementos de processamento (células) interligados por diversas conexões que processam a informação de forma paralela. Esses sistemas são conhecidos como redes neurais artificiais (RNAs) devido à tentativa de imitação das células cerebrais, de suas características básicas e organização (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

Dentre as diversas partes que compõem o cérebro, os hemisférios cerebrais são as que ocupam a maior parte do crânio. A camada mais externa dessas regiões é conhecida como córtex cerebral e, apesar de pouco conhecido, há indícios de que essa região concentre as principais funções do cérebro. No córtex cerebral as células nervosas estão extremamente adensadas permitindo uma grande interconectividade entre elas (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

A unidade básica do cérebro, o neurônio, isoladamente assemelha-se a uma unidade de processamento. O neurônio, ilustrado na Figura 2.1, aceita e combina estímulos de vários outros neurônios (muitas entradas), porém possui somente uma saída. A intensidade dos estímulos recebidos em suas entradas (dendritos) depende da proximidade física dessas entradas com a saída (axônio) de cada um dos neurônios emissores dos estímulos. O axônio localiza-se na saída do corpo do neurônio (soma) e pode ser considerado como um dispositivo não-linear de disparo, que produz um pulso elétrico sempre que o somatório dos sinais dentro do soma atinge um certo limiar crítico. O soma é capaz de realizar funções mais complexas do que uma simples adição dos estímulos recebidos, porém o somatório simples é uma aproximação razoável (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

A ligação entre um neurônio e outro é realizada através de uma junção química temporária (sinapse) entre um axônio e um dendrito. A sinapse libera substâncias químicas (neurotransmissores) em função do pulso elétrico disparado pelo axônio. Os neurotransmissores fluem através de entradas quimicamente ativas dos dendritos (neuroreceptores) permitindo o fluxo de íons que altera a carga elétrica dos dendritos. Essa alteração cria

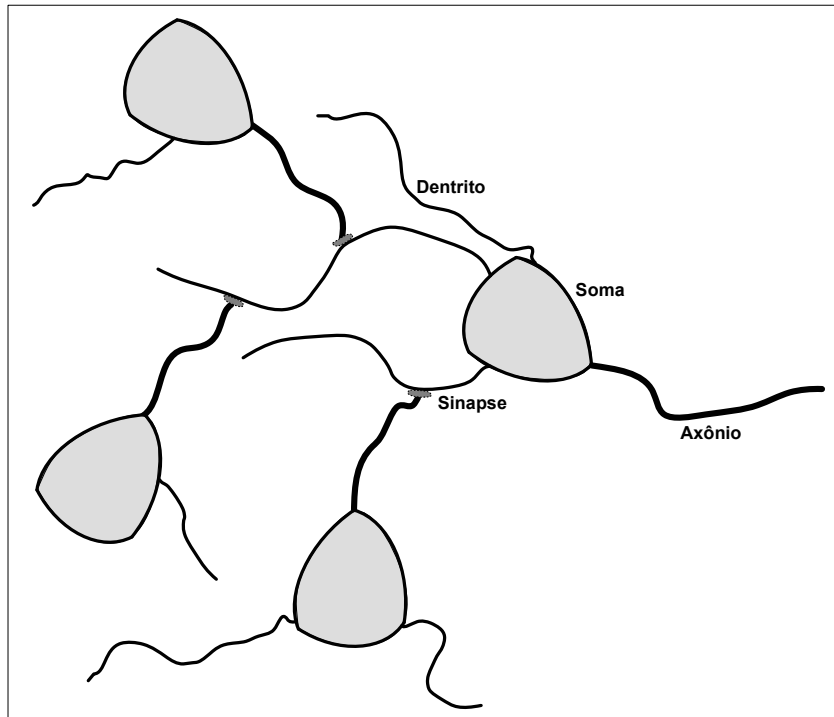


Figura 2.1: Representação simplificada dos neurônios.

uma diferença de potencial conduzida até o próximo soma (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

As sucessivas e efetivas modificações nas sinapses que interconectam os neurônios causadas em função da maior ou menor liberação de neurotransmissores, fazem com que os neurônios “aprendam”. À medida que novos eventos são apresentados, determinadas ligações são reforçadas enquanto outras enfraquecidas. Essa adaptação, ocasionada durante o processo de aprendizado, é uma das mais importantes características das redes neurais artificiais (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

O neurônio artificial pode ser representado por um simples dispositivo capaz de calcular o somatório de N entradas ponderadas, cujo resultado passa através de uma função não-linear. Esse dispositivo é caracterizado basicamente por um limiar interno e por algum tipo de função não-linear como mostra a Figura 2.2. Uma RNA é composta por um arranjo de elementos de processamento e pode ser categorizada por sua topologia, pelas características de seus elementos de processamento e pelas leis de aprendizagem a

que são submetidas (KRÖSE e SMAGT, 1996; HAYKIN, 1998; LIPPMANN, 1988).

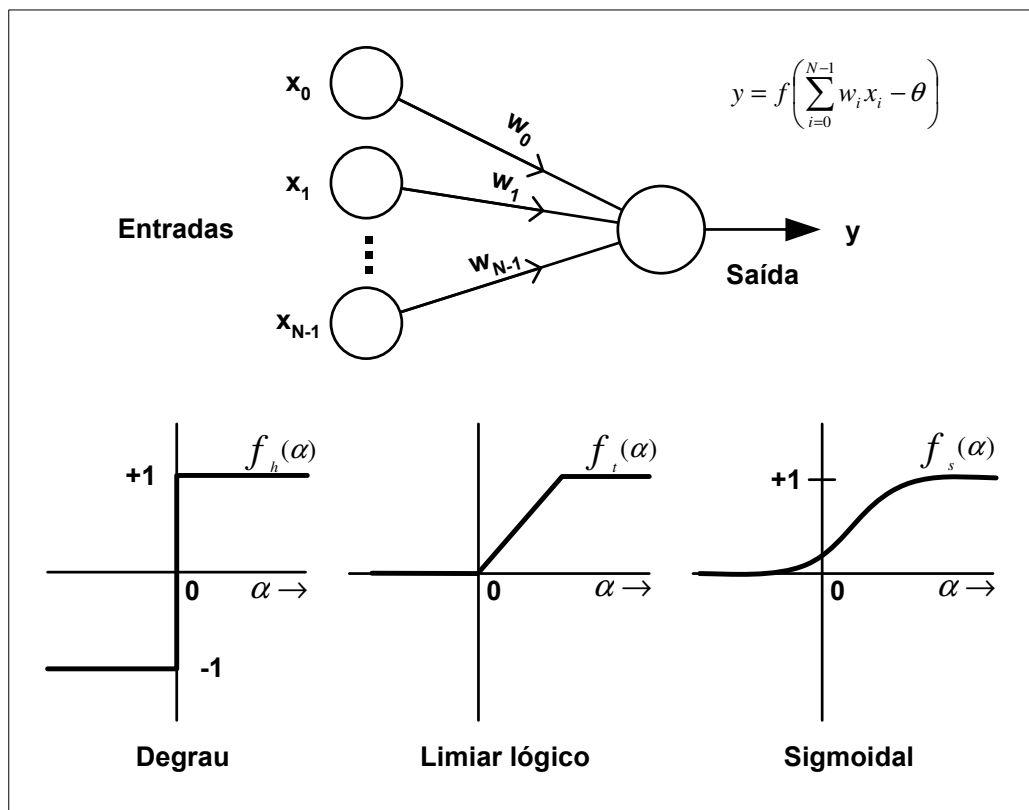


Figura 2.2: Elementos ou nós de processamento e tipos de funções mais comuns (Adaptado de LIPPMANN (1988)).

Existem diferentes maneiras de classificar as RNAs. A classificação mais importante é quanto à forma de aprendizado que pode ser supervisionada ou não-supervisionada. As redes também podem ser classificadas quanto às suas características como contínua, discreta, determinística ou estocástica; ou quanto à sua estrutura como redes de múltiplas camadas (*multilayer feedforward network*) ou redes recursivas (*recurrent network*) (KRÖSE e SMAGT, 1996; HAYKIN, 1998).

2.2 Rede Neural de Pulso Acoplado

Os neurônios de uma rede neural artificial geralmente são simples elementos somatórios e limitadores de sinal. Os neurônios biológicos, no entanto, são muito mais

complexos e realizam sofisticadas operações que não são contempladas pelos modelos computacionais. Esses neurônios são altamente especializados, as informações trafegam de um neurônio a outro através de pulsos e existem centenas de tipos distintos (LINDBLAD e KINSER, 2005).

Pesquisas recentes sobre o córtex visual de mamíferos forneceram importantes informações sobre os neurônios biológicos, permitindo o desenvolvimento de algoritmos mais sofisticados e direcionados ao processamento de imagens digitais. Esses algoritmos são conhecidos como PCNN (*Pulse-Coupled Neural Network*) ou Rede Neural de Pulso Acoplado, sendo o modelo baseado em neurônios pulsantes (*spiking neuron*) mais conhecido atualmente (LINDBLAD e KINSER, 2005; BECANOVIC, 2000; NAZMY, 2004).

A PCNN é uma rede neural que produz uma série temporal de pulsos binários e são classificadas por alguns autores como tipos elementares de osciladores (CAMPBELL, 1998). Segundo Yu e Zhang (2004), as PCNNs representam informação via oscilação de fase ou tempo de pulsos e os neurônios, em diferentes localizações, são relacionados pela sincronização de suas atividades. Em geral, também não são necessários algoritmos explícitos de treinamento: a similaridade surge do comportamento de seus neurônios.

As PCNNs podem ser facilmente implementadas em hardware por não serem compostas por diversas camadas como ocorre com as tradicionais redes neurais. São muito versáteis e poderosas podendo ser utilizadas na realização de diversas tarefas relacionadas ao processamento de imagens digitais como a segmentação de imagens, a extração de bordas, a identificação de objetos, a separação de objetos, a redução de ruídos, a fusão de imagens, entre outras (LINDBLAD e KINSER, 2005).

2.2.1 O Córtex Visual

Os seres humanos utilizam vários processos estruturados para interpretar as imagens obtidas do ambiente ao seu redor. Um dos processos ocorre no córtex visual que é responsável por receber e processar as imagens captadas pelos olhos. Essas imagens são significativamente modificadas pelos olhos antes de serem transmitidas ao córtex visual, que transforma o resultado dessa modificação em um fluxo (*stream*) de pulsos (LINDBLAD e KINSER, 2005).

Estudos realizados em primatas permitiram criar um modelo do córtex visual (Figura 2.3) que pode ser descrito basicamente por dois canais hierárquicos: o canal parvocelular, responsável principalmente por processar a informação de cor; e o canal magnocelular, responsável por processar a informação de forma e movimento. A retina possui detectores de cor e luminância que interpretam as imagens e as pré-processam antes de transmitir a informação. O NGL (Núcleo Geniculado Lateral) separa a imagem em componentes como luminância, contraste, frequência, etc., antes de enviá-la ao córtex visual (YOUNG, 1994; LINDBLAD e KINSER, 2005).

A área CV1 do córtex visual, ilustrada na Figura 2.3, representa o córtex visual primário onde permanece armazenada a imagem mais detalhada e menos processada. A área CV2 contém um mapa visual menos detalhado e pré-processado do que a área CV1. As áreas CV3, CV4 e CV5 podem ser interpretadas como áreas especializadas que processam somente informações específicas como cor e forma, forma estática e movimento, respectivamente (LINDBLAD e KINSER, 2005).

A Figura 2.3 ilustra apenas o sentido do fluxo de informações dentro do córtex visual utilizado pelos modelos matemáticos existentes. No entanto, sabe-se que as informações fluem em diversas direções o que torna as áreas CV1 a CV5 semelhantes a sistemas realimentados (LINDBLAD e KINSER, 2005).

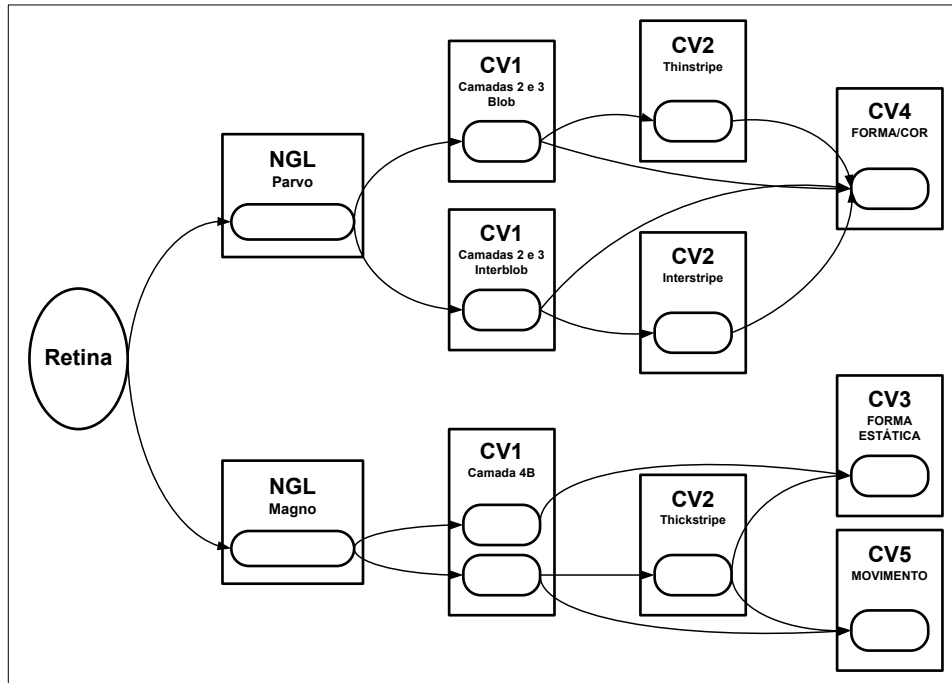


Figura 2.3: Modelo do sistema visual (Adaptado de LINDBLAD e KINSER (2005)).

2.2.2 Modelos Matemáticos do Córtex Visual

Existem vários modelos matemáticos do córtex visual. Esses modelos são importantes por descreverem os neurônios como uma equação diferencial e a sua dinâmica como um processo oscilatório (LINDBLAD e KINSER, 2005).

O primeiro modelo matemático do córtex visual foi introduzido por Eckhorn (1990) que modelou o córtex visual de gatos, tornando-se a base para as implementações direcionadas ao processamento digital de imagens. Nesse modelo o neurônio possui duas entradas: *feeding* (F_k) e *linking* (L_K). O *feeding* recebe estímulos externos e locais enquanto que o *linking* recebe apenas estímulos locais. Essas entradas são combinadas para criar uma diferença de potencial (U_m) que é comparada a um limiar local (θ). Esse modelo é descrito pelas seguintes equações:

$$F_k(t) = \sum_{i=1}^N \left[w_{ki}^f Y_i(t) + S_k(t) + N_k(t) \right] \otimes I(V^a, \tau^a, t) \quad (2.1)$$

$$L_k(t) = \sum_{i=1}^N \left[w_{ki}^l Y_i(t) + N_k(t) \right] \otimes I(V^l, \tau^l, t) \quad (2.2)$$

$$U_{m,k}(t) = F_k(t) [1 + L_k(t)] \quad (2.3)$$

$$Y_k(t) = \begin{cases} 1, & U_{m,k}(t) \geq \Theta_k(t) \\ 0, & U_{m,k}(t) < \Theta_k(t) \end{cases} \quad (2.4)$$

onde N é o número de neurônios, w é o peso sináptico, Y é a saída binária, S é o estímulo externo e os valores típicos são $\tau^a = [10; 15]$, $\tau^l = [0, 1; 1, 0]$, $\tau^s = [5; 7]$, $V^a = 0, 5$, $V^l = [5; 30]$, $V^s = [50; 70]$, $\Theta_0 = [0, 5; 1, 8]$ (LINDBLAD *et al.*, 1997).

As implementações computacionais dos modelos matemáticos do córtex visual foram iniciadas por Johnson e Ritter (1993) *apud* Nazmy (2004), que vislumbrou a utilização das PCNNs para o processamento de imagens digitais. A arquitetura das implementações direcionadas a essa aplicação é muito simples, pois se utiliza apenas uma camada bidimensional de neurônios conectados lateralmente. Como mostra a Figura 2.4, cada neurônio recebe dados de um *pixel* da imagem original e gera um *pixel* monocromático (pulso) para compor a imagem de saída (NAZMY, 2004).

Com base nos trabalhos de Eckhorn (1990) e Johnson e Ritter (1993), Lindblad e Kinser (2005) propuseram uma implementação de PCNN direcionada ao processamento de imagens digitais onde o neurônio é um tipo particular de integrador com fuga contendo os compartimentos *feeding* e *linking*. Esses compartimentos, ilustrados na Figura 2.5, recebem dados de neurônios vizinhos definidos a partir de uma distância r que representa o raio de um círculo ou a largura de um quadrado centrado no *pixel* atual.

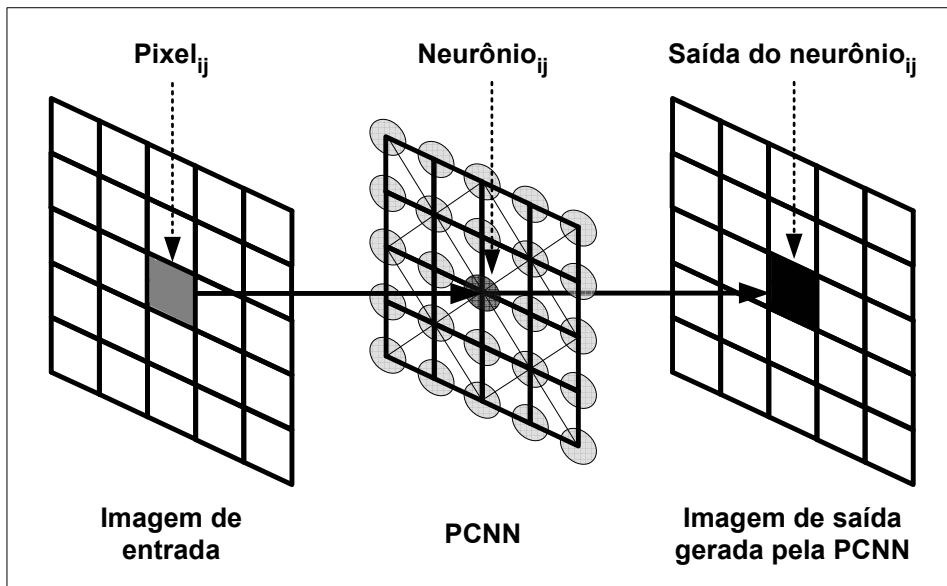


Figura 2.4: PCNN aplicada ao processamento de imagens digitais (Adaptado de NAZMY (2004)).

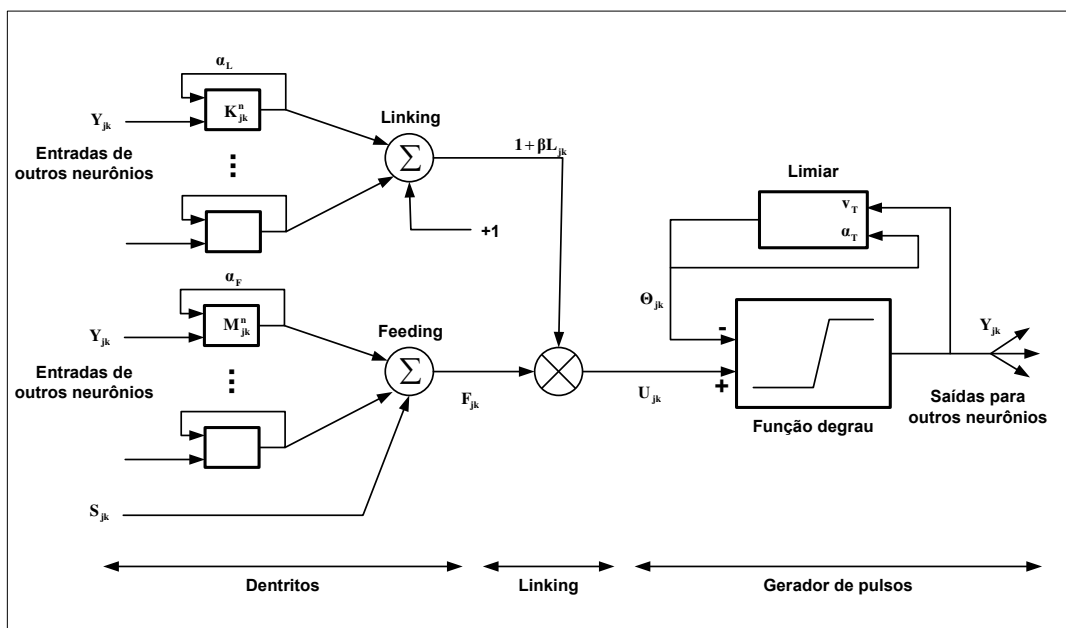


Figura 2.5: O neurônio PCNN (Adaptado de LINDBLAD e KINSER (2005)).

Na implementação proposta por Lindbland e Kinser (2005), a ativação de um neurônio ocorre quando os sinais acumulados internamente ultrapassam um limiar dinâmico ocasionando um pulso de saída, alterando o limiar e os compartimentos *linking* e *feeding* do neurônio e de seus vizinhos de acordo com as seguintes equações:

$$F_{ij}[n] = e^{\alpha_F} F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \quad (2.5)$$

$$L_{ij}[n] = e^{\alpha_L} L_{ij}[n-1] + V_L + \sum_{kl} W_{ijkl} Y_{kl}[n-1] \quad (2.6)$$

onde i, j representa o i -ésimo e j -ésimo neurônio ($1 \leq i \leq a, 1 \leq j \leq b$), α_F e α_L as constantes de decremento, S_{ij} a intensidade do *pixel* x, y (estímulo de entrada) e V_F e V_L os potenciais. M e W são as matrizes de pesos sinápticos (*kernels*) de dimensões $(2r+1) \times (2r+1)$ com valores típicos definidos por uma distribuição normal. As atividades de $F_{i,j}$ e $L_{i,j}$ são reduzidas exponencialmente a uma taxa determinada pelas constantes de tempo α_F e α_L respectivamente. O termo Y_{kl} se refere à saída dos neurônios a partir de $n-1$ iterações prévias. Os dois componentes F e L são combinados para formar o estado interno U como:

$$U_{ij}[n] = F_{ij}[n](1 + \beta L_{ij}[n]) \quad (2.7)$$

onde β é a força de ligação entre os dois componentes. O estado U é comparado com o limiar dinâmico Θ para compor a saída Y do *pixel* i, j como:

$$\Theta_{ij}[n] = e^{\alpha\Theta}[n-1] + V_{\Theta}Y_{ij}[n] \quad (2.8)$$

$$Y_{ij}(n) = \begin{cases} 1, & U_{i,j}[n] > \Theta_{i,j}[n-1] \\ 0, & U_{i,j}[n] \leq \Theta_{i,j}[n-1] \end{cases} \quad (2.9)$$

O limiar dinâmico Θ é acrescido de V_{Θ} sempre que o neurônio dispara ($Y > \theta$), inibindo subseqüentes disparos até que Θ seja reduzido a um valor menor do que Y . Os estados prévios de Θ , F e L são armazenados em memória para o cálculo de seus respectivos valores atuais. F é muito afetado pela intensidade do *pixel* (S) enquanto que L é afetado pelos neurônios vizinhos.

As conexões neurais existentes em uma rede PCNN permitem que neurônios vizinhos com estímulos semelhantes pulsem conjuntamente. Quando os neurônios possuem níveis similares de atividades em relação aos respectivos limiares, ou seja, quando existem similaridades nos *pixels* de entrada ou texturas presentes, a sincronicidade ocorre através da captura de pulso. Esse fenômeno é conhecido como *autowave* e se assemelha a propagação de ondas na superfície de um lago produzidas pelo impacto de um objeto, exceto pelo fato de que em uma PCNN não existe reflexão ou refração de ondas (JUNYING *et al.*, 2004).

Lindblad e Kinser (2005) também propuseram uma implementação simplificada do modelo matemático do córtex visual proposto por Eckhorn (1990). Denominado de ICM (*Intersecting Cortical Model*), essa implementação foi desenvolvida com o intuito de minimizar o custo computacional e é descrita como:

$$F_{ij}[n+1] = fF_{ij}[n] + S_{ij} + W\{Y\}_{ij} \quad (2.10)$$

$$Y_{ij}[n+1] = \begin{cases} 1, & \text{se } F_{ij}[n+1] > \Theta_{i,j}[n] \\ 0, & \text{se } F_{ij}[n+1] \leq \Theta_{i,j}[n] \end{cases} \quad (2.11)$$

e

$$\Theta_{ij}[n+1] = g\Theta_{i,j}[n] + hY_{i,j}[n+1] \quad (2.12)$$

onde S representa o vetor de entrada, F o vetor de estado dos neurônios, Y o vetor de saída e Θ os limiares dinâmicos. Os termos f e g são valores escalares menores do que 1 e $g < f$ para impedir que o neurônio pulse caso o limiar fique abaixo do estado atual. O termo h é utilizado para incrementar drasticamente o limiar sempre que um neurônio pulse, impedindo que o mesmo pulse novamente nas iterações subseqüentes. A função $W\{\}$, normalmente $1/r$, descreve as conexões entre os neurônios.

2.2.3 Aplicações

Experimentos já realizados demonstraram que as PCNNs podem ser utilizadas em várias aplicações relacionadas ao processamento de imagens digitais como o reconhecimento de objetos, a fusão de imagens e a análise de textura. Uma característica importante também demonstrada é a invariância à rotação, distorção, escala e deslocamento. Bordas e segmentos podem ser facilmente obtidos a um custo computacional pequeno. As PCNNs também não necessitam de treinamento prévio ou ajustes realizados por meio de um conjunto de imagens (LINDBLAD e KINSER, 2005; BECANOVIC, 2000; NAZMY,

2004; YU e ZHANG, 2004; RUGHOOPUTH *et al.* , 2003; HASSANIEN e ALI, 2004; GU *et al.* , 2005).

A Figura 2.6 apresenta algumas imagens processadas com uma implementação do modelo simplificado ICM proposto por Lindblad e Kinser (2005). A Tabela 2.1 contém os parâmetros utilizados em cada operação. Detalhes de implementação do modelo ICM podem ser obtidos no Apêndice A.

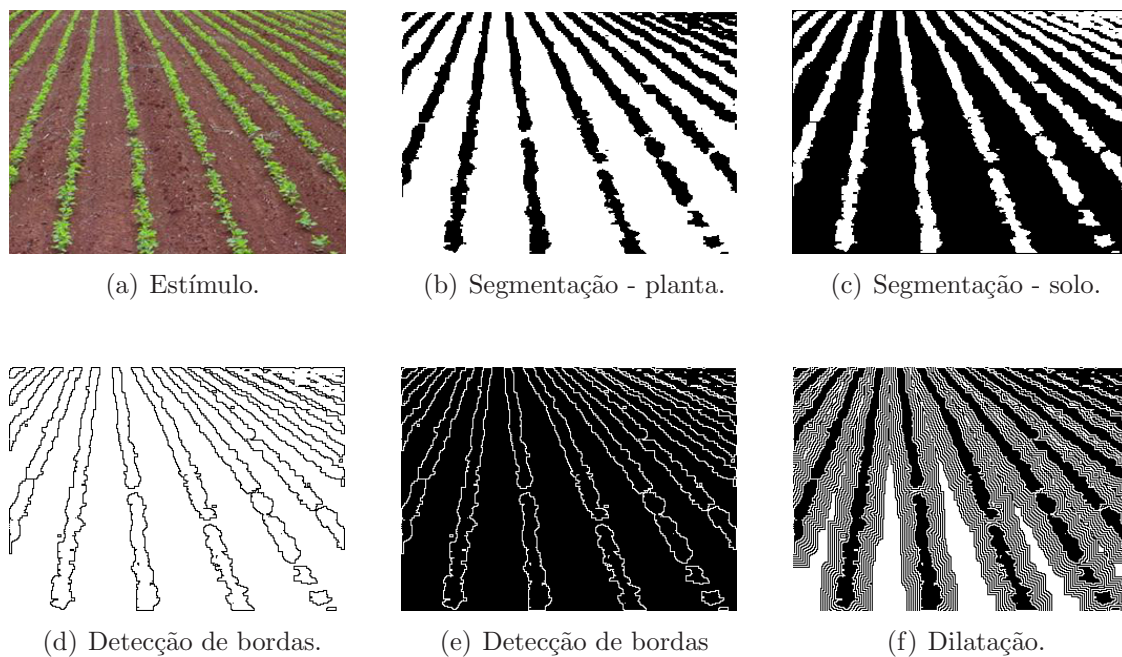


Figura 2.6: Imagem original e os resultados obtidos.

Tabela 2.1: Parâmetros da rede ICM.

Figura	Iterações	f	g	h
(a)	-	-	-	-
(b)	3	0,1	0,09	16
(c)	4	0,1	0,09	16
(d)	6	0,1	0,09	16
(e)	7	0,1	0,09	16
(f)	45	0,1	0,09	16

2.3 Controle Lateral

O direcionamento automático é a principal, em alguns casos a única, função de um sistema de controle automático de veículos. Esses sistemas são normalmente compostos por um controlador longitudinal responsável principalmente pelo controle de velocidade e por um controlador lateral que automaticamente guia o veículo mantendo a sua posição no centro da trajetória desejada (SOTELO, 2003; ZHANG e QIU, 2004; E. J. ROSSETTER, 2002; FU *et al.*, 2004; GULDNER e TAN, 1997).

As técnicas existentes para o controle automático de direção podem ser agrupadas em duas categorias principais: busca antecipada (*look-ahead*) e busca para trás (*look-down*), em função da localização do ponto de medição do deslocamento lateral em relação à rota seguida. Os sistemas de busca antecipada replicam o comportamento humano de dirigir pela medição do deslocamento lateral à frente do veículo. A distância “à frente” é normalmente incrementada com o aumento da velocidade, semelhante ao comportamento humano. No entanto, os sistemas de busca para trás medem o deslocamento lateral em uma posição muito próxima do veículo, comumente por meio de marcadores magnéticos fixados no solo. Esses sistemas apresentam algumas limitações no que se refere à velocidade máxima de operação, conforto e robustez (SOTELO, 2003; ZHANG e QIU, 2004; E. J. ROSSETTER, 2002; FU *et al.*, 2004; GULDNER e TAN, 1997).

O método de busca antecipada é freqüentemente utilizado no desenvolvimento de sistemas de direcionamento automático por simplificar o ajuste e por garantir a estabilidade dos controladores realimentados mesmo em altas velocidades. Guldner e Tan (1997) demonstraram que esse método provê compensação por avanço de fase em velocidades elevadas, permitindo que controladores realimentados do tipo proporcional sejam suficientes para o controle lateral do veículo.

O objetivo final de um controlador lateral baseado no método de busca antecipada

é o de zerar o erro de posicionamento calculado a partir da possível posição futura do veículo e da trajetória a ser seguida (BOUGLER *et al.* , 2003; PATWARDHAN *et al.* , 1997). Em um sistema de controle de direção automático baseado em dados providos por um receptor GPS, a distância futura de um veículo em relação a sua correta trajetória pode ser calculada por meio da seguinte função trigonométrica:

$$D_F = D_C + (\sin(C) * D_A) \quad (2.13)$$

onde D_F é a distância futura, D_C e C são, respectivamente, a distância atual e a direção do veículo calculadas a partir do trajeto a ser seguido e D_A é a distância “à frente” do veículo obtida a partir da velocidade.

A direção do veículo, como mostra a Figura 2.7, corresponde ao ângulo calculado a partir da direção fornecida pelo receptor GPS (*heading*) e da direção do segmento de reta formado pelos *waypoints* do trajeto desejado. Nessa configuração, C é negativa quando o veículo gira para a esquerda e positiva quando gira para a direita e D_C é negativa quando o veículo está à esquerda da trajetória desejada e positiva quando estiver a sua direita.

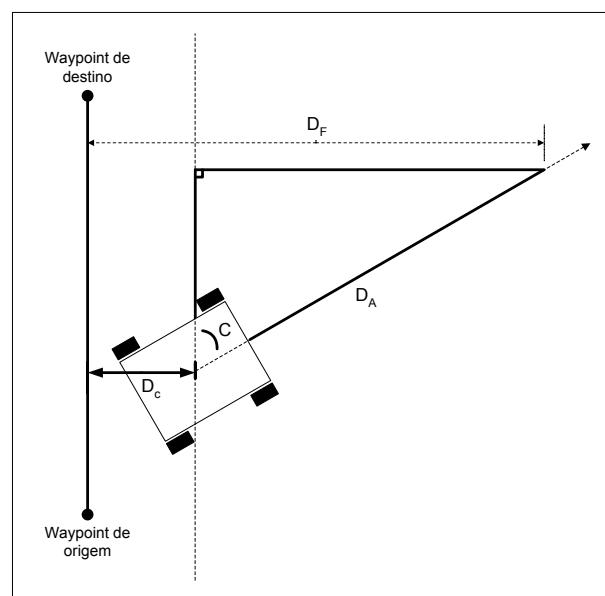


Figura 2.7: Trajetória do veículo.

2.4 Modelo de Câmera e Pontos de Fuga

O modelo de câmera comumente utilizado em visão computacional é o modelo pontual ou *pinhole model*. Esse modelo se baseia na passagem dos raios luminosos por um pequeno orifício na entrada da câmara e na sua projeção em uma superfície plana, como ilustrado na Figura 2.8. O orifício de passagem da luz é conhecido como ponto focal e a superfície plana de projeção é conhecido como plano de projeção. As câmeras baseadas nesse princípio são chamadas câmeras perspectivas, sendo uma maneira simples e funcional de modelar as câmeras digitais (FAUGERAS, 1993).

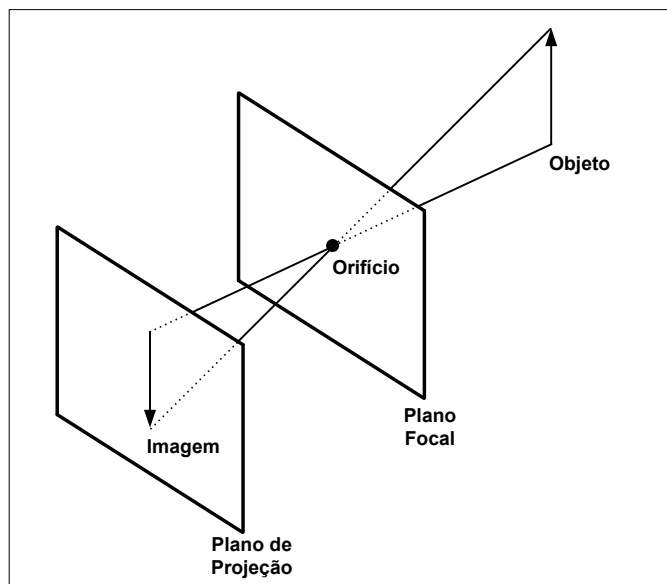


Figura 2.8: Modelo de câmera.

Para que as descrições de um objeto possam ser projetadas em um plano de projeção, e conseqüentemente analisadas, é necessário transferi-las para o sistema de coordenadas do plano de projeção. A conversão de coordenadas é realizada através de operações geométricas simples de translação e rotação. Inicialmente translada-se o ponto de referência do plano de projeção para a origem do sistema de coordenadas do mundo e aplicam-se rotações para alinhar respectivamente os eixos x_p , y_p e z_p do plano de projeção com os

eixos x_m , y_m e z_m do mundo (HEARN e BAKER, 1997; VINCE, 2006).

A conversão dos sistemas de coordenadas permite projetar um objeto tridimensional em um plano de projeção bidimensional, eliminando a informação de profundidade existente no espaço tridimensional. Existem dois métodos básicos de projeção: a paralela e a perspectiva, diferenciados pela distância do plano de projeção ao centro de projeção. Na projeção paralela, ilustrada na Figura 2.9(a), todos os raios de projeção são paralelos devido ao centro de projeção estar localizado a uma distância infinita em relação ao plano de projeção, preservando retas paralelas e ângulos apenas em planos paralelos ao plano de projeção. No entanto, na projeção perspectiva (Figura 2.9(b)) o centro de projeção está localizado a uma distância finita do plano de projeção, fazendo com que os raios de projeção concentre-se em um ponto chamado de centro de projeção, causando deformações no objeto projetado (HEARN e BAKER, 1997; VINCE, 2006).

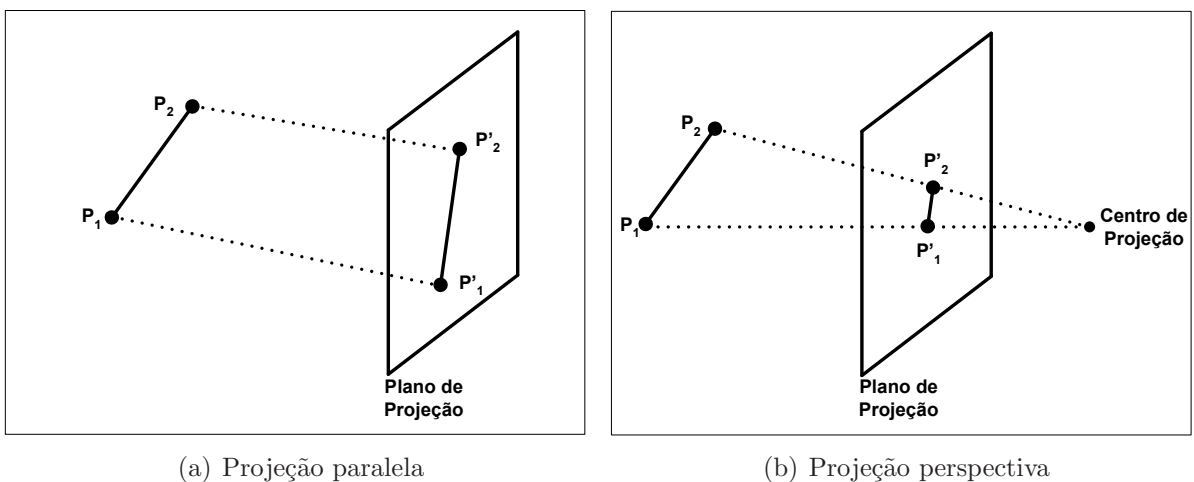


Figura 2.9: Projeções de um objeto no plano de projeção.

As deformações da projeção perspectiva criam uma representação bem realista do objeto projetado, por simularem a maneira como o olho percebe os objetos no espaço. Nela, encontram-se duas características básicas: o encurtamento perspectivo, ou seja, a ilusão de que objetos são cada vez menores à medida que sua distância do plano de projeção aumenta; e os pontos de fuga (*vanishing points*), decorrentes da ilusão de que

linhas paralelas (desde que não sejam paralelas ao plano de projeção) convergem a um determinado ponto (Figura 2.10). Cada conjunto de linhas paralelas projetadas pode ter diferentes pontos de fuga e, geralmente, uma cena poder ter vários pontos de fuga dependendo da quantidade de conjuntos de linhas paralelas existentes (HEARN e BAKER, 1997; VINCE, 2006).



Figura 2.10: Pontos de fuga (cultura de milho).

Grande parte das imagens reais apresenta algum tipo de distorção em sua projeção. Essas irregularidades podem ser utilizadas para analisar cenas tridimensionais e extrair informações úteis das imagens projetadas. Os pontos de fuga, por exemplo, têm sido utilizados em diversas aplicações como a calibração dos parâmetros de câmeras, orientação de objetos e para navegação *indoor* e *outdoor* (CANTONI *et al.* , 2001; KOSMOPOULOS e CHANDRINOS, 2002; SIMOND e RIVES, 2003; STENTIFORD, 2006; HAVASI e SZIRANYI, 2006).

Para se obter a projeção perspectiva de um objeto 3D, transformam-se os pontos ao longo dos projetores que se encontram no centro de projeção. Supondo estar o centro de projeção posicionado em z_{prp} ao longo do eixo z_v , e o plano de projeção posicionado

em z_{vp} , como mostra a Figura 2.11, podem-se formular as equações que descrevem as coordenadas x_0, y_0, z_0 de qualquer ponto ao longo da linha de projeção perspectiva como:

$$\begin{aligned}x' &= x - xu \\y' &= y - yu \\z' &= z - (z - z_{prp})u\end{aligned}\tag{2.14}$$

onde o parâmetro u assume valores entre 0 e 1. Quando $u = 0$ a posição é $P = (x, y, z)$. O centro de projeção $(0, 0, z_{prp})$ está localizado na outra extremidade e é obtido quando $u = 1$. Como $z' = z_{vp}$ no plano de observação, pode-se resolver a equação z' para obter o valor do parâmetro u ao longo da linha de projeção pela seguinte equação:

$$u = \frac{z_{vp} - z}{z_{prp} - z}\tag{2.15}$$

Substituindo esse valor de u nas equações de x' e y' obtém-se as equações de transformação perspectiva:

$$\begin{aligned}x_p &= x \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z} \right) = x \left(\frac{d_p}{z_{prp} - z} \right) \\y_p &= y \left(\frac{z_{prp} - z_{vp}}{z_{prp} - z} \right) = y \left(\frac{d_p}{z_{prp} - z} \right)\end{aligned}\tag{2.16}$$

onde $d_p = z_{prp} - Z_{vp}$ é a distância do plano de projeção em relação ao centro de projeção (HEARN e BAKER, 1997; VINCE, 2006).

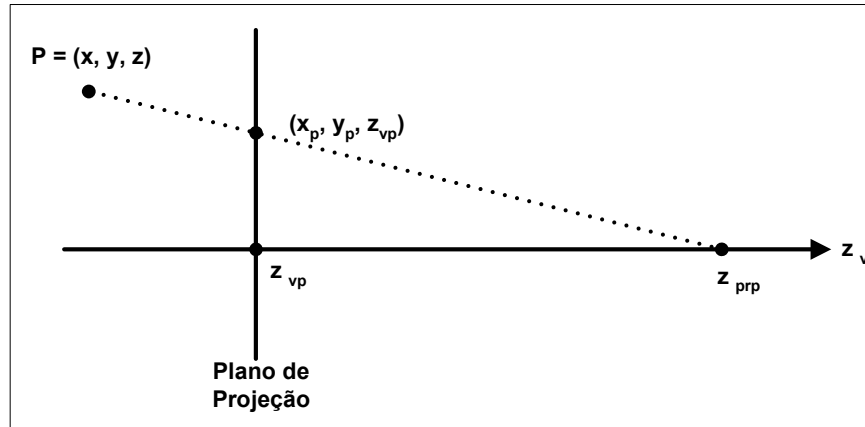


Figura 2.11: Projeção perspectiva de um ponto $P(x, y, z)$ em um plano de projeção.

Utilizando coordenadas homogêneas pode-se escrever a transformação da projeção perspectiva na forma matricial como:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{prp}/d_p & z_{prp}(z_{prp}/d_p) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.17)$$

Nessa representação o fator homogêneo é:

$$h = \frac{z_{prp} - z}{d_p} \quad (2.18)$$

e as coordenadas do ponto projetado no plano de observação são obtidas a partir das coordenadas homogêneas pela seguinte equação:

$$\begin{aligned} x_p &= \frac{x_h}{h} \\ y_p &= \frac{y_h}{h} \end{aligned} \quad (2.19)$$

Em geral, o centro da projeção não necessita ser posicionado ao longo do eixo

z_v . As equações acima podem ser generalizadas para considerar o centro em um ponto qualquer $(x_{prp}, y_{prp}, z_{prp})$ (HEARN e BAKER, 1997; VINCE, 2006).

Capítulo 3

Métodos de Extração da Posição e de Orientação de Máquinas Agrícolas

O direcionamento automático de veículos não é um item novo na mecanização agrícola. Seguidores mecânicos de sulco e sistemas de direcionamento por cabos magnéticos foram desenvolvidos para guiar máquinas agrícolas através de uma rota previamente definida (RICHEY, 1959; RUSHING, 1971) *apud* (BENSON *et al.* , 1998). Sistemas de posicionamento a laser, óticos e por rádio também foram desenvolvidos e aplicados na orientação de máquinas agrícolas com níveis de sucesso variado (NOGUCHI *et al.* , 1997; GORDON e HOLMES, 1988) *apud* (BENSON *et al.* , 1998). Esses sistemas necessitam de equipamentos permanentes ou temporários localizados na área de cultivo ou próximas a elas. Apesar de muito precisos esses sistemas possuem a desvantagem de serem limitados em relação ao alcance (BENSON *et al.* , 1998).

Os primeiros trabalhos relacionados à automação agrícola eram muito prejudicados pela limitação dos sensores utilizados, sendo em muitos casos o responsável pelo fracasso das pesquisas. O surgimento dos modernos receptores GPS e dos sensores de baixo custo, de tamanho reduzido e de maior precisão, possibilitou um grande avanço das pesquisas

em automação agrícola, principalmente as relacionadas ao direcionamento automático e assistido. (ELKAIM *et al.* , 1997).

Sistemas de posicionamento baseados em satélites podem fornecer a posição de um receptor localizado em qualquer parte do mundo. Apesar desses sistemas possuírem um alcance global, eles não podem fornecer informações sobre o ambiente local. Os sistemas de visão computacional, no entanto, podem fornecer informações locais, mas não globais (BENSON *et al.* , 1998).

Diversas pesquisas envolvendo a visão computacional estão sendo realizadas na área agrícola (REID e SEARCY, 1987; HANA *et al.* , 2004; PILARSKI *et al.* , 2002; FONTAINE e CROWE, 2006; ZHANG *et al.* , 2005). Os trabalhos vão desde a detecção em tempo real de ervas daninhas (PERNOMIAN, 2002) até o desenvolvimento de sistema de orientação automática para colhedoras de grãos (BENSON *et al.* , 2003b).

A seguir são descritos alguns trabalhos relevantes quando comparados ao trabalho proposto.

3.1 Sistema de Orientação Baseado na Detecção de Borda

Benson, Reid e Zhang (2003b) desenvolveram um sistema de orientação baseado em visão computacional capaz de guiar automaticamente uma colhedora de milho durante a colheita. O sistema calcula a posição da colhedora em relação à área a ser colhida a partir das imagens de câmeras posicionadas sobre as laterais da plataforma de colheita, como mostra a Figura 3.1. O sistema obtém os dados de orientação analisando a diferença entre a região colhida e a região não colhida da imagem.

A colhedora de grãos utilizada como plataforma de testes foi modificada para se

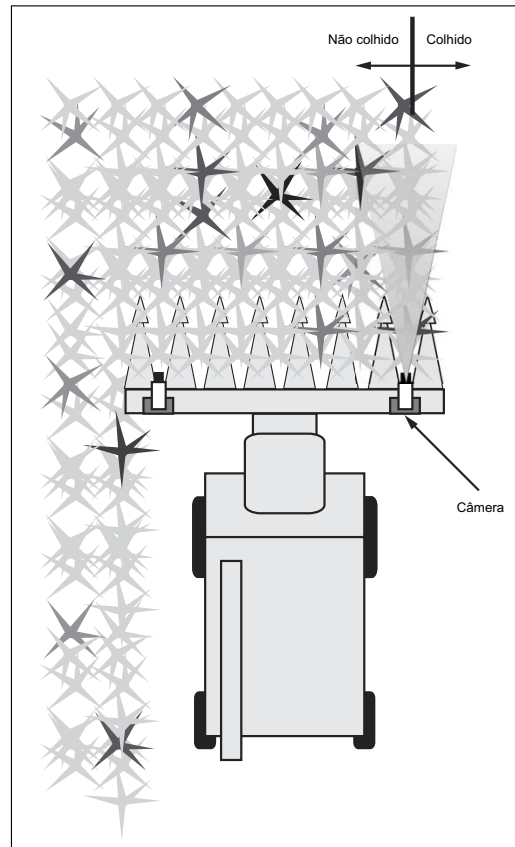


Figura 3.1: Posicionamento das câmeras (Adaptado de BENSON *et al.* (2003b)).

adequar ao sistema desenvolvido. O sistema de direção da colhedora foi alterado para permitir o controle eletrônico. Dois computadores, um para o processamento das imagens e navegação e o outro para o controle da direção, foram instalados na colhedora.

Como sensor de imageamento foi utilizada uma câmera analógica monocromática com resolução de 752×582 *pixels*. Para reduzir o tempo de processamento a resolução da câmera foi configurada para 320×243 *pixels*. A câmera utilizada também possui um controle de ganho automático capaz de reduzir os efeitos de mudanças não muito drásticas de iluminação. A Figura 3.2 ilustra uma imagem coletada pela câmera descrita anteriormente.

Após a aquisição da imagem, um módulo de detecção de sombras é acionado para determinar a sua presença. No passo seguinte, um módulo de segmentação adaptativo calcula o apropriado nível de segmentação a ser aplicado a partir do histograma da imagem



Figura 3.2: Exemplo de imagem coletada (BENSON *et al.* , 2003b).

e do estado obtido pelo módulo de detecção de sombras. A segmentação é realizada por um simples algoritmo classificador de *pixel* (colhido, não colhido) implementado a partir da seguinte equação:

$$p_c(i, j) = \begin{cases} 1, & p(i, j) > p_{gsl} \\ 0, & p(i, j) < p_{gsl} \end{cases} \quad (3.1)$$

onde $p_c(i, j)$ é a classe do *pixel*, $p(i, j)$ é o nível de cinza do *pixel* e p_{gsl} é o nível de segmentação adaptativo.

Após a segmentação da imagem e da detecção da borda, um módulo *fuzzy* é executado para determinar a qualidade do resultado obtido e conseqüentemente a sua aceitabilidade. Um controlador proporcional foi utilizado para manter a máquina em sua trajetória através da seguinte equação:

$$\delta_d = K_p(x_{goal} - x_{act}) \quad (3.2)$$

onde K_p é o ganho do controlador proporcional, x_{goal} é a localização horizontal ideal da linha de colheita, x_{act} é localização horizontal calculada da linha de colheita e δ_d é o ângulo, em graus, da roda necessário para manter ou corrigir a trajetória.

Os testes realizados em campo apresentaram resultados semelhantes ao de uma máquina guiada por um operador humano. No entanto, problemas relacionados à qualidade das imagens limitaram a utilização do sistema desenvolvido.

3.2 Detecção de Linhas Baseado em Visão Estéreo-cópica

Um método de detecção das linhas de cultivo baseado em visão estereoscópica foi proposto por Kise, Zhang e Más (2005). Para validar o método proposto foi desenvolvido um sistema de direcionamento automático para um trator John Deere 7700.

O trator utilizado como plataforma de testes foi modificado para permitir o controle eletrônico da direção através de um sistema eletro-hidráulico. O processamento das imagens estereoscópicas, a navegação e o cálculo do ângulo das rodas são realizados por um *notebook*. Um microcontrolador é responsável pelo controle das rodas do trator. Para a validação dos testes realizados em campo foi utilizado um receptor GPS RTK.

O principal componente do sistema é uma câmera estereoscópica ST-MD1 com lente de 12 *mm* fabricada pela empresa Videre Design. Ilustrada na Figura 3.3, essa câmera possui sensores CMOS (*Complementary Metal Oxide Semiconductor*) capazes de fornecer imagens com 1,3 *megapixel* de resolução a 7 (1280x1032 *pixels*), 25 (640x512 *pixels*) e 80 (320x256 *pixels*) quadros por segundo. As câmeras são controladas individualmente através de uma interface IEEE 1394 (*FireWire*). Um filtro de infravermelho próximo (880 *nm*) foi utilizado para realçar a vegetação presente no solo. A câmera foi

fixada a 1,7 m do solo a um ângulo de visão horizontal e vertical (HFOV e VFOV) de 50° e 38° respectivamente, permitindo captura de imagens do campo a distâncias aproximadas de 1 a 8 metros.



Figura 3.3: Câmera estereoscópica (KISE *et al.* , 2005).

O método desenvolvido constrói inicialmente um mapa tridimensional de elevação das linhas de cultivo a partir das imagens estereoscópicas e então localiza um ponto de navegação ótima a partir do mapa gerado. O mapa de elevação é construído a partir das posições 3D dos pontos da cena e utiliza um *array* bidimensional de intensidade para indicar a altura das plantas. O valor de correção da rota, a ser aplicado ao sistema de direção eletro-hidráulico, é determinado a partir da diferença entre o ponto central do mapa de elevação e do ponto mediano entre duas linhas de cultivo como ilustra a Figura 3.4.

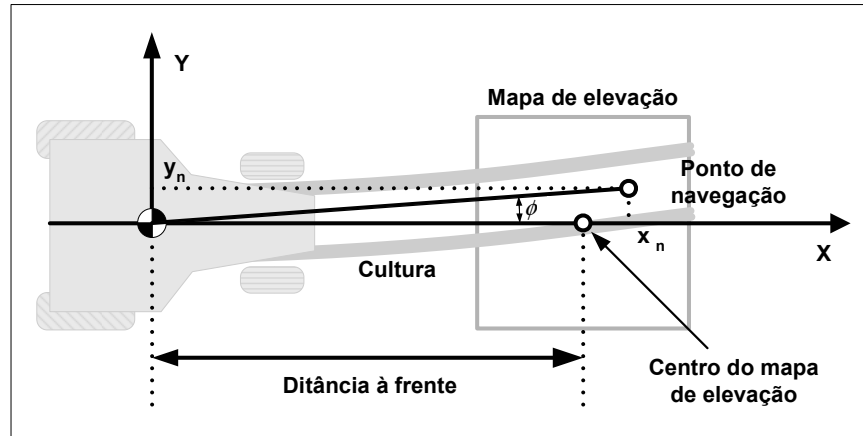


Figura 3.4: Cálculo do ponto de navegação (Adaptado de BENSON *et al.* (2003b)).

O ângulo da roda necessário para manter ou corrigir a trajetória é calculado através da seguinte equação:

$$\phi = \tan^{-1} \left(\frac{y_n}{x_n} \right) \quad (3.3)$$

$$\delta = a\phi \quad (3.4)$$

onde (x_n, y_n) é um ponto de navegação identificado em relação às coordenadas do veículo, ϕ é o *heading* do trator em relação às linhas de cultivo e a é o ganho do controlador proporcional. A distância “à frente” e o ganho do controlador foram obtidos empiricamente sendo de 4,5 m e 1,0 respectivamente.

A Figura 3.5 ilustra uma imagem de uma área cultivada com soja (a) e o respectivo mapa de elevação (b). A distância entre as linhas é de 75 cm e a média de altura das plantas é de 40 cm.

Segundo os autores, os testes realizados em campo apresentaram bons resultados. O sistema foi capaz de guiar automaticamente o trator através de uma área cultivada com soja. O desvio lateral não ultrapassou a 0,05 m (RMS). A velocidade máxima foi de 10,0 km/h.

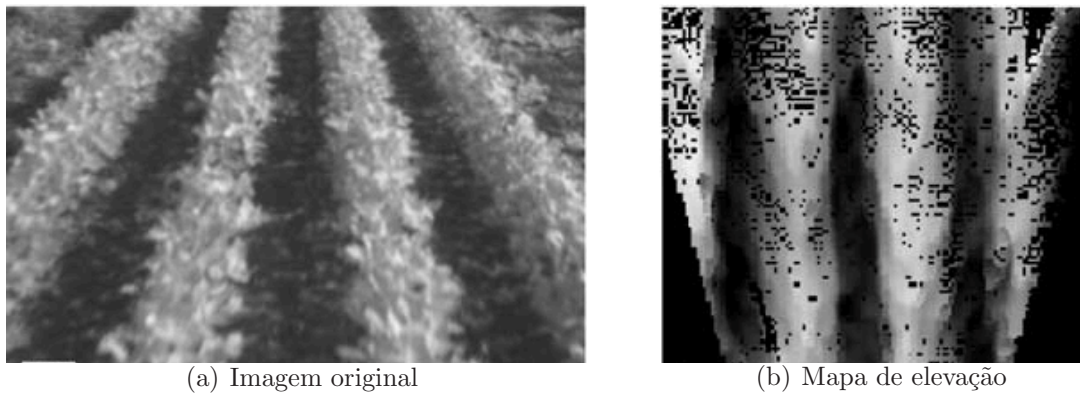


Figura 3.5: Imagens processadas pelo sistema.

3.3 Detecção de Linhas Baseado na Transformada de Hough

Astrand e Baerveldt (2005) desenvolveram um algoritmo de detecção de linhas de cultivo capaz de se adaptar a diferentes tamanhos de plantas. O algoritmo é robusto o suficiente para detectar as linhas de cultivo mesmo quando infestadas por ervas daninhas.

A detecção das linhas de cultivo a partir das imagens capturadas permite extrair as informações de *heading* e de deslocamento lateral da máquina. Essas informações são extraídas por meio de um modelo geométrico capaz de transformar o sistema de coordenadas do mundo no sistema de coordenadas da imagem através da transformação de projeção perspectiva. A Figura 3.6 mostra a geometria do sistema de coordenadas mundo-câmera utilizado.

Para valores pequenos de *heading* demonstrou-se, pela transformação perspectiva, que existe uma relação linear entre o *heading* e o deslocamento lateral em relação às linhas de cultivo dado pelas seguintes equações:

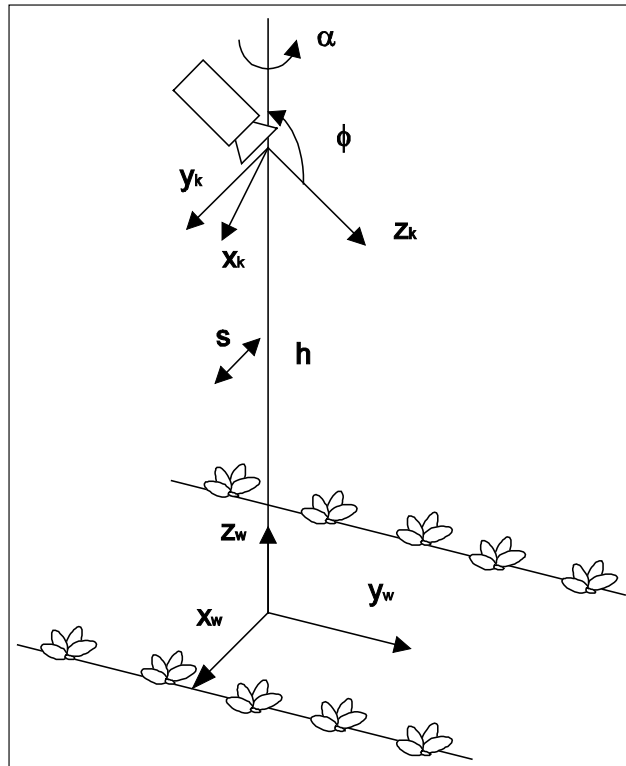


Figura 3.6: Sistema de coordenada Mundo-Câmera (ASTRAND e BAERVELDT, 2005).

$$A\alpha + Bs + C = 0 \quad (3.5)$$

e

$$A = (-hy_i \cos(\phi) - fh \sin(\phi)) \quad (3.6)$$

$$B = (f \cos(\phi) - y_i \sin(\phi)) \quad (3.7)$$

$$C = -x_w B + hx_i \quad (3.8)$$

onde α é o *heading*, s é o deslocamento lateral, x_w é a posição da linha no sistema de coordenadas do mundo, ϕ e h são respectivamente o ângulo e a altura da câmera em relação ao solo e f é a distância focal.

Uma câmera CCD (*Charge Coupled Device*) monocromática com lentes de 8,5 *mm* equipada com um filtro de infravermelho próximo (780 *nm*) foi utilizada para a captura das imagens. Um computador Pentium II 333MHz equipado com uma placa digitalizadora de vídeo Matrox Pulsar foi utilizado para o processamento das imagens. As imagens são capturadas com 384x288 *pixels* e binarizadas através de algumas operações morfológicas e através da limiarização (*thresholding*). A transformada de Hough (GONZALES e WOODS, 2000) é então aplicada à imagem binarizada para detectar as linhas de cultivo. Um controlador PID (*Proportional Integral Derivative*) foi utilizado para manter ou corrigir a trajetória da máquina agrícola.

O sistema foi instalado em um trator equipado com um cultivador utilizado no controle das ervas daninhas presentes entre as linhas de cultivo. O sistema foi testado em dois tipos de culturas: beterraba e canola. A distância entre as linhas era de 48 *cm* e a distância entre as plantas na linha era de 12 *cm* para a beterraba e de 15 *cm* para a canola. O tamanho das plantas e a iluminação variaram durante os testes.

O sistema foi capaz de processar de 8 a 12 *imagens/s* dependendo do tamanho das plantas e da densidade das ervas daninhas. A velocidade máxima atingida foi de 2 *km/h* e a precisão do sistema foi de $\pm 4,5$ *cm*. Os testes mostraram que o sistema é capaz de detectar linhas incompletas e com elevado nível de infestação por ervas daninhas (até 200 *plantas/m*²).

3.4 Considerações Finais

Diversos trabalhos envolvendo a visão computacional e o direcionamento automático de máquinas agrícolas podem ser encontrados na literatura. A maioria dos trabalhos utiliza imagens monocromáticas e necessitam de grande poder de processamento para extrair as informações necessárias ao controle automático das máquinas agrícolas.

Devido à grande quantidade de dados contidos em cada imagem é fundamental que os métodos de extração sejam simples para que os controladores funcionem adequadamente. Os trabalhos encontrados na literatura utilizam técnicas complexas para a extração das informações de posição exigindo a redução da resolução das imagens e a redução do número de quadros processados por segundo. Essas restrições dificultam o controle e limitam a velocidade de operação das máquinas agrícolas, inviabilizando a sua utilização em veículos que operam em grandes velocidades como os pulverizadores. Para que o sistema possa ser utilizado satisfatoriamente na pulverização é fundamental que não haja limitação da velocidade de operação.

O próximo capítulo destina-se à apresentação do método proposto para a extração da posição da máquina agrícola em relação às linhas de cultivo.

Capítulo 4

Método de Extração da Posição de Máquinas Agrícolas Baseado no Ponto de Fuga e na Busca Antecipada

Uma das tarefas de grande importância para o setor agrícola é a pulverização. A incorreta aplicação de fungicidas, herbicidas, adubos foliares, entre outros, pode acarretar desperdício de recursos e grandes perdas de produtividade. Os prejuízos também podem ser oriundos do amassamento das plantas causado pelas rodas do pulverizador. As plantas danificadas não se desenvolvem normalmente e podem se tornar foco de doenças.

Guiar manualmente um pulverizador sobre uma área de cultivo é uma tarefa complexa atualmente. A complexidade se deve ao aumento do tamanho das máquinas, ao alongamento das barras de pulverização e ao maior número de controles e sensores a serem manipulados e monitorados pelo operador. Muitos agricultores também contribuem para o aumento da complexidade da pulverização reduzindo a distância entre as linhas de

cultivo com o intuito de aumentar a produtividade da área. Todos esses fatores tornam o direcionamento automático vital para o sucesso da pulverização e conseqüentemente para o aumento da produtividade, para a redução do impacto ambiental e para a redução de custos. Devido aos altos preços dos dispositivos GPS de elevada precisão, a visão computacional surge como uma opção muito atraente para o direcionamento automático de máquinas agrícolas.

Um dos problemas dos métodos de extração da posição a partir de imagens das linhas de cultivo é a obtenção de um sinal preciso e estável em diferentes condições externas. As linhas normalmente possuem falhas de plantio, plantas em diferentes estágios de crescimento ao longo do campo e também podem apresentar diferentes tonalidades devido a doenças, falta de nutrientes ou escassez de água. Outro problema é a existência de ervas daninhas e de sobras da cultura anterior como palha e plantas entre as linhas. Esses fatores tornam a tarefa de reconhecimento das linhas de cultivo difícil, exigindo algoritmos complexos e conseqüentemente maior poder de processamento, elevando o tempo de resposta do sistema e, por conseguinte a redução da velocidade de operação da máquina agrícola (ASTRAND e BAERVELDT, 2005).

Um requisito importante que deve ser considerado no desenvolvimento de sistemas de direcionamento automático para máquinas agrícolas que operam em velocidades elevadas é a utilização de sensores de posição capazes de fornecer altas taxas de dados. Nos sistemas de direcionamento automático para máquinas agrícolas baseados em receptores GPS a taxa de dados de posição normalmente utilizada é de 10 *Hz* (HAGUE *et al.* , 2000; JAHNS, 1997). Portanto, para obter essa taxa de dados a partir de um *stream* de vídeo o tempo de processamento de cada imagem não deve ser superior a 100 *ms*. Em um sistema de direcionamento baseado em visão computacional esse requisito é obtido através de um método eficiente e veloz de extração da posição a partir das imagens capturadas. A redução do tempo de processamento permite aumentar a resposta dos controladores,

selecionar os melhores quadros de vídeo e utilizar imagens de alta resolução.

A seguir é proposto um método para a extração da posição de máquinas agrícolas em relação às linhas de cultivo de imagens da área cultivada. O método proposto pode ser utilizado para o desenvolvimento de sistemas de direcionamento automático para máquinas agrícolas baseados em visão computacional.

4.1 Extração da Posição

Como discutido na Seção 2.4, devido às deformações da projeção perspectiva as linhas paralelas em uma imagem convergem a um determinado ponto conhecido como ponto de fuga. Essa deformação é o item fundamental do método proposto neste trabalho.

Nas imagens de uma área cultivada, as linhas de cultivo convergem a um único ponto como mostra a Figura 4.1. A localização desse ponto na imagem fornece a posição da câmera em relação às linhas de cultivo e pode ser utilizado em um sistema de direcionamento automático como a informação de entrada do controlador responsável por manter a rota da máquina agrícola.

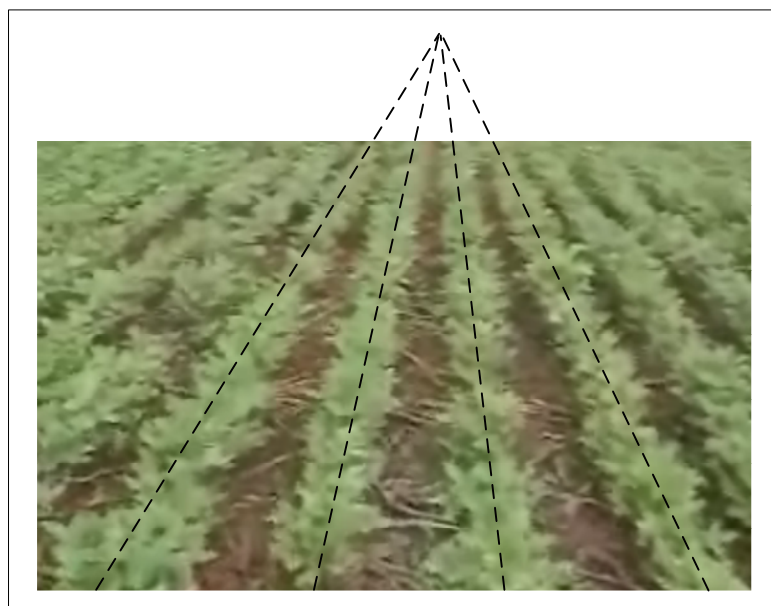


Figura 4.1: Imagem de uma área cultivada (soja).

Existem diversas técnicas para a obtenção dos pontos de fuga de uma imagem (SHUFELT, 1999; LEEMANS e DESTAIN, 2006; MINAGAWA *et al.*, 1999). Essas técnicas variam em termos de complexidade e tempo de processamento, inviabilizando, em muitos casos, a sua utilização no desenvolvimento de sistemas de direcionamento automático baseado em visão computacional.

O ponto de fuga de uma imagem semelhante à ilustrada na Figura 4.1 pode ser obtido através da soma dos *pixels* de cada coluna da imagem. O menor valor das somas realizadas indica a coluna na qual está localizado o ponto de fuga da imagem. No método proposto, o ponto de fuga está localizado na coluna central da imagem quando a câmera estiver alinhada e centrada na entrelinha de plantio; em uma das colunas à direita da coluna central da imagem quando a câmera estiver posicionada à esquerda da entrelinha de plantio ou em uma das colunas à esquerda da coluna central da imagem quando a câmera estiver posicionada à direita da entrelinha de plantio.

A Figura 4.2 apresenta três imagens de uma cultura de soja obtidas com uma câmera digital. A câmera foi posicionada em relação ao eixo vertical, perpendicular ao plano horizontal, em três posições: à esquerda, ao centro e à direita da entrelinha de plantio selecionada. Os respectivos pontos de fuga calculados são apresentados nos gráficos subsequentes e representam a soma dos *pixels* das colunas das imagens.

O menor valor de cada gráfico da Figura 4.2 indica a coluna na qual estão localizados os pontos de fuga correspondentes. O valor do *pixel* utilizado no somatório foi obtido através da seguinte equação:

$$P = 2 * G - R - B \quad (4.1)$$

onde P é o valor do *pixel* e R , G e B são respectivamente as componentes vermelho, verde e azul do espaço de cor RGB. Essa equação permite realçar as plantas verdes e é

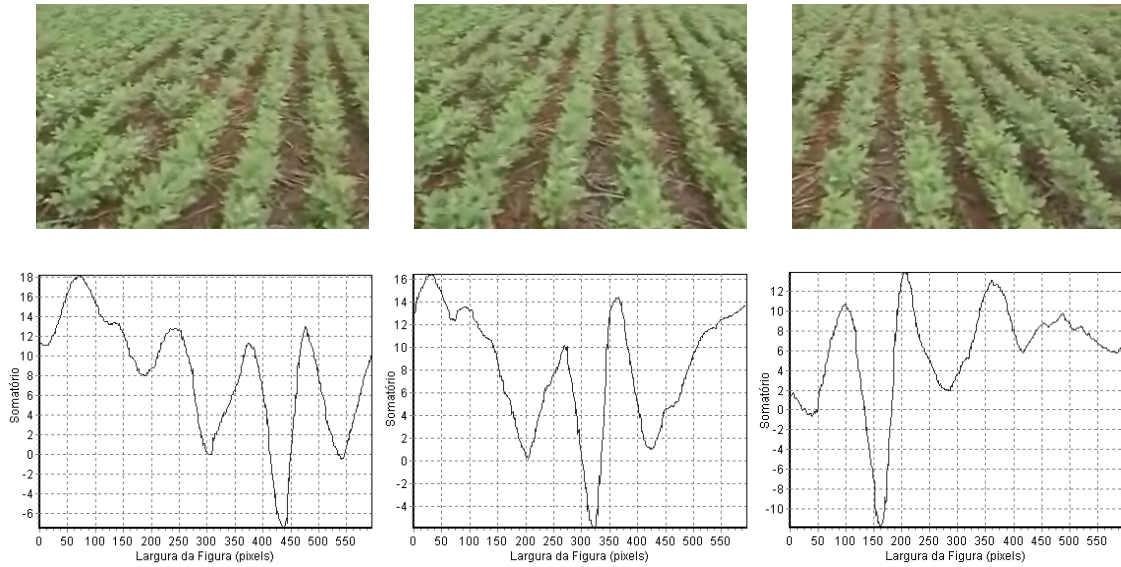


Figura 4.2: Diferentes localizações do ponto de fuga.

relativamente insensível às variações de luminosidade e à sombras (WOEBBECKE *et al.*, 1995). O realce do verde proporcionado e a simplicidade são as principais razões para a utilização dessa equação, contribuindo, portanto, no objetivo do método proposto que é a redução do tempo de processamento de cada imagem e conseqüentemente o aumento da taxa de dados.

A obtenção do ponto de fuga a partir da soma de *pixels* também reduz alguns problemas que geralmente estão presentes nas imagens de uma área cultivada. Imagens contendo falhas de plantio e ervas daninhas entre as linhas podem ser utilizadas diretamente no método proposto sem a necessidade de calibrações ou ajustes prévios, como mostram os testes estáticos preliminares ilustrados na Figura 4.3.

4.2 Posição da Câmera

Para maximizar a distorção gerada pela projeção perspectiva é necessário posicionar a câmera de forma a obter cenas distantes da posição atual. Esse posicionamento, conseqüentemente, acrescenta ao método proposto as características de aumento de esta-

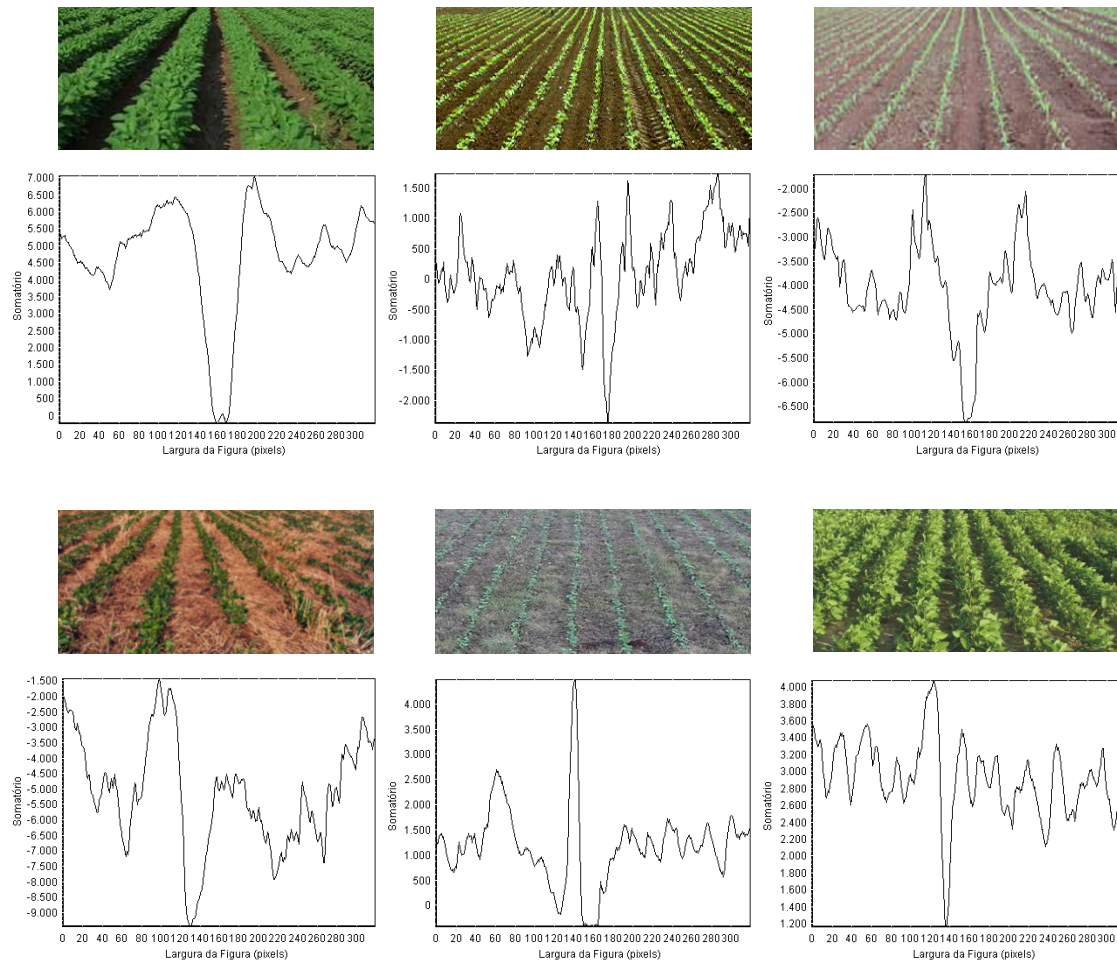


Figura 4.3: Imagens de culturas e os gráficos de soma.

bilidade descritas na Seção 2.3, enquadrando-o como um método de busca antecipada.

O número da coluna de *pixels* da imagem que contém o menor valor das somas realizadas, o ponto de fuga, pode ser utilizado para estimar a distância futura da câmera em relação ao centro da entrelinha de plantio. Testes iniciais mostraram que para ângulos de movimento inferiores a 20° é possível assumir a diferença entre o número da coluna do ponto de fuga e a coluna que representa o centro da imagem como a distância do centro da câmera, em *pixels*, em relação ao centro da entrelinha. Esta distância representa o erro lateral futuro e pode ser utilizado como o *set point* de um controlador responsável por gerar as correções necessárias para manter uma máquina agrícola em sua trajetória.

Os teste iniciais, citados anteriormente, foram realizados com o intuito de validar

o método proposto para a obtenção da posição da câmera em relação a entrelinha de plantio. Para a realização dos testes foram fixadas em bancada 4 folhas de papel A4 de gramatura 180 g/m^2 e de cor palha para simular o solo. Para representar as linhas de cultivo foram traçadas sobre o papel, com uma caneta marca texto verde, linhas paralelas eqüidistantes 1 cm umas das outras como mostra a Figura 4.4. Para a captura das imagens foi utilizada uma *webcam* Genius modelo VideoCAM GF112 capaz de capturar imagens a uma resolução máxima de $352 \times 288 \text{ pixels}$ a 30 quadros por segundo. A câmera utilizada foi fixada em uma régua de 20 cm de comprimento para facilitar a movimentação e a medição dos deslocamentos realizados.

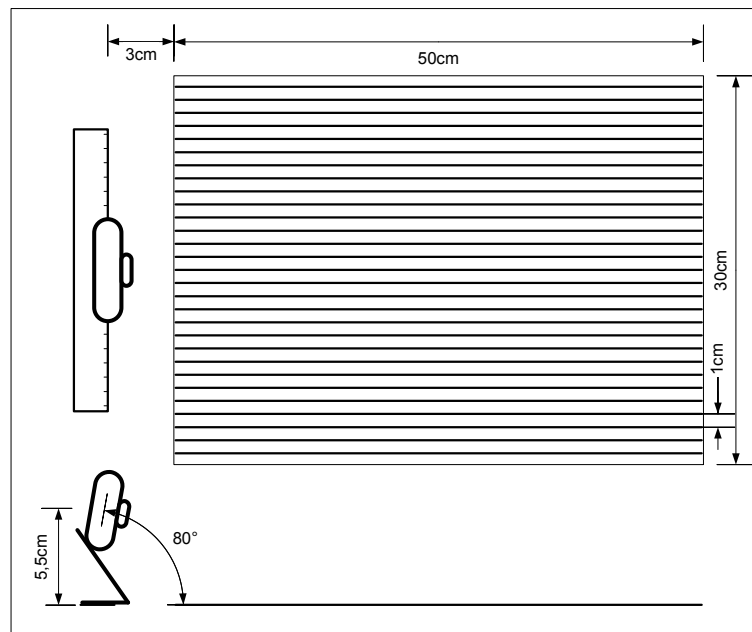


Figura 4.4: Experimento de validação.

A Figura 4.5 ilustra três posições da câmera em relação a linha central para demonstrar a possibilidade de obtenção da posição: 5 cm à esquerda da linha central (a), sobre a linha central (b) e 5 cm à direita da linha central (c). A câmera foi deslocada perpendicularmente em relação às linhas traçadas. O ângulo de rotação sob o eixo vertical foi mantido em zero nos três casos. Nesse experimento foi obtido um deslocamento de 5 pixels em ambas as direções.

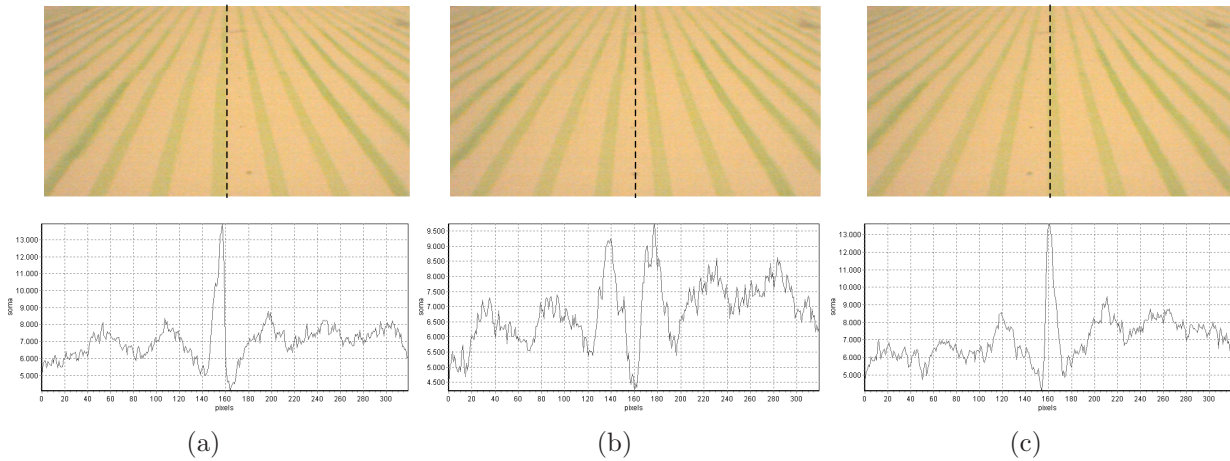


Figura 4.5: Diferentes localizações do ponto de fuga - câmera deslocada.

A Figura 4.6 ilustra duas posições da câmera em relação a linha central: 10° de rotação para a direita sobre a linha central (a) e 10° de rotação para a direita e deslocada 5 cm perpendicularmente à esquerda da linha central (b). Nesse experimento também foi obtido uma diferença de 5 *pixels* entre os deslocamentos.

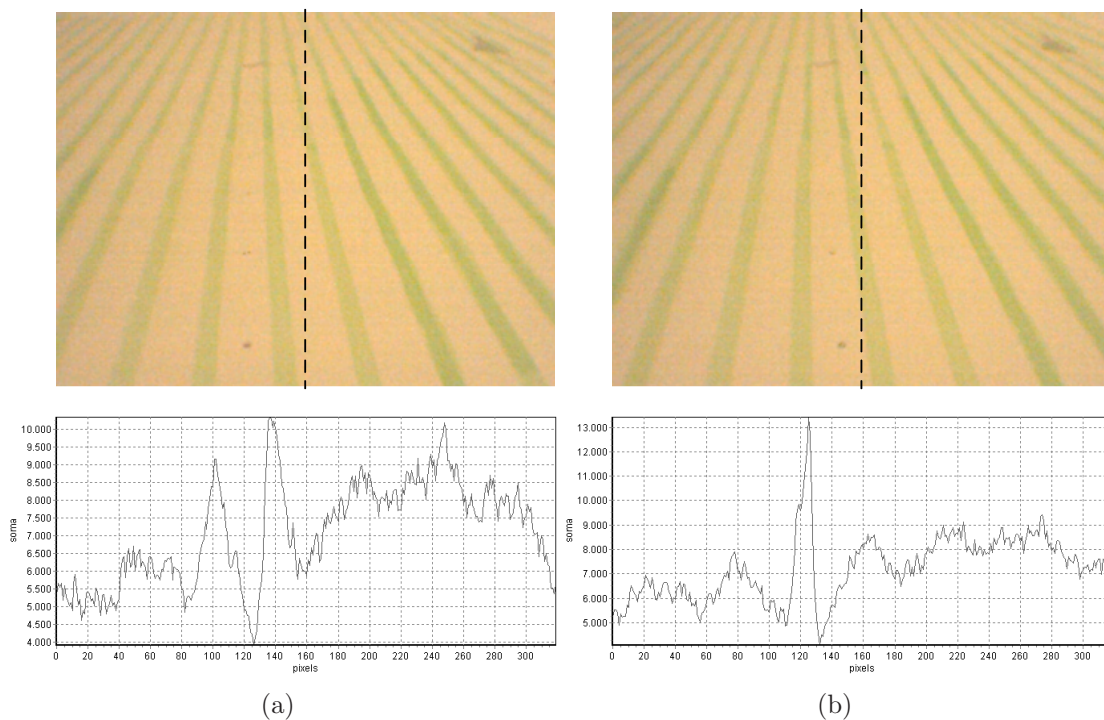


Figura 4.6: Diferentes localizações do ponto de fuga - câmera deslocada e rotacionada.

Alguns experimentos também foram realizados com o intuito de avaliar o comportamento do método proposto no processamento de imagens contendo falhas de plantio,

sombras e ervas daninhas. Para simular a presença de ervas daninhas foram espalhados sobre as folhas de papel pequenos quadrados de $0,5\text{ cm}$ de lado pintados na cor verde. Os resultados podem ser observados na Figura 4.7.

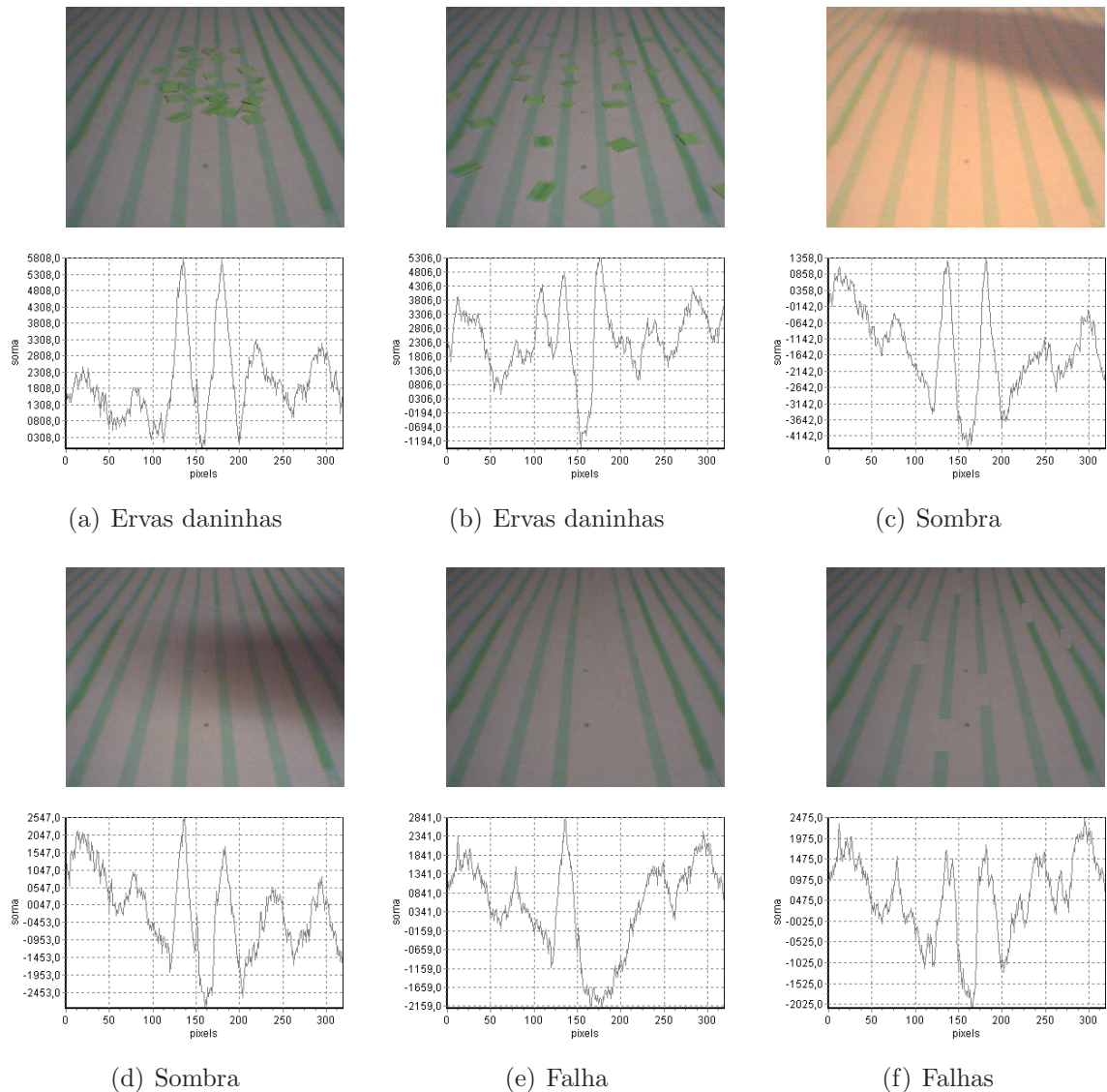


Figura 4.7: Experimentos realizados.

O processamento das imagens descrito anteriormente foi realizado em um computador AMD Athlon XP 2600+ com 512 MB de memória RAM a um *clock* de 2.07 GHz . O software de processamento foi desenvolvido na ferramenta RAD (*Rapid Application Development*) Delphi da Borland para a plataforma Windows XP. Os tempos de processamento obtidos foram de 4 ms para imagens de $320 \times 240\text{ pixels}$ codificada no formato BMP com 24bits por *pixel*.

4.3 Pré-Processamento

Diversos problemas podem dificultar a identificação das linhas de cultivo em uma imagem da área cultivada. Descritos no início deste capítulo, esses problemas direcionaram os esforços iniciais de desenvolvimento deste trabalho para a busca de um método de pré-processamento de imagens digitais. Devido à necessidade de tempos de resposta reduzidos e a variedade e a complexidade das imagens, foi avaliado o desempenho das redes neurais artificiais nessa tarefa.

As redes PCNN, descritas nos Capítulo 2, têm despertado grande interesse em diversas áreas relacionadas ao processamento de imagens digitais devido à simplicidade e principalmente por não necessitarem de treinamento prévio. Com o intuito de tornar mais robusto o método de extração da posição proposto neste trabalho, foi avaliado o desempenho de uma rede ICM no pré-processamento das imagens das linhas de cultivo. O modelo simplificado ICM foi utilizado devido ao reduzido tempo de processamento quando comparado ao modelo completo.

As redes PCNN produzem como saída uma matriz de dimensões semelhantes a da imagem utilizada como entrada. Cada célula da matriz representa a saída de um neurônio e pode conter o valor 1 para representar a sua ativação ou o valor 0 para o caso contrário. Para adequar o método proposto aos dados de saída da PCNN, a obtenção do ponto de fuga deve ser realizada através da contagem dos neurônios ativados em cada coluna da matriz de saída.

O valor do *pixel* apresentado a rede PCNN foi calculado através da equação 4.1 e normalizado entre 0 e 1. Os primeiros testes, ilustrados na Figura 4.8, foram realizados para a obtenção do número de iterações necessárias e dos melhores parâmetros para a rede.

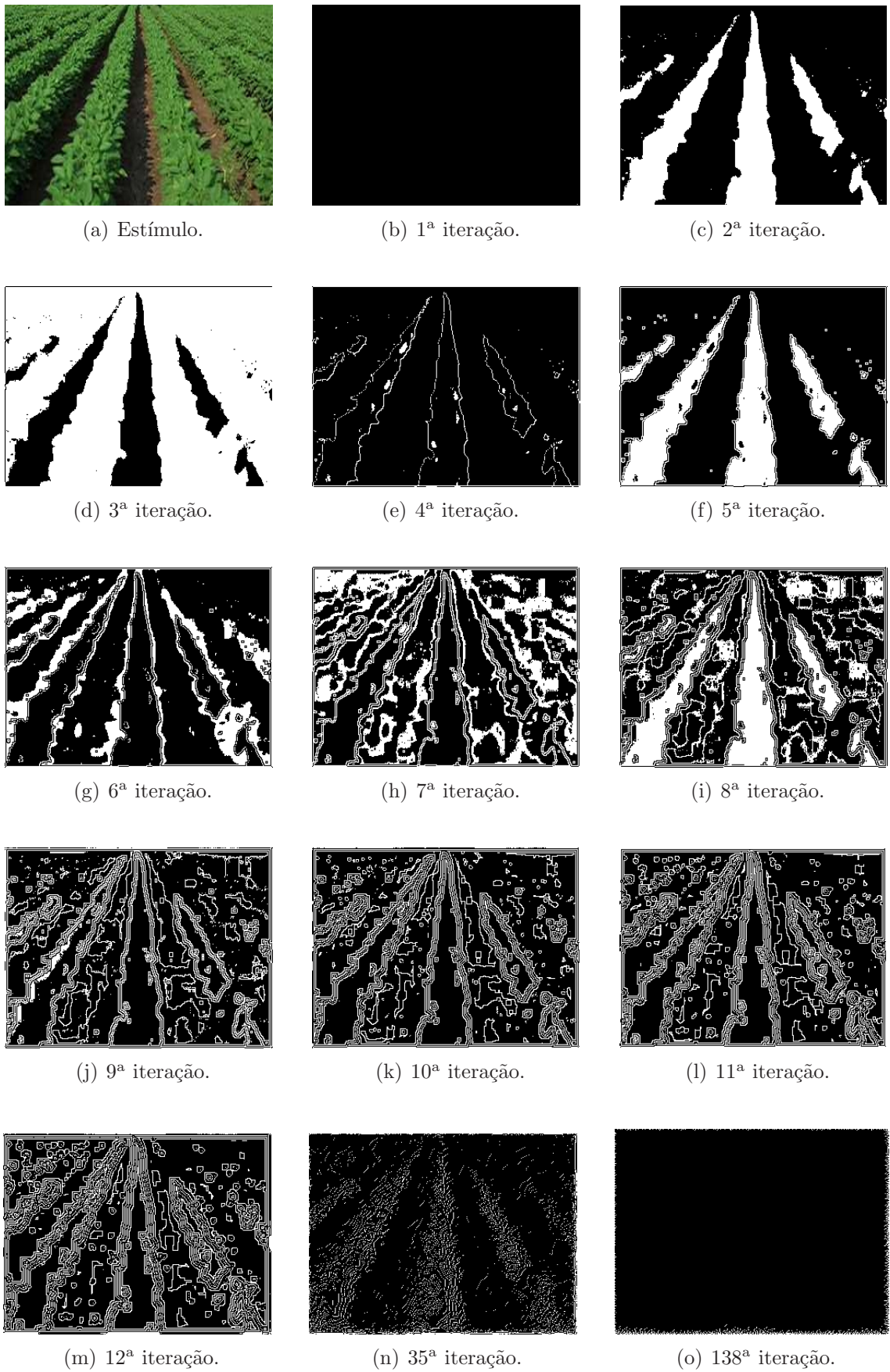


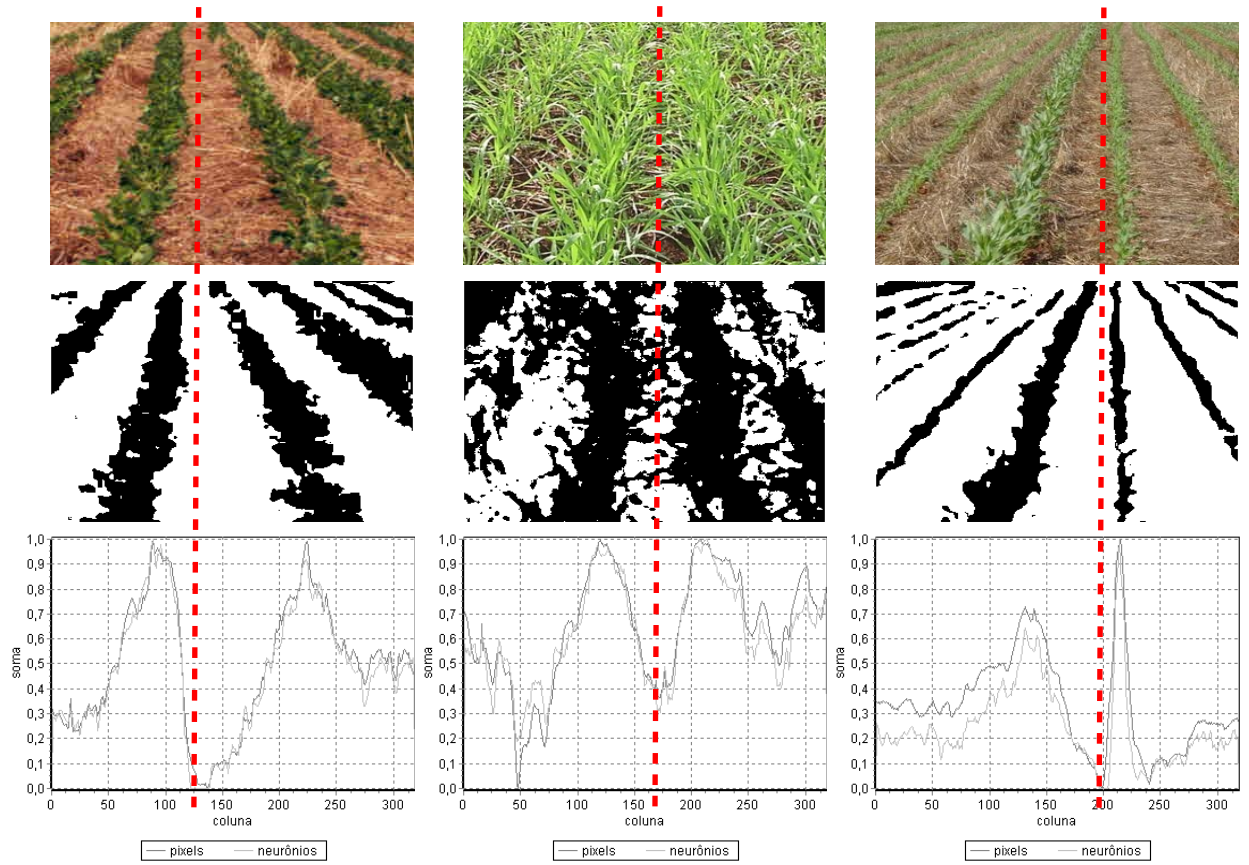
Figura 4.8: Imagem original e os resultados em diferentes iterações.

Os testes iniciais foram conduzidos de forma a se obter o menor número de iterações para o pré-processamento das imagens. O tempo de cada iteração foi de 23 *ms* para imagens de 320x240 *pixels* utilizando o computador descrito na seção anterior. Observou-se nesses testes que variando apenas o parâmetro β do modelo ICM é possível alcançar resultados satisfatórios em qualquer tipo de imagem com apenas 2 iterações.

Testes posteriores foram realizados para avaliar o desempenho da rede ICM com diferentes valores de entrada. Quatro valores de *pixel* foram avaliados: H, S, I do espaço de cor HSI e o calculado pela equação 4.1. Os resultados podem ser observados nas Figuras 4.9, 4.10, 4.11 e 4.12 onde as linhas vermelhas tracejadas representam os pontos de fuga das imagens. Os gráficos apresentam a soma dos valores dos *pixels* da imagem original e a soma dos neurônios ativados. Os parâmetros da rede ICM estão na Tabela 4.1.

Tabela 4.1: Parâmetros da rede ICM e a quantidade de neurônios ativados.

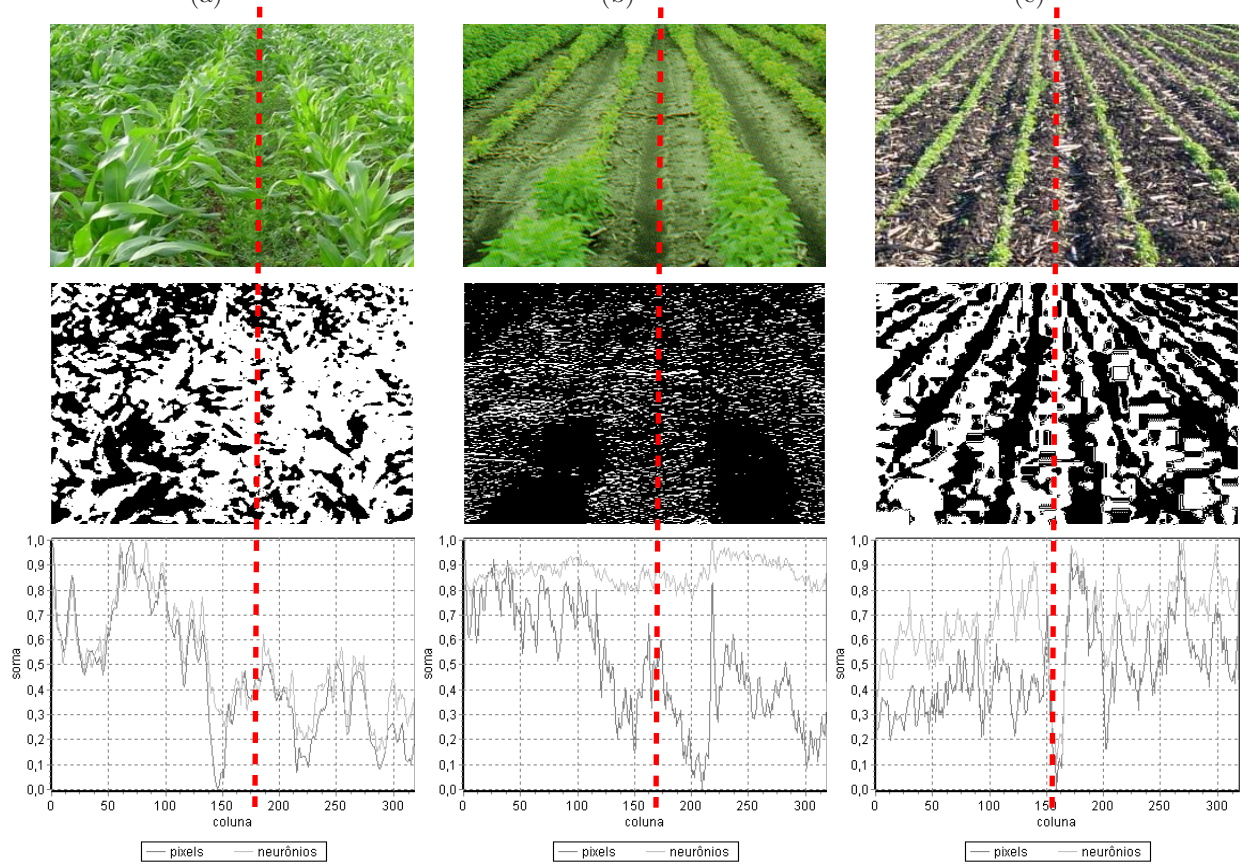
Imagem	H				S				I				2 * G - R - B			
	IF	IG	IH	Pulsos	IF	IG	IH	Pulsos	IF	IG	IH	Pulsos	IF	IG	IH	Pulsos
(a)	0,90	0,89	8,89	27086	0,90	0,89	9,50	20394	0,90	0,89	9,60	26351	0,90	0,89	9,64	28972
(b)	0,90	0,89	9,06	47167	0,90	0,89	9,10	60246	0,90	0,89	9,60	49649	0,90	0,89	9,86	19541
(c)	0,90	0,89	9,00	16348	0,90	0,89	9,20	20998	0,90	0,89	9,60	43610	0,90	0,89	9,65	31473
(d)	0,90	0,89	9,17	27745	0,90	0,89	10,00	13098	0,90	0,89	9,70	29891	0,90	0,89	9,95	13411
(e)	0,90	0,89	9,05	64933	0,90	0,89	9,50	34091	0,90	0,89	9,70	24990	0,90	0,89	9,88	26781
(f)	0,90	0,89	9,01	39694	0,90	0,89	8,95	20368	0,90	0,89	9,80	19031	0,90	0,89	9,68	15177
(g)	0,90	0,89	9,40	38783	0,90	0,89	8,75	24784	0,90	0,89	9,70	10064	0,90	0,89	9,66	8090
(h)	0,90	0,89	8,97	44928	0,90	0,89	9,10	20591	0,90	0,89	9,80	19755	0,90	0,89	9,67	41794
(i)	0,90	0,89	9,00	46910	0,90	0,89	9,10	55341	0,90	0,89	9,10	58947	0,90	0,89	9,66	45260
(j)	0,90	0,89	8,90	25853	0,90	0,89	8,90	12711	0,90	0,89	9,75	15056	0,90	0,89	9,64	13720
(k)	0,90	0,89	9,60	53848	0,90	0,89	8,80	30720	0,90	0,89	10,00	5035	0,90	0,89	9,70	7332
(l)	0,90	0,89	9,14	57860	0,90	0,89	9,30	43823	0,90	0,89	9,00	58274	0,90	0,89	9,75	48752



(a)

(b)

(c)



(d)

(e)

(f)

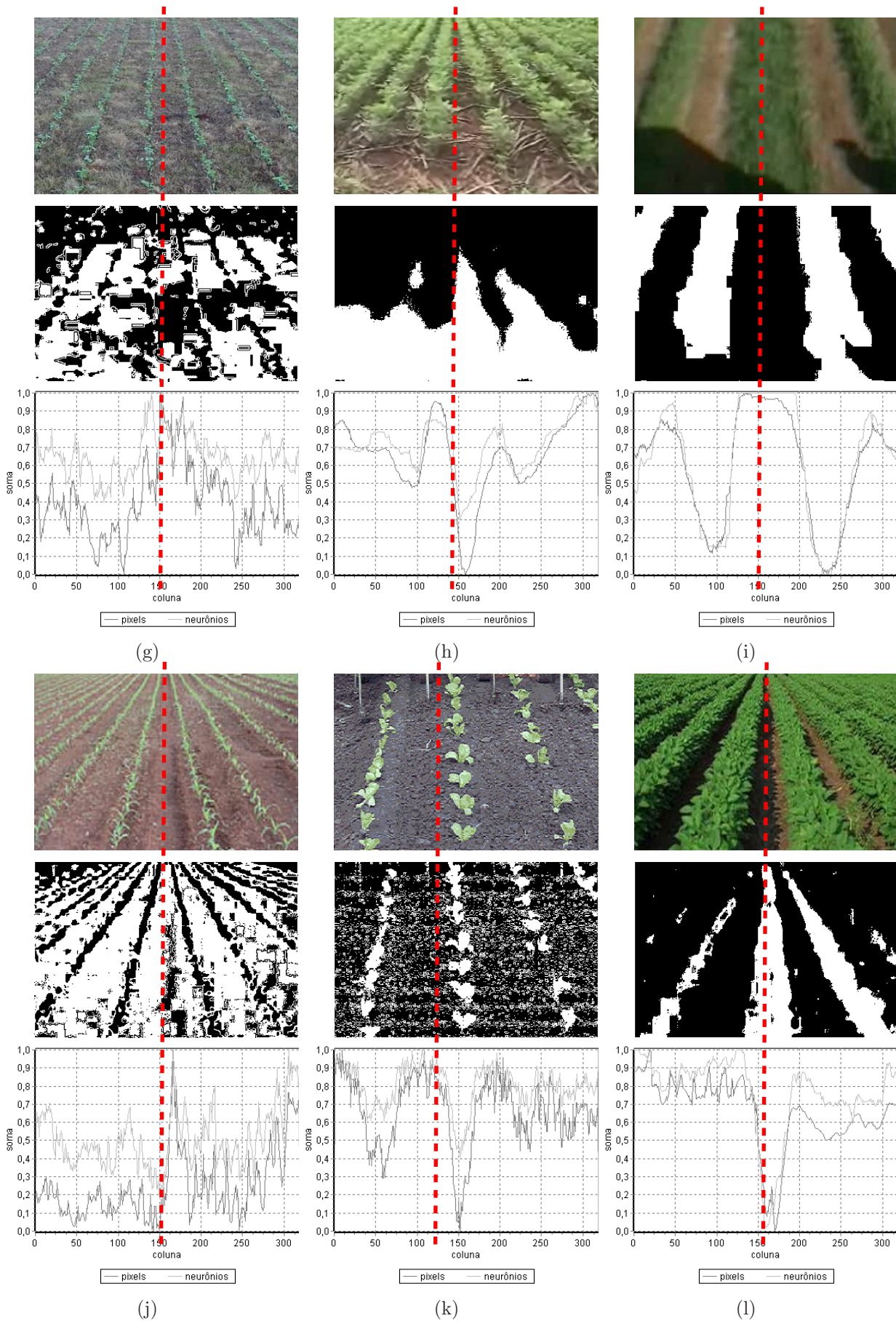
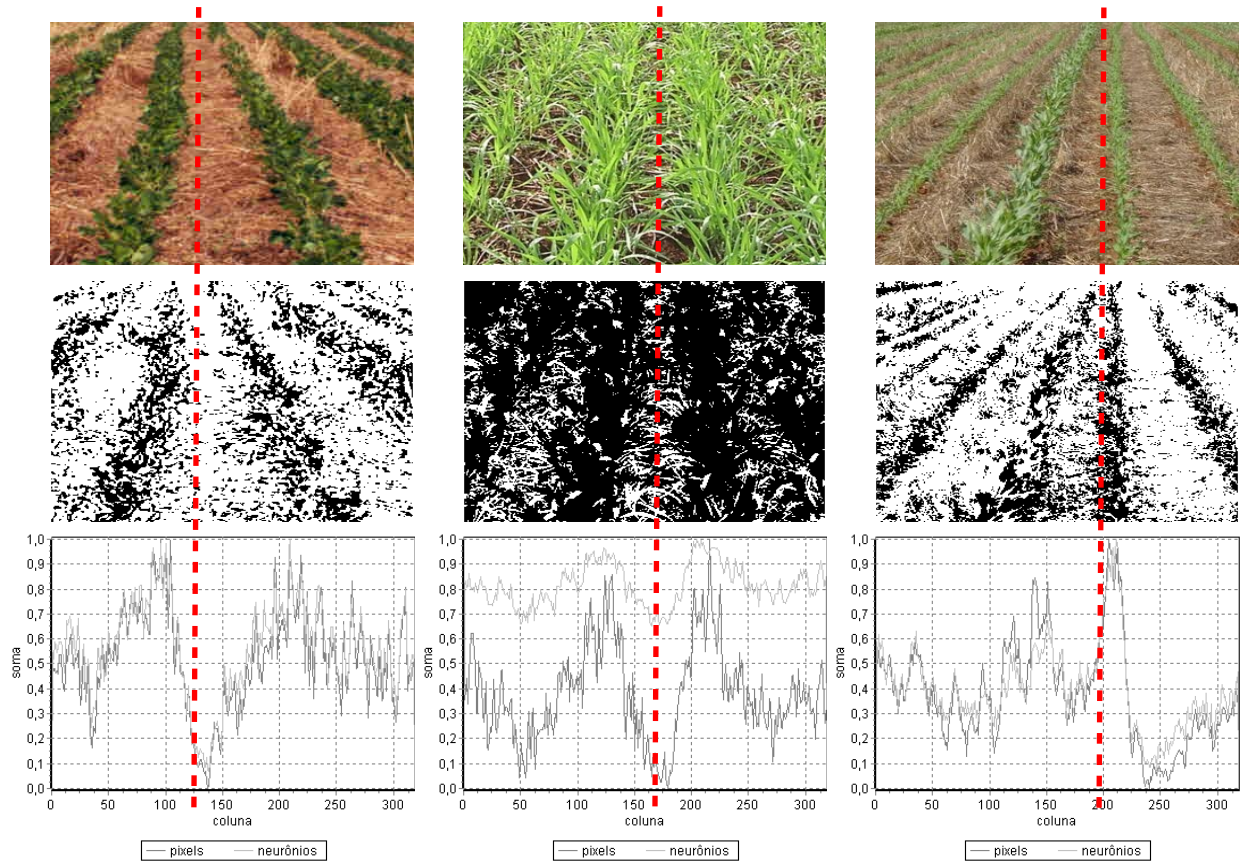


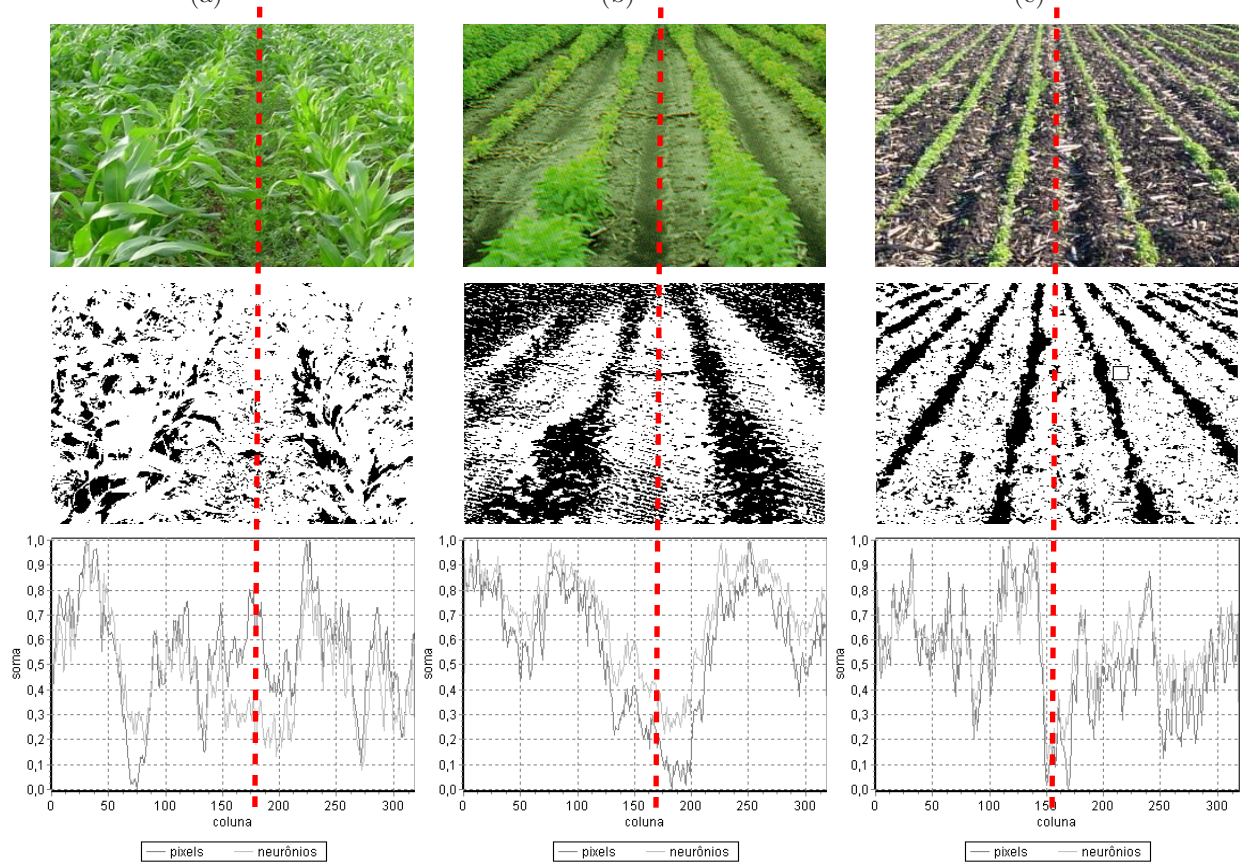
Figura 4.9: Imagens originais, imagens resultantes e os gráficos de soma - canal H do modelo HSI.



(a)

(b)

(c)



(d)

(e)

(f)

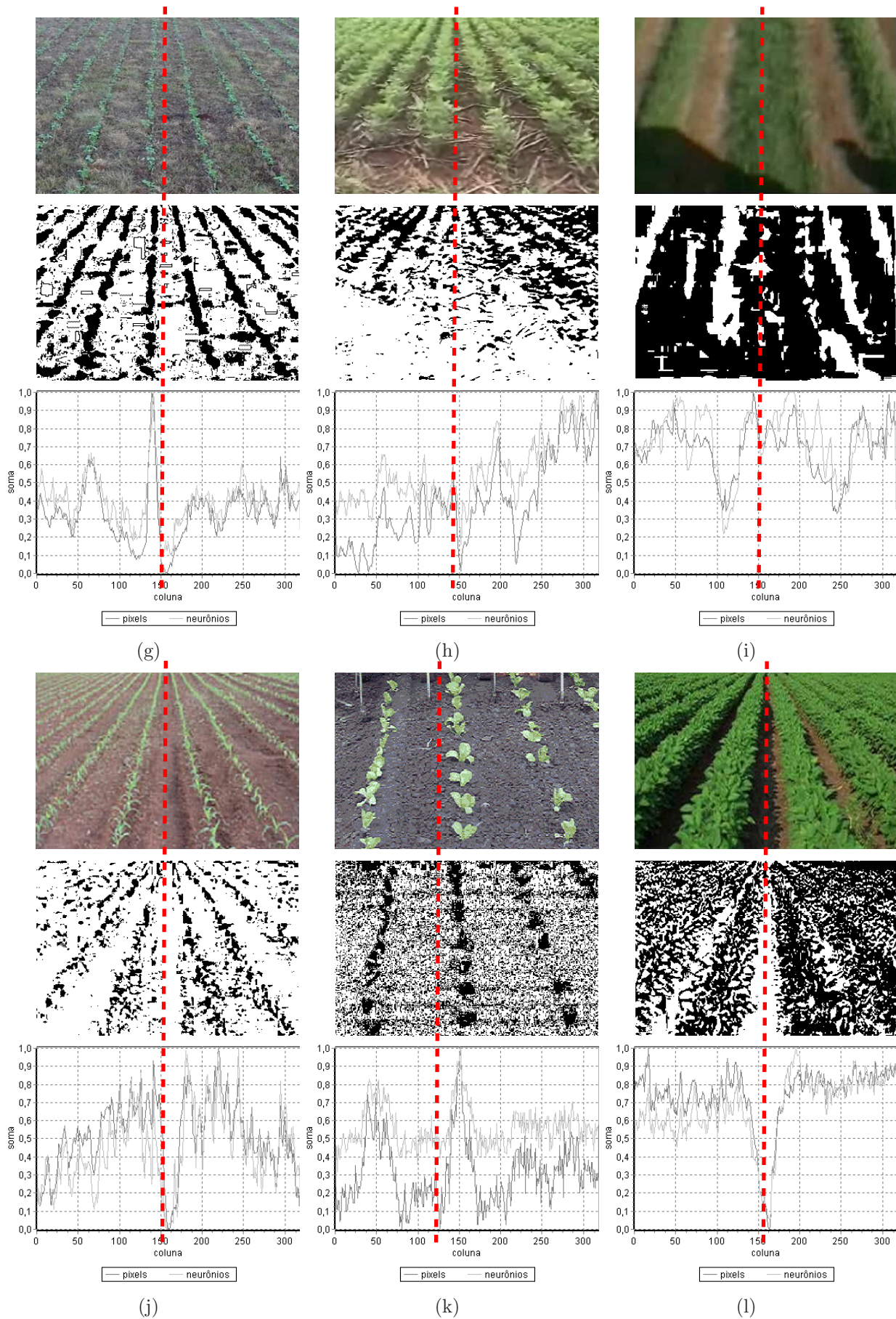
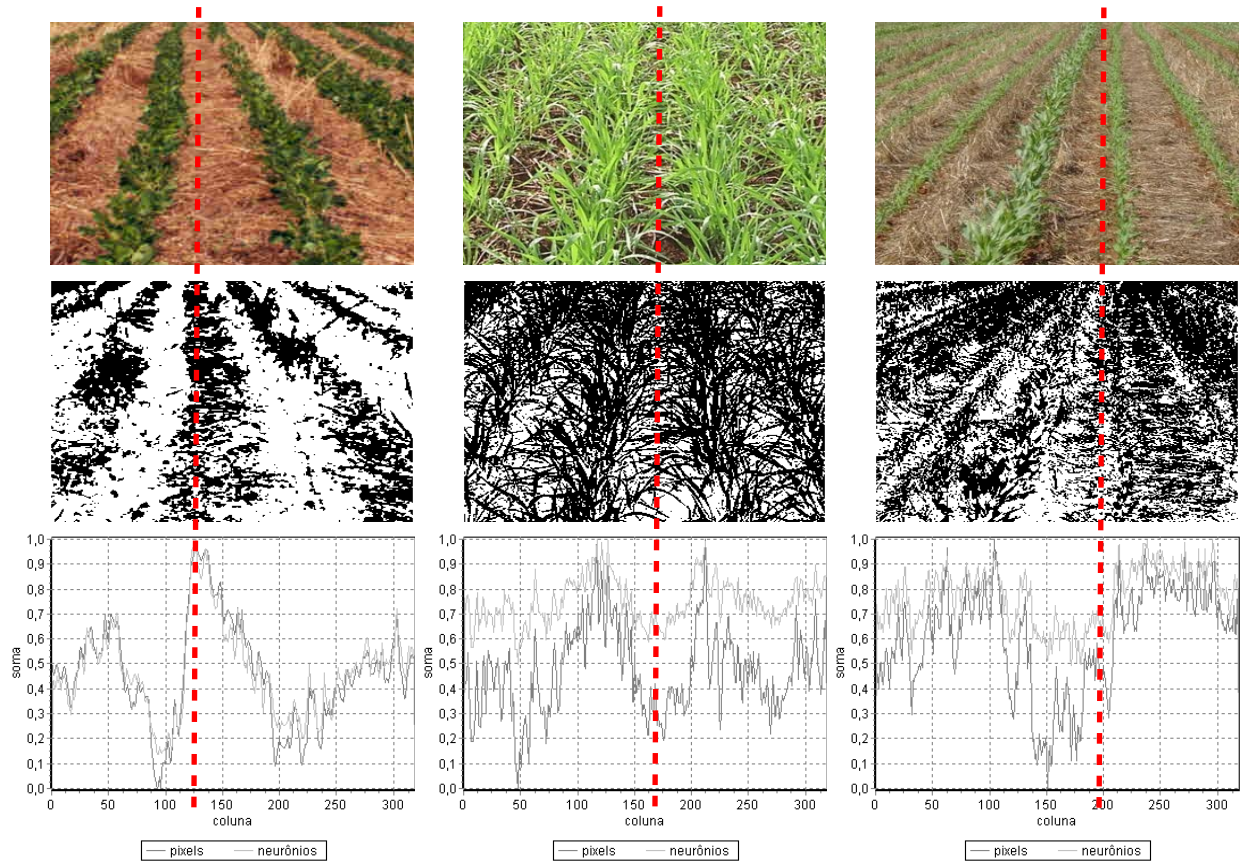


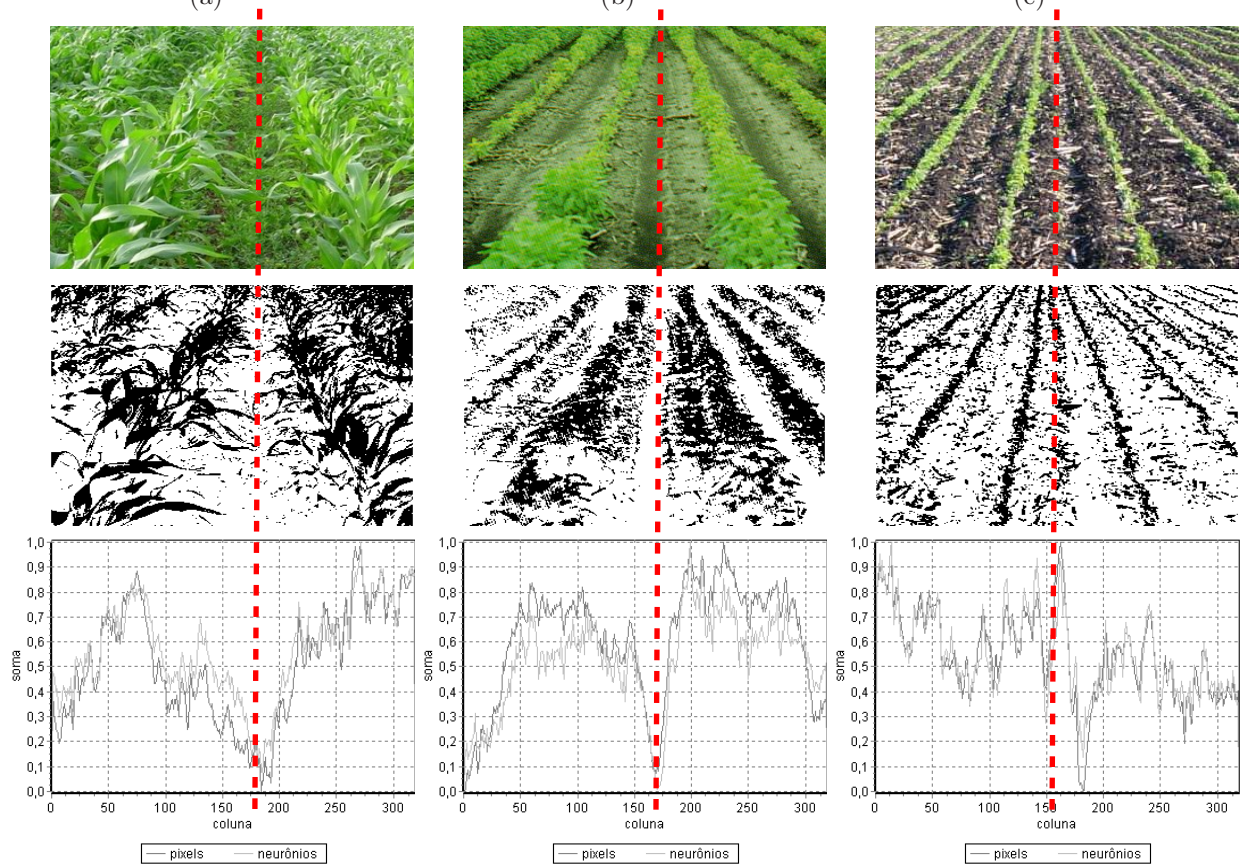
Figura 4.10: Imagens originais, imagens resultantes e os gráficos de soma - canal S do modelo HSI.



(a)

(b)

(c)



(d)

(e)

(f)

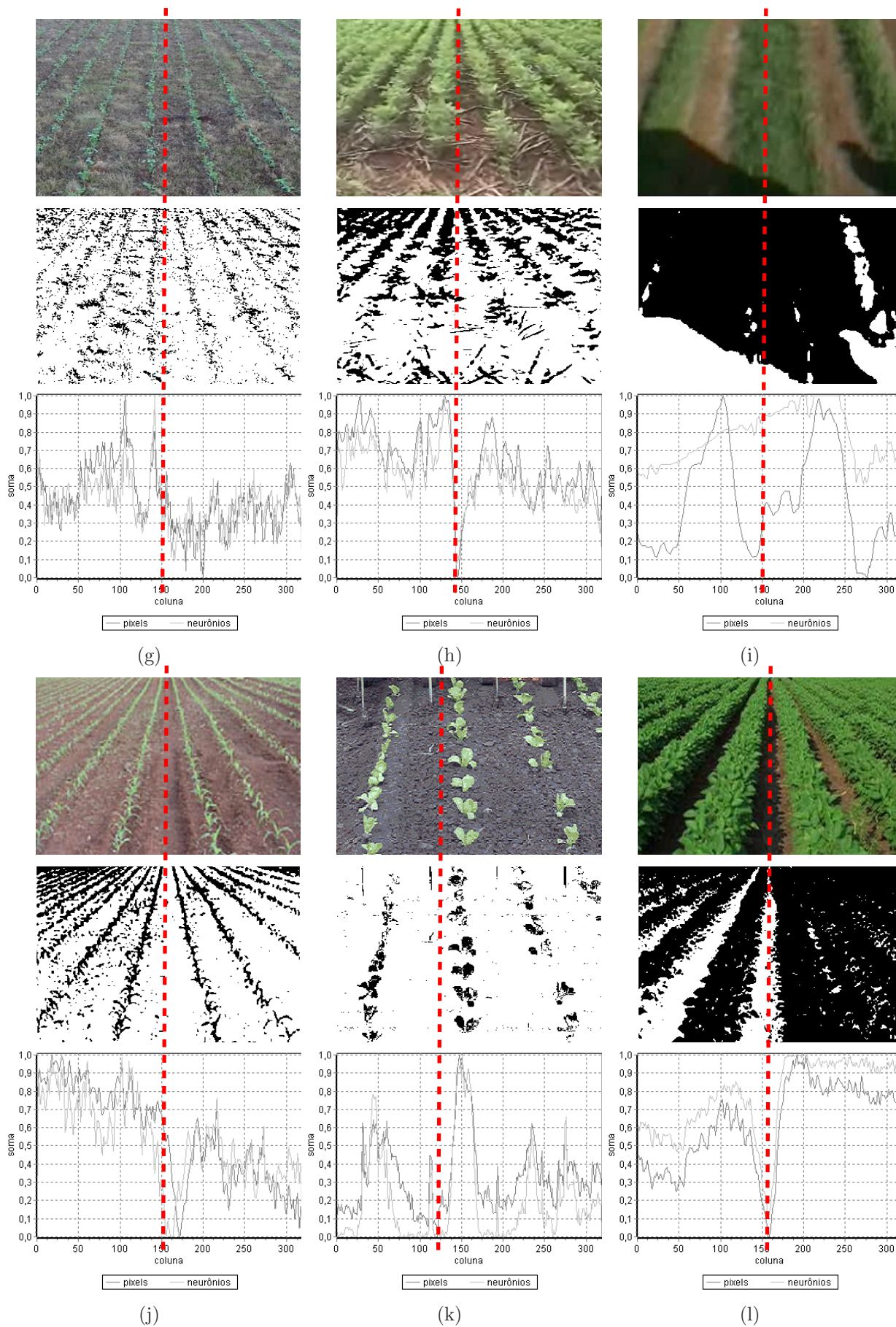
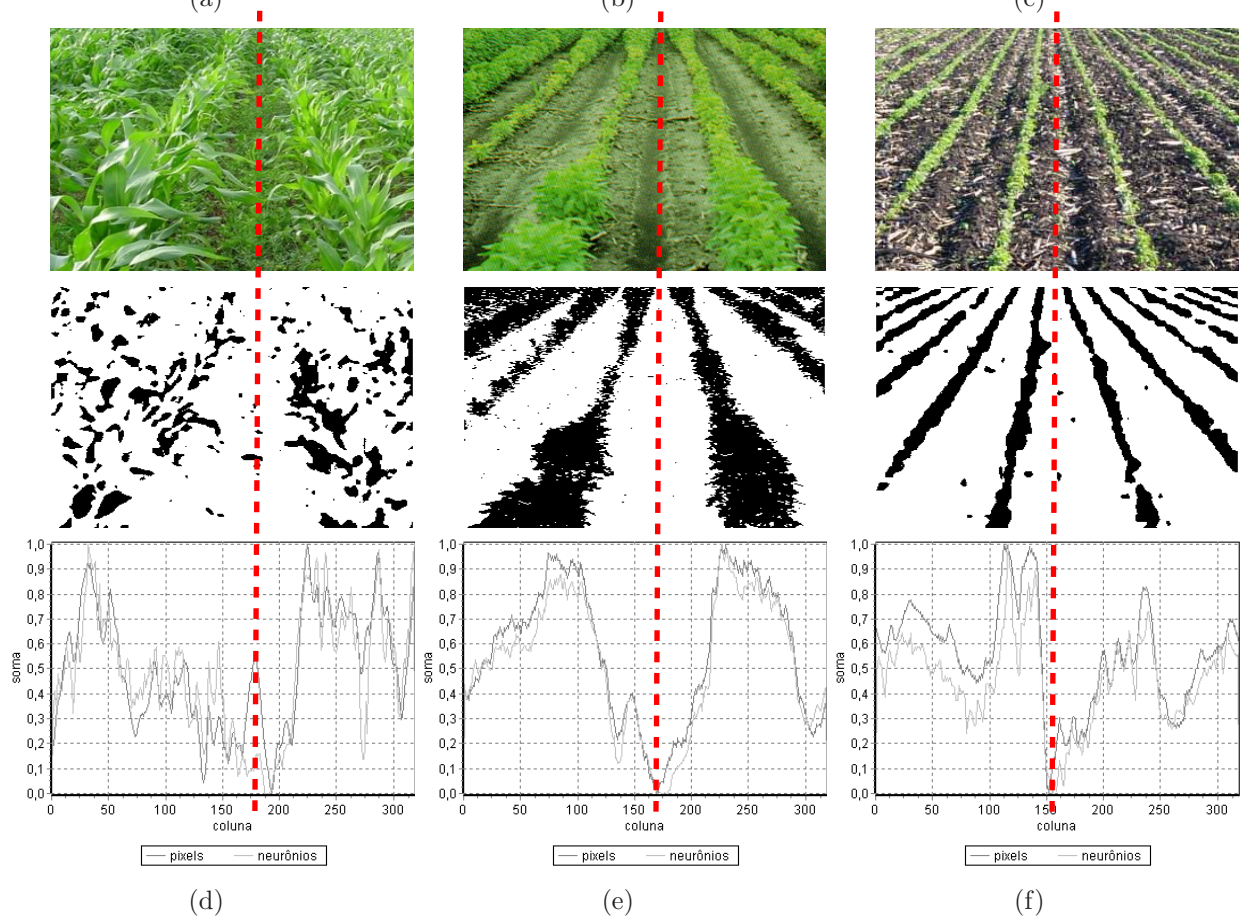
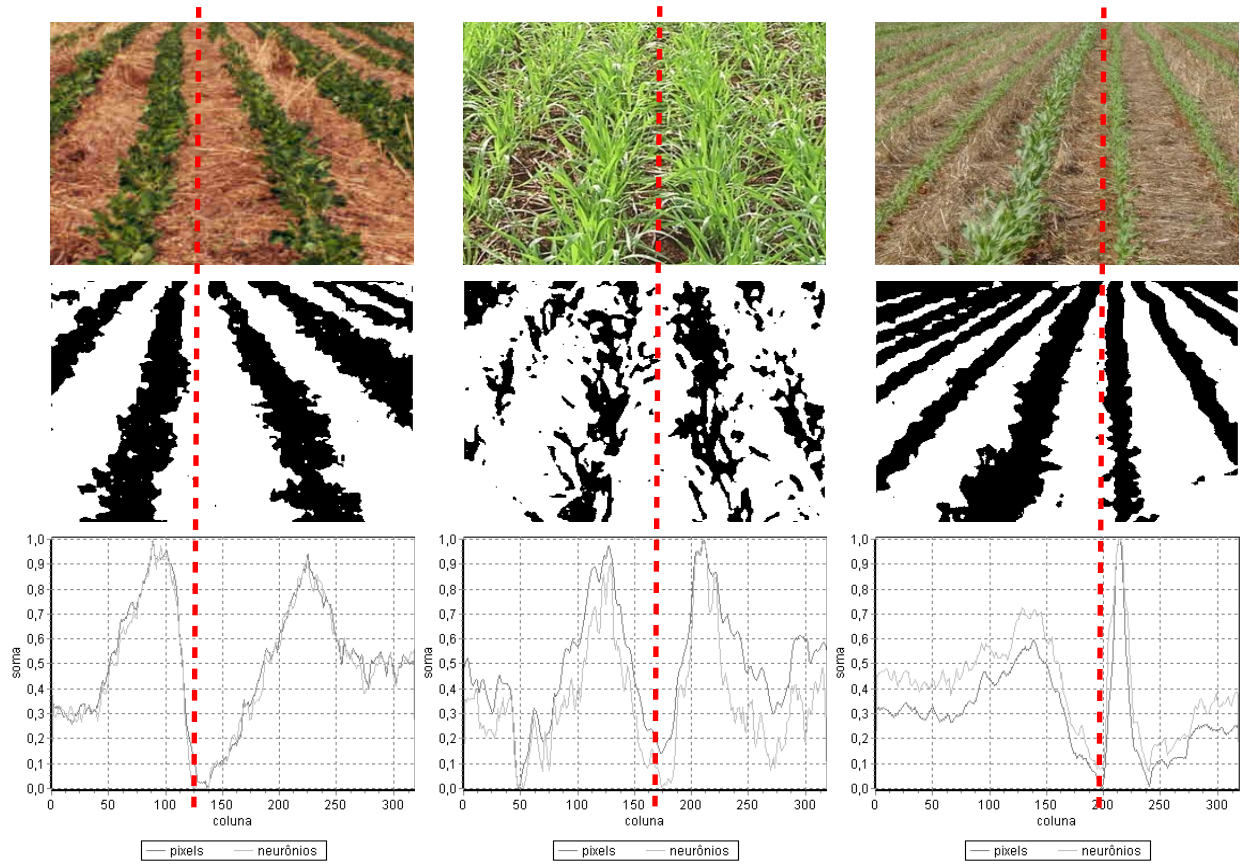


Figura 4.11: Imagens originais, imagens resultantes e os gráficos de soma - canal I do modelo HSI.



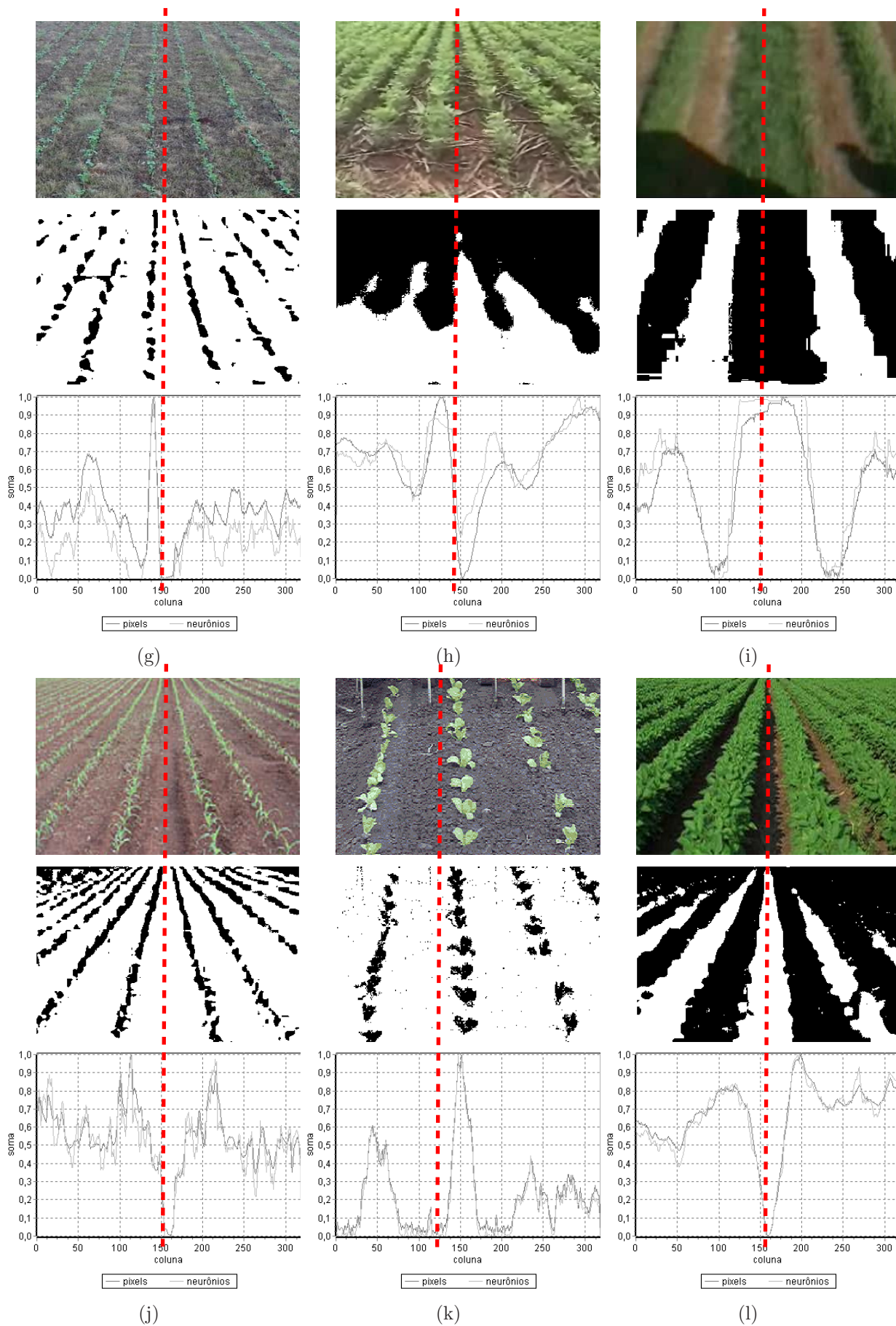


Figura 4.12: Imagens originais, imagens resultantes e os gráficos de soma - 2^*G-R-B .

A Tabela 4.2 apresenta uma classificação dos resultados obtidos com a rede ICM. Essa classificação tem como objetivo determinar o valor de *pixel* mais adequado para ser utilizado como entrada da rede ICM. Os valores contidos em cada célula da tabela representam a distância em *pixels* do ponto de fuga da imagem em relação à coluna contendo o menor valor das somas dos *pixels*. Nessa classificação valores pequenos indicam bons resultados por representarem maior proximidade entre o ponto de fuga real e o calculado através do método proposto neste trabalho. Os valores das imagens (g), (i) e (k) não estão contidos na tabela por não ser possível determinar corretamente o ponto de fuga dessas imagens.

Tabela 4.2: Comparação dos resultados obtidos com a rede ICM.

Imagem	Diferença (pixels)			
	H	S	I	2*G - R - B
(a)	0	12	31	0
(b)	120	3	118	118
(c)	0	53	25	43
(d)	110	94	6	8
(e)	30	10	0	0
(f)	3	3	31	0
(g)				
(h)	12	12	0	6
(i)				
(j)	0	3	6	0
(k)				
(l)	0	6	0	0
Total:	275	196	217	175

Somando-se os resultados de cada experimento obtêm-se aos valores 275 para a componente H , 196 para a componente S , 217 para a componente I e 175 para a equação 4.1. Esses valores mostram que o valor de *pixel* calculado pela equação 4.1 apresenta os melhores resultados quando comparado com as componentes do espaço de cor HSI.

4.4 Conclusões

O método de extração da posição de imagens da área cultivada proposto neste trabalho apresentou resultados satisfatórios nos testes de validação realizados. Os tempos de

processamento obtidos também demonstram que é possível obter elevadas taxas de dados, permitindo a utilizá-lo em máquinas agrícolas que operam em velocidades elevadas. O método proposto também apresentou resultados muito bons no processamento de imagens contendo falhas de plantio, sombras e ervas daninhas. Também foi demonstrado que o tamanho e o tipo de planta não afeta o processamento tornando desnecessários ajustes ou calibrações.

Os testes de pré-processamento utilizando redes PCNNs também apresentaram bons resultados. Os melhores resultados foram obtidos com valores de entrada calculados através da equação 4.1. É possível observar, nesse caso, a eliminação de sombras em algumas das imagens. Apesar dos testes com as componentes do espaço de cor HSI terem apresentado resultados inferiores aos testes realizados com a equação 4.1, foram observados resultados interessantes no processamento de imagens contendo sombras. Esses resultados podem ser visualizados em algumas imagens processadas utilizando a componente I do referido espaço de cor.

Os resultados obtidos com a soma dos neurônios ativados se assemelharam muito aos resultados obtidos com a soma dos valores dos *pixels*. No entanto, para o processamento de cada imagem é necessário no mínimo 23 ms sendo muito superior aos 4 ms obtidos com a soma dos *pixels*. Apesar de ser possível processar imagens contendo diferentes culturas com apenas 2 iterações não foi possível obter um único parâmetro satisfatório para toda a classe de imagens, o que exige uma etapa de calibração.

Os testes realizados não contemplaram problemas como a inclinação da câmera ocasionada por ondulações no solo. Apesar desse problema alterar a localização do ponto de fuga, observações realizadas em campo mostraram que a inclinação da máquina agrícola ocorre esporadicamente e em um intervalo de tempo muito curto, podendo ser facilmente eliminado com a utilização de alguma técnica de filtragem. Vale ressaltar que o método proposto foi idealizado para ser inicialmente utilizado em máquinas agrícolas utilizadas

na pulverização de regiões relativamente planas como a região oeste do Mato Grosso do Sul.

Para aumentar a confiabilidade do método proposto pode-se posicionar uma câmera sobre o eixo vertical de cada roda dianteira da máquina agrícola. Esta configuração adiciona redundância ao sistema de controle permitindo que o mesmo possa selecionar a melhor informação de posição a ser utilizada.

O próximo capítulo dedica-se a apresentação da plataforma de teste do método proposto e do sistema AGpilot[®]. São apresentados os módulos e a descrição detalhada de suas funções.

Capítulo 5

Plataforma de Testes

O plantio direto tem se tornado um método de cultivo muito utilizado pelos produtores agrícolas principalmente em regiões tropicais e subtropicais. Esse método traz grandes benefícios às áreas agrícolas por melhorar as características físico-químicas e biológicas do solo devido à retenção da matéria orgânica. O acúmulo de material orgânico reduz a erosão, aumenta a infiltração de água e diminui a compactação do solo (MACHADO e SILVA, 2001; MADARI *et al.* , 2005; ACCIARESI *et al.* , 2003).

Entretanto, realizar o plantio sobre os restos da cultura anterior exige a aplicação de herbicidas antes do plantio para eliminar as ervas daninhas e as plantas da cultura anterior. Essa operação, conhecida como dessecação, é essencial nesse método de cultivo para eliminar a competição por nutrientes entre a nova cultura e as outras plantas. É extremamente importante que a dessecação seja realizada com o mínimo de sobreposição e de falhas para reduzir o impacto ambiental e os custos de aplicação (MACHADO e SILVA, 2001; MADARI *et al.* , 2005).

A dessecação de uma área sem sobreposições ou falhas é uma tarefa difícil. Guiar em faixas paralelas os enormes pulverizadores atuais sobre uma área infestada por ervas daninhas torna-se praticamente impossível sem algum tipo de orientação. Os marcadores

de espuma utilizados são pouco precisos por serem influenciados pelo vento e pelo sol, tornando essencial um sistema de orientação eletrônico como os baseados em receptores GPS.

Com o intuito de disponibilizar aos produtores brasileiros um sistema de direcionamento automático nacional e de menor custo em relação aos sistemas importados, a AGX Tecnologia Ltda. iniciou em 2003 o desenvolvimento do sistema AGpilot® voltado especificamente às tarefas de dessecação realizadas com o auxílio de pulverizadores autopropelidos. Devido a pouca precisão, esse sistema não é adequado para as tarefas de pulverização pós-plantio sendo esse o ponto de partida para o desenvolvimento do método baseado em visão computacional apresentado neste trabalho.

Para simplificar os testes e a validação do método proposto antes que o mesmo fosse utilizado no sistema AGpilot® foi utilizado um triciclo devidamente modificado para permitir o controle eletrônico de seu deslocamento. Essa plataforma de testes e o sistema AGpilot® são descritos a seguir.

5.1 Triciclo

Um sistema de direcionamento automático baseado no método proposto neste trabalho foi desenvolvido para um triciclo de carga da empresa Dream Bike, modificado para ter tração e controle de direção por motores elétricos, como ilustrado na Figura 5.1. Esse triciclo suporta uma carga máxima de 50 *Kg*, possui tração nas duas rodas traseiras e os motores adaptados para o controle da direção e da tração são controlados eletronicamente, por microcontroladores e controle de potência por largura de pulso (*PWM - Pulse Width Modulation*). Por ser o foco deste trabalho, apenas o controle automático da direção foi implementado. O controle da tração é realizado remotamente por meio de um rádio controle.



Figura 5.1: Plataforma de testes.

O diagrama ilustrado na Figura 5.2 apresenta os módulos que compõem a plataforma de testes. A descrição de cada módulo encontra-se a seguir.

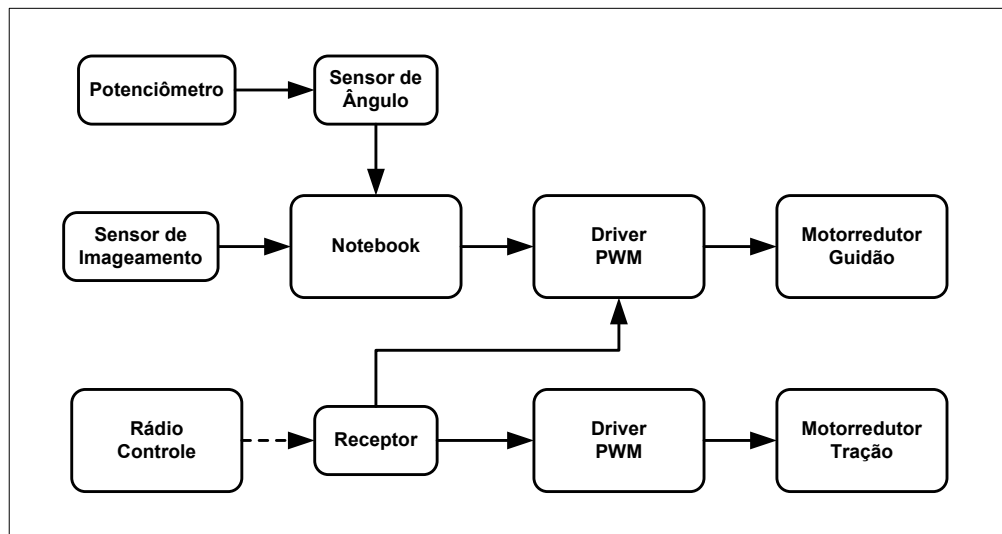


Figura 5.2: Diagrama de blocos da plataforma de testes.

Sensor de Imageamento

O sensor de imageamento utilizado é uma *webcam* colorida modelo Live! Ultra da empresa Creative. Essa câmera possui um sensor CCD, conexão USB (*Universal Serial Bus*) e lentes grande-angular. Sua resolução máxima é de 640x480 *pixels* a 30 quadros por segundo. A câmera utilizada também possui *zoom* digital e controle de ganho automático capaz de reduzir os efeitos de mudanças não muito drásticas de iluminação. A Figura 5.3 ilustra a câmera descrita anteriormente e o seu posicionamento no triciclo.

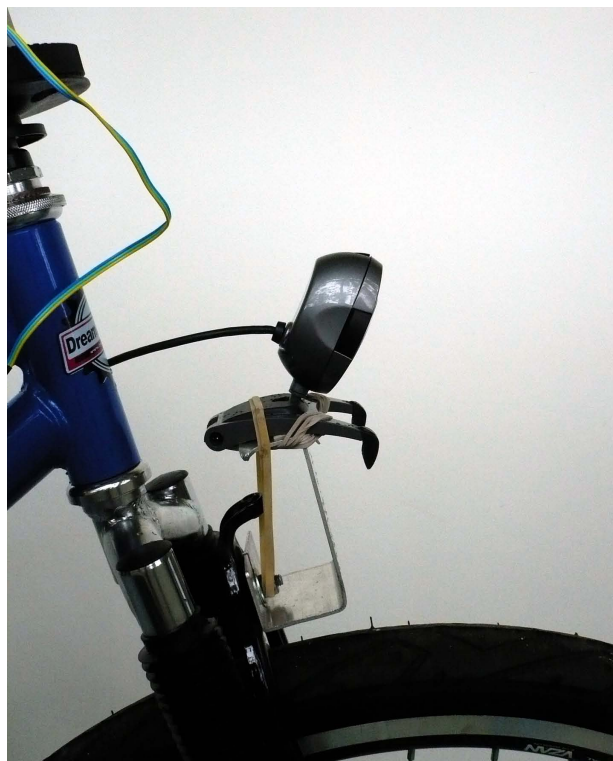


Figura 5.3: Sensor de imagem da plataforma de testes.

A câmera foi fixada a uma distância de 70 *cm* do solo em um suporte de alumínio acoplado ao garfo dianteiro do triciclo. O ângulo de inclinação em relação ao eixo vertical foi ajustado em 30°.

Sensor de Ângulo

O módulo Sensor de Ângulo, ilustrado na Figura 5.4, calcula o ângulo de esterçamento da roda dianteira do triciclo. O ângulo é calculado por um microcontrolador PIC16f876 da Microchip a partir dos dados digitalizados de um potenciômetro posicionado sobre o eixo vertical da roda dianteira. A transmissão do ângulo calculado é realizada através de um *link* de comunicação RS232.

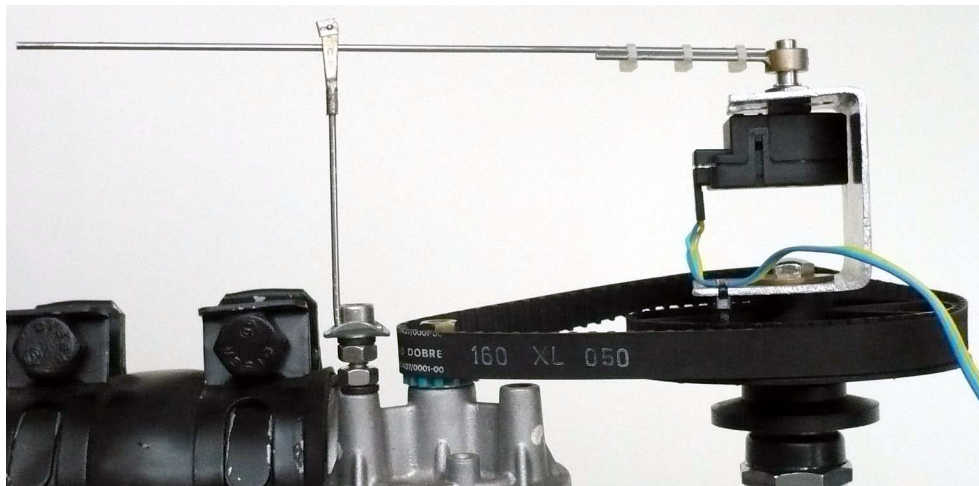


Figura 5.4: Sensor de ângulo da plataforma de testes.

O potenciômetro foi fixado sobre a polia de movimentação do eixo da direção. Nessa configuração o eixo do potenciômetro permanece fixo enquanto o seu corpo se movimenta. Essa instalação foi realizada com o intuito de minimizar os ajustes mecânicos necessários a sua fixação.

O módulo Sensor de Ângulo foi desenvolvido para fornecer ângulos negativos quando a roda dianteira estiver virada para a esquerda e ângulos positivos no caso contrário. O valor de ângulo 0° é fornecido quando a roda dianteira estiver alinhada ao eixo longitudinal do triciclo.

Rádio Controle

Um sistema de rádio controle (transmissor e receptor) Futaba PCM 8 canais, utilizado em aeromodelos, foi adaptado ao triciclo para permitir o acionamento remoto da tração e da direção. Esse sistema foi instalado com o intuito principal de elevar o nível de segurança operacional do triciclo, permitindo a desativação do sistema elétrico em caso de falha do sistema de direcionamento automático.

Notebook

Para simplificar e agilizar o desenvolvimento do sistema de direcionamento da plataforma de testes foi utilizado um *notebook* Toshiba modelo A70. Esse computador possui um processador Intel Core 2 Duo de 1.66Ghz e o sistema operacional Windows XP Professional. Essa plataforma de hardware/software foi utilizada para o processamento das imagens, para a navegação, para o controle da roda dianteira e para a geração do sinal PWM enviado ao *driver* de potência conectado ao motorreductor da direção.

Devido à ausência de portas seriais foi utilizado um conversor USB/Serial para conectar o módulo Sensor de Ângulo ao *notebook*. O sensor de imagem foi conectado em uma segunda porta USB e a porta paralela foi utilizada para a geração do sinal PWM.

Motorreductores

Dois motorreductores responsáveis pela movimentação do eixo da direção e das rodas traseiras foram instalados no triciclo. O primeiro motorreductor foi posicionado sobre o tubo horizontal e conectado ao eixo da direção do triciclo por meio de uma correia e de polias dentadas. A Figura 5.5 ilustra o motorreductor citado anteriormente.

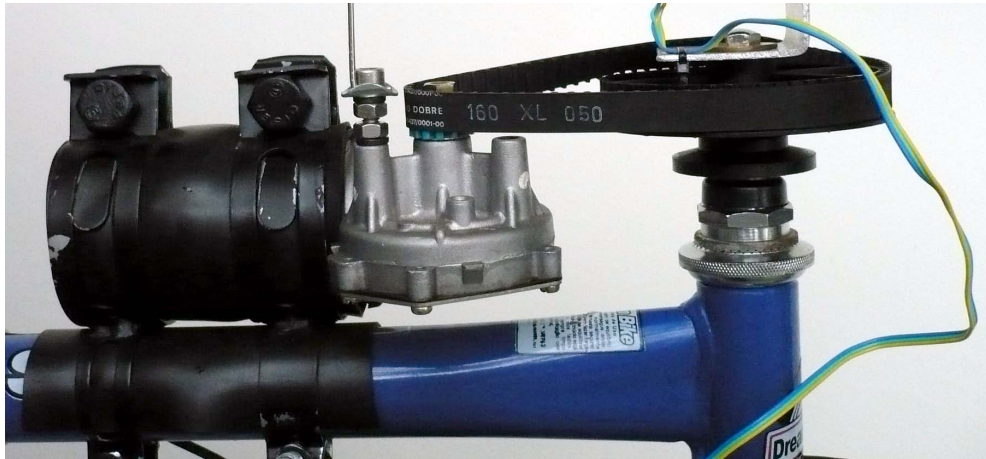


Figura 5.5: Motorreductor - direção.

O segundo motorreductor, ilustrado na Figura 5.6, foi posicionado atrás do cubo central e conectado às rodas traseiras por meio de uma corrente.

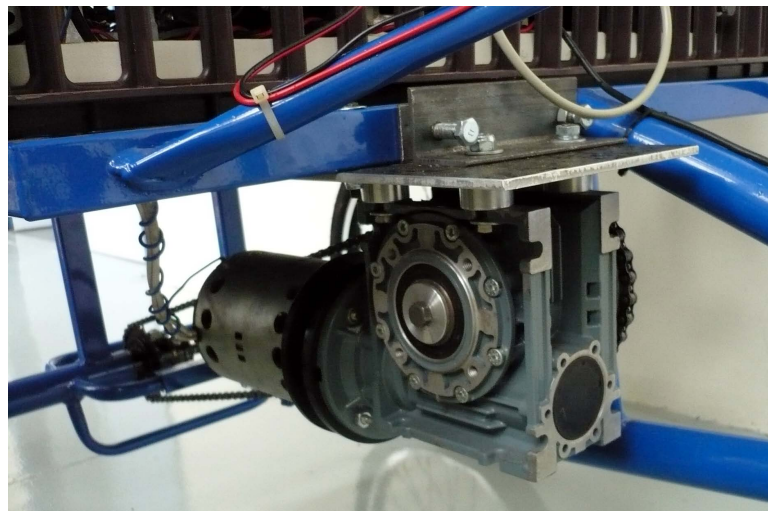


Figura 5.6: Motorreductor - tração.

Os motorredutores instalados são alimentados por duas baterias de 70Ah - 12v e controlados eletronicamente através de sinais PWM por dois *drivers* proporcionais de potência. Um circuito eletrônico acionado por rádio controle é responsável pela ativação do circuito elétrico.

5.1.1 Software

O software do sistema de direcionamento desenvolvido para a plataforma de testes, executado no *notebook* descrito anteriormente, foi implementado no ambiente de desenvolvimento Borland Delphi 6. Utilizou-se o paradigma de programação multitarefa para possibilitar a realização das diversas operações necessárias. O Apêndice B apresenta as principais rotinas implementadas.

Para a captura de vídeo são utilizados os componentes VCL (*Visual Component Library*) VideoLab 3.1 desenvolvido pela empresa MITOV (MITOV, 2007). O acesso à porta serial é realizado através do componente TComPort 3.1. O acionamento da porta paralela para a geração do sinal PWM é realizado através da DLL (*Dynamic Link Library*) inpout32.dll. O diagrama ilustrado na Figura 5.7 apresenta os processos e as principais rotinas existentes.

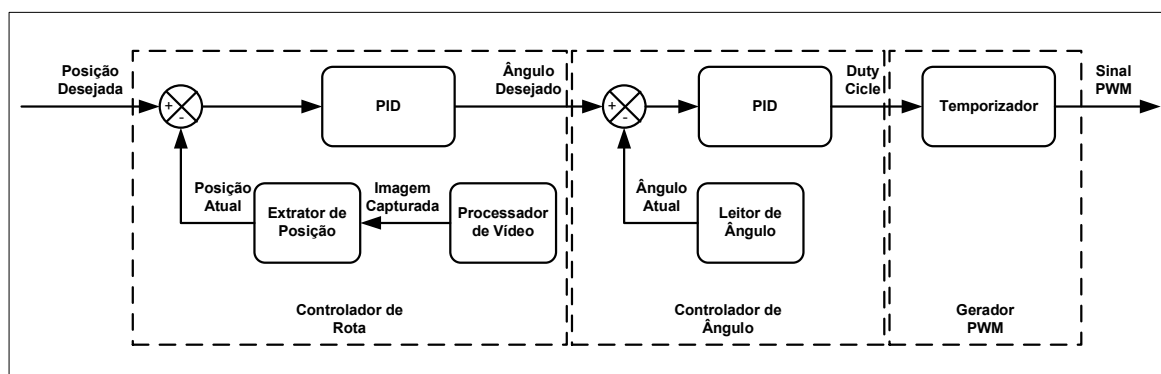


Figura 5.7: Diagrama de blocos do sistema desenvolvido.

Os processos Controlador de Rota, Controlador de Ângulo e Gerador PWM são executados concorrentemente no software desenvolvido. Esses processos são descritos a seguir.

Controlador de Rota

O processo Controlador de Rota obtém o quadro de vídeo do componente de captura, calcula a posição atual a partir do ponto de fuga extraído e calcula, através de um controlador PID, o ângulo de esterçamento da roda necessário para corrigir ou manter a posição na rota atual. O ângulo calculado é enviado ao processo Controlador de Ângulo para que o mesmo atue no sistema de direção do triciclo. Esse controlador é responsável por manter o ponto de fuga no centro da imagem, ou seja, mantê-lo em zero (posição desejada) a distância em *pixels* do ponto de fuga em relação à coluna central da imagem. O Controlador de Rota foi ajustado para ser executado após a captura dos quadros de vídeo sendo, portanto, executado a cada 33.3 ms (30 Hz). No entanto, observou-se que a taxa de quadros fornecida pela câmera não é constante, sendo dependente das condições de iluminação do ambiente.

Controlador de Ângulo

O processo Controlador de Ângulo é responsável por manter ou posicionar as rodas dianteiras do triciclo em um ângulo pré-determinado. Um controlador proporcional calcula um valor percentual de potência entre -100% e 100% a ser aplicado no motorreductor. Esse valor é obtido através de um controlador PID a partir do ângulo desejado e do ângulo atual obtido do módulo Sensor de Ângulo. Valores negativos indicam o sentido anti-horário de rotação do motor enquanto que os valores positivos indicam o sentido horário. Esse processo foi configurado para ser executado a cada 10 ms (100 Hz).

O valor percentual de potência calculado é convertido em um pulso de largura proporcional e é enviado ao processo Gerador PWM. Essa conversão é realizada através de uma interpolação linear com os valores de largura de pulso suportados pelo circuito de potência PWM conectado ao motorreductor. Esses valores são de $1,50\text{ ms}$ para nenhuma

movimentação, acima de 1,50 *ms* a 1,8 *ms* para movimentos no sentido horário e valores abaixo de 1,5 *ms* a 1,12 *ms* para movimentos anti-horários.

Gerador PWM

O processo Gerador PWM é responsável por gerar um sinal PWM com período de 20 *ms* e largura de pulso configurável entre 1 *ms* e 2 *ms* no pino 9 da porta paralela. Esse processo foi configurado como um processo de tempo real para garantir a correta temporização do sinal.

5.2 AGpilot[®]

O sistema AGpilot[®] foi desenvolvido, previamente ao presente trabalho, para um pulverizador Jacto modelo Uniport 2500 da Fazenda Campo Bom, localizada em Chapadão do Sul - MS. A Fazenda Campo Bom possui aproximadamente 40.000 *ha* cultivados principalmente com soja e milho utilizando o método de plantio direto. O pulverizador Uniport 2500 possui uma barra de pulverização de 24 metros de comprimento, tanque de pulverização com capacidade para 2500 litros, peso total de 9300 *kg* e possui um sistema de direção hidrostático.

O AGpilot[®] foi desenvolvido de forma a ser o mais simples e intuitivo possível. Para que o sistema possa ser utilizado é necessário posicionar a máquina sobre o início e o sobre fim da primeira linha de trabalho para georreferenciá-la e fornecer a largura da barra de pulverização. Esse sistema não dispensa completamente o operador da máquina, pois o mesmo não realiza as manobras ao final de cada linha necessárias para posicionar o pulverizador sobre a próxima linha de trabalho. Fica a cargo do operador detectar o final da linha, desligar o controle automático, manobrar a máquina e acioná-lo novamente. Nesse contexto, o sistema pode ser considerado como um sistema de auxílio à pilotagem

por somente manter constante a rota do pulverizador.

O diagrama ilustrado na Figura 5.8 apresenta os módulos que compõem o sistema. A descrição de cada módulo encontra-se a seguir.

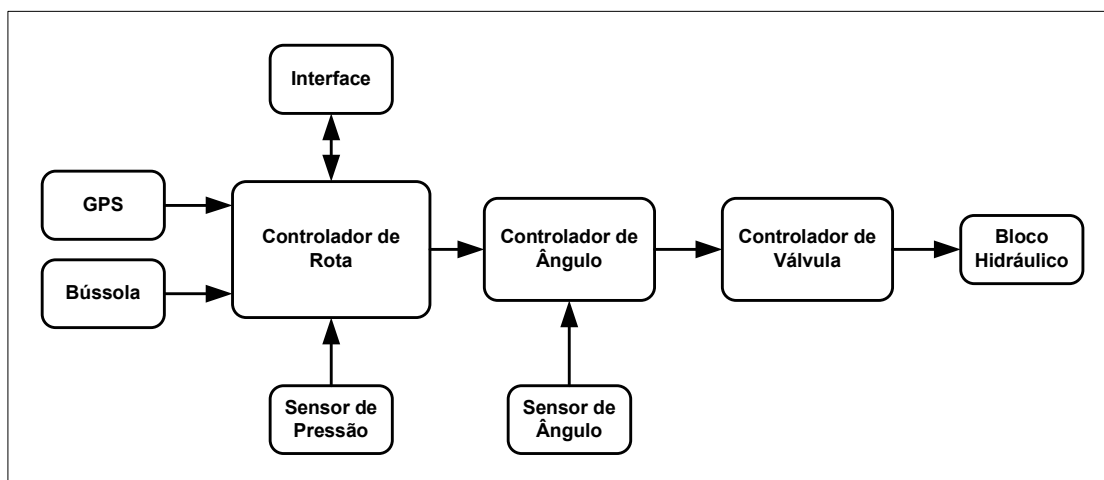


Figura 5.8: Diagrama de blocos do sistema AGpilot®.

Interface

A interação com o sistema é realizada através de uma interface gráfica desenvolvida para um computador portátil (*Pocket PC*) modelo iPAQ H3950 da HP. Essa plataforma foi escolhida devido a grande disponibilidade no mercado e devido à simplicidade e à facilidade no desenvolvimento de software.

A interface, ilustrada na Figura 5.9, recebe os dados e/ou comandos do usuário e permite que o mesmo visualize o campo e a posição da máquina agrícola sobre ele. Esse módulo também apresenta a qualidade dos dados fornecidos pelo receptor GPS, a velocidade e a distância da máquina em relação à linha de trabalho corrente. Uma barra formada por triângulos permitir ao operador guiar manualmente a máquina sobre a linha de trabalho atual ou posicioná-la sobre a próxima linha. O envio e o recebimento de dados é realizado através de uma porta de comunicação serial RS232 a 57600 *bps*.

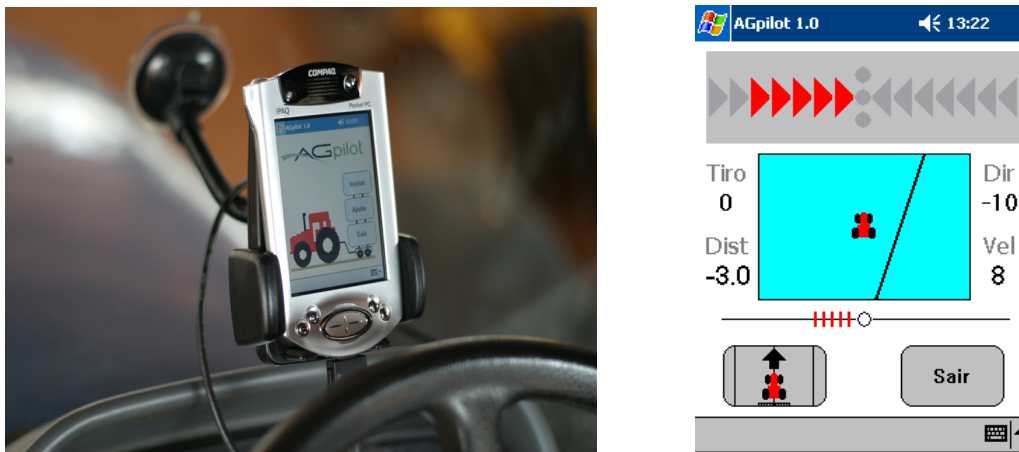


Figura 5.9: Interface do sistema AGpilot®.

Receptor GPS

Os dados de posição da máquina sobre o campo são fornecidos por um receptor GPS responsável por calcular e enviar, ao módulo Controlador de Rota, a distância em relação a linha de trabalho, a posição geográfica, a velocidade, a altitude e outros dados relacionados aos satélites e ao sinal recebido. O modelo utilizado é o OEM4 L1 da Novatel capaz de fornecer dados a 20 Hz . A sua precisão sem correção é de $1,8\text{ m}$ CEP. Para a recepção do sinal GPS foi utilizada uma antena Novatel GPS-701 fixada no topo da cabine. A antena foi devidamente posicionada à frente das rodas dianteiras para facilitar o ajuste dos controladores (WILL *et al.*, 1998). A Figura 5.10 ilustra a antena utilizada.



Figura 5.10: Antena GPS.

Bússola Eletrônica

Devido a grande variação do *heading* fornecido pelo receptor GPS, foi utilizada uma bússola magnética eletrônica para fornecer a direção de deslocamento do pulverizador. O modelo utilizado é o TNT Revolution da True North Technologies também fixada sobre a cabine do pulverizador (Figura 5.11). Além de fornecer a informação de *heading*, esse dispositivo fornece dados de *pitch* e de *roll*. Esse dispositivo possui uma porta de comunicação serial RS232 e a sua precisão é de 0,5°.

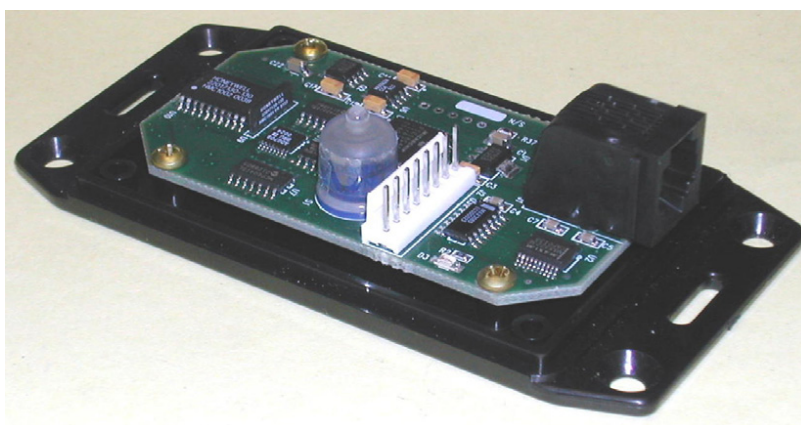


Figura 5.11: Bússola.

Sensor de Pressão

Instalado na caixa de direção, esse dispositivo detecta a elevação de pressão hidráulica causada pela movimentação do volante realizada pelo operador. Ilustrado na Figura 5.12, esse sensor atua como um dispositivo de segurança, pois o sistema é desligado automaticamente quando esse dispositivo é acionado.

Sensor de Ângulo

O módulo Sensor de Ângulo, ilustrado na Figura 5.13, calcula o ângulo de esterçamento das rodas dianteiras do pulverizador. O ângulo é calculado por um microcontrolador PIC16f876 da Microchip a partir dos dados digitalizados de um potenciômetro



Figura 5.12: Sensor de pressão.

posicionado sobre o eixo vertical de uma das rodas. A transmissão do ângulo calculado é realizada através de um *link* de comunicação RS485.



Figura 5.13: Sensor de ângulo.

Bloco Hidráulico

Esse dispositivo é responsável por atuar nos pistões hidráulicos conectados às rodas dianteiras do pulverizador. Ilustrado na Figura 5.14, esse dispositivo é acoplado ao sistema de direção hidráulico e atua em resposta aos comandos enviados pelo módulo Controlador de Válvula. O Bloco Hidráulico permite controlar eletronicamente a quantidade de óleo enviada aos pistões influenciando na velocidade de movimentação da rodas.

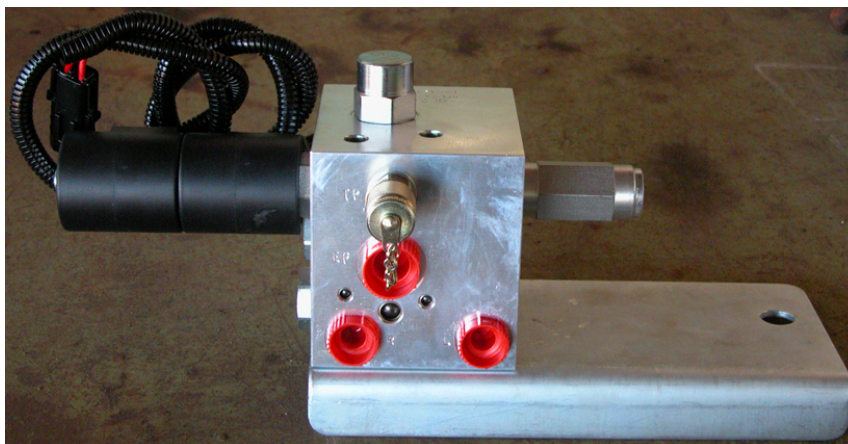


Figura 5.14: Bloco hidráulico.

Controlador de Válvula

Esse módulo aciona a válvula proporcional presente no bloco hidráulico citado anteriormente. Essa válvula é responsável pelo controle do fluxo de óleo enviado aos pistões conectados às rodas dianteiras do pulverizador. O acionamento é realizado por um microcontrolador PIC16f876 através da variação da corrente aplicada às bobinas que envolvem a válvula. Para o correto funcionamento da válvula proporcional é necessário fornecer um sinal específico de corrente (*dither*) para aumentar o tempo de resposta e reduzir a histerese. Um controlador proporcional mantém constante a corrente aplicada por meio de um sinal PWM.

Controlador de Rota

Esse módulo, a partir dos dados do GPS, dos dados da bússola eletrônica e da rota previamente selecionada, calcula o ângulo de esterçamento das rodas a ser mantido pelo módulo Controlador de Ângulo. Esse ângulo, calculado por um controlador do tipo PID (OGATA, 1993; KUO, 1985), corresponde à correção necessária a ser aplicada para manter ou posicionar a máquina agrícola sobre a rota desejada.

Para melhorar a estabilidade do sistema e para simplificar a etapa de calibração

é utilizado o método de busca antecipada, descrito no Capítulo 2, no cálculo do ângulo de esterçamento citado anteriormente. Esse método permite prever a posição da máquina em um tempo futuro pré-determinado. A posição futura da máquina é calculada a partir do *heading*, do tempo futuro desejado e da distância real da máquina em relação à linha de trabalho corrente.

Controlador de Ângulo

O módulo Controlador de Ângulo é responsável por manter ou posicionar as rodas dianteiras do pulverizador em um ângulo pré-determinado. Um controlador proporcional calcula o fluxo de óleo (0..100%) necessário para manter ou posicionar as rodas no ângulo desejado. Após o cálculo, o valor do fluxo é enviado ao módulo Controlador de Válvula para que o mesmo acione a válvula proporcional responsável pelo controle da vazão.

Os módulos Controlador de Rota, Controlador de Ângulo e as rotinas de recepção e de extração de dados da bússola e do receptor GPS são executadas em uma unidade central composta por um *core* RCM3110 da Rabbit Semiconductor. Esse *core* possui 6 portas seriais RS232, *clock* de 29 *Mhz* e suporta multitarefa preemptiva. Devido à pouca precisão das rotinas de ponto flutuante do *core* RCM3110, optou-se pela utilização das rotinas de navegação disponíveis no receptor GPS. Essas rotinas simplificaram o desenvolvimento e reduziram o tempo de implementação do sistema.

5.3 Considerações Finais

A plataforma de testes apresentada na Seção 5.1 é uma ferramenta de pesquisa importante para o desenvolvimento e testes de sistemas que operam em ambientes externos. A sua praticidade possibilita diminuir o tempo necessário para a validação de módulos

de hardware e de software a um custo reduzido. Essa plataforma deve ser utilizada futuramente como a plataforma de testes de um sistema de direcionamento automático para um pequeno barco a motor.

Os resultados dos testes do sistema AGpilot[®] realizados em campo mostraram que o sistema pode ser utilizado de forma satisfatória nas tarefas de dessecação para evitar a sobreposição e as falhas de pulverização. Descritos no Apêndice C, os testes também mostraram que os módulos responsáveis pela movimentação das rodas atuaram de forma precisa tornando-os aptos a serem utilizados como parte de um sistema baseado em visão computacional.

O próximo capítulo apresenta os testes realizados e os resultados obtidos do sistema de direcionamento automático desenvolvido a partir do método proposto no Capítulo 4 para a plataforma de testes descrita na Seção 5.1.

Capítulo 6

Resultados Experimentais

A realização de experimentos em um ambiente não controlado é uma tarefa complexa e exige um grande esforço humano e material. A disponibilidade de equipamentos, de áreas adequadas para a realização de testes e, principalmente, de condições climáticas favoráveis são alguns dos requisitos que necessitam ser preenchidos para a obtenção de resultados satisfatórios.

Realizar testes em equipamentos e máquinas agrícolas apresenta outros desafios adicionais que dificultam consideravelmente a realização de experimentos: a sazonalidade dos cultivos e a distância das áreas cultivadas.

Devido aos fatores descritos anteriormente foi utilizada a plataforma de testes apresentada na Seção 5.1 para a validação e ajustes do método de direcionamento automático por visão computacional desenvolvido neste trabalho. A utilização dessa plataforma simplificou e agilizou a obtenção dos resultados experimentais.

A seguir, são apresentados a metodologia dos testes realizados e os resultados obtidos.

6.1 Materiais e Métodos

Os testes experimentais do sistema de direcionamento automático baseado no método proposto neste trabalho foram realizados no campo de futebol do Campus I da USP São Carlos. Para simular uma área cultivada em linhas, foi utilizada uma fita de demarcação zebraada nas cores amarelo/preto de 7,5 cm de largura. A fita utilizada foi disposta diagonalmente sobre o campo para representar o sulco de separação das linhas de cultivo. Simulou-se nesta configuração apenas duas linhas de cultivo separadas pela fita de demarcação (Figura 6.1).



Figura 6.1: Posicionamento do triciclo.

O objetivo dos testes realizados foi o de avaliar o comportamento do sistema no direcionamento do triciclo sobre a fita de demarcação. Os testes foram realizados em um dia nublado, com garoa e com vento de aproximadamente 10 *km/h*. Um receptor GPS portátil posicionado no triciclo foi utilizado para a obtenção da velocidade média atingida durante os testes.

Duas pessoas com tarefas distintas conduziram os testes. A primeira foi responsável pelo controle da tração e pelo acionamento do sistema elétrico do triciclo através do rádio controle e a segunda pessoa se responsabilizou pelo acionamento do sistema de direcionamento automático através do *notebook*.

6.1.1 Calibração

Os diversos módulos que compõem o sistema foram calibrados previamente em laboratório para a realização dos testes em campo. Uma ferramenta de calibração foi desenvolvida para facilitar a calibração dos módulos Sensor de Ângulo, Controlador de Ângulo e Controlador de Rota descritos na Seção 5.1.

A calibração do módulo Sensor de Ângulo corresponde ao armazenamento em memória não volátil de dados digitalizados do potenciômetro. Os valores armazenados são os valores de máximo à esquerda, máximo à direita e de centro do potenciômetro. Esses valores, juntamente com o máximo ângulo atingido pelas rodas do triciclo, são utilizados no cálculo do ângulo atual. Os comandos de armazenamento e o ângulo máximo são enviados ao módulo Sensor de Ângulo através da interface ilustrada na Figura 6.2.

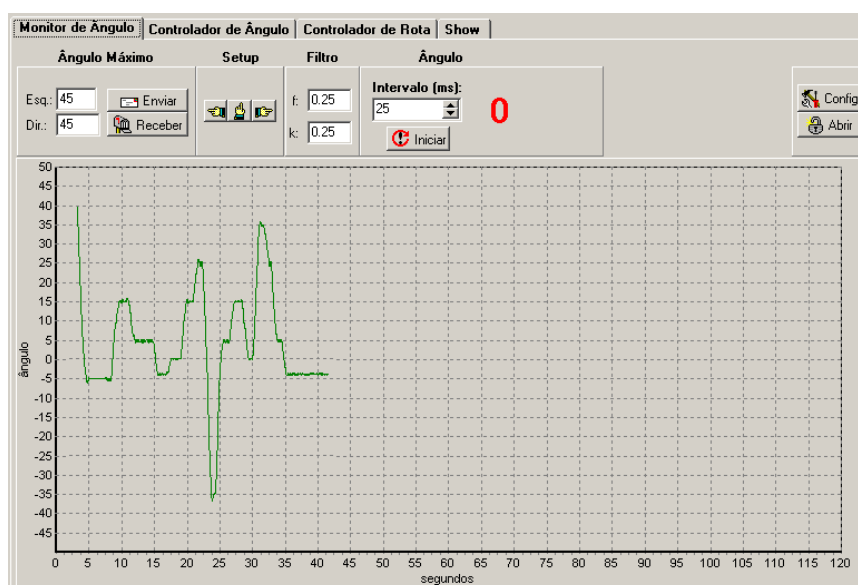


Figura 6.2: Interface de configuração do módulo Sensor de Ângulo.

As constantes configuráveis do módulo Controlador de Ângulo são os ganhos do controlador PID implementado. Um valor de ajuste (*offset*) pode ser utilizado para refinar o posicionamento realizado. Nesse controlador apenas o ganho proporcional foi ajustado. Observou-se nos testes realizados que apenas um controlador do tipo proporcional é sufi-

ciente para o correto controle das rodas do triciclo.

O gráfico contido na interface de configuração ilustrada na Figura 6.3 apresenta o comportamento do controlador de ângulo após a inserção consecutiva dos *set points* -5° , 5° , -10° , 10° , -20° , 20° e 0° . A velocidade máxima de esterçamento das rodas foi de 40 graus/s com um ganho proporcional 3.

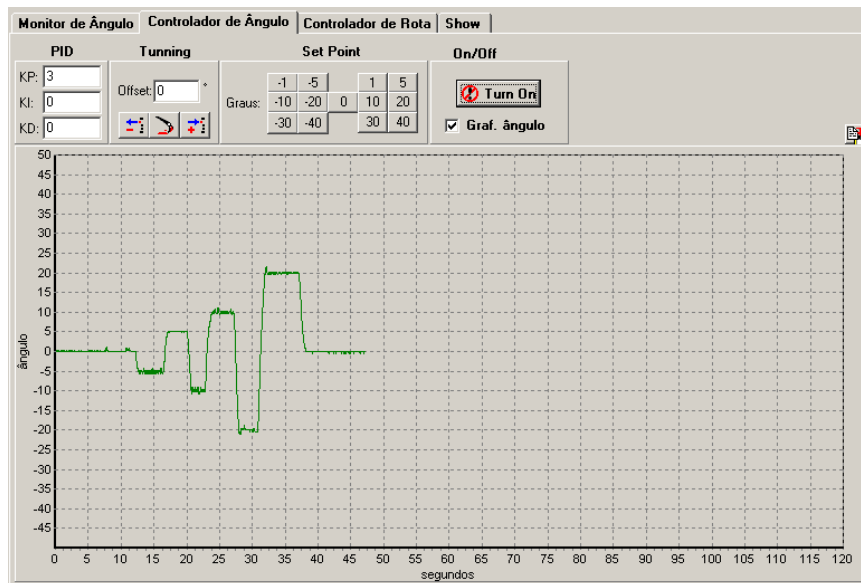


Figura 6.3: Interface de configuração do módulo Controlador de Ângulo.

A Figura 6.4 ilustra a interface principal do sistema desenvolvido. Nesta interface é apresentada a imagem capturada da *webcam*, o gráfico de soma do quadro atual e o histórico das distâncias calculadas a partir dos pontos de fuga extraídos das imagens. A configuração dos ganhos do controlador PID implementado, a configuração dos parâmetros do sensor de imageamento e o acionamento do sistema de direcionamento automático também são realizados através dessa interface.

6.2 Resultados

Os primeiros testes do sistema de direcionamento foram realizados com o intuito de avaliar o comportamento dos dispositivos instalados no triciclo. Foram avaliados os

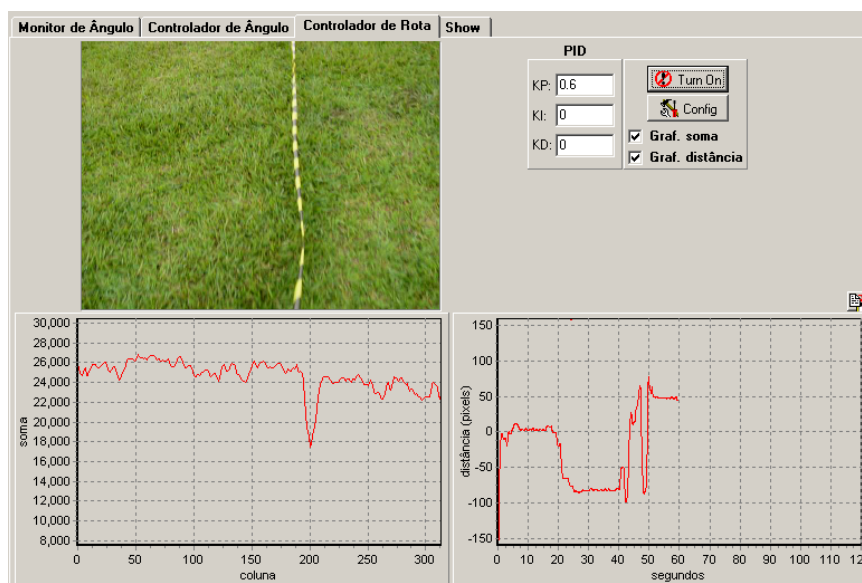


Figura 6.4: Interface de configuração do módulo Controlador de Rota.

dispositivos eletrônicos e mecânicos bem como o dispositivo de segurança acionado através do rádio controle. Estes testes foram realizados na lateral do campo devido à existência de plantas verdes em ambos os lados como mostra a Figura 6.5. O objetivo deste teste foi o de conduzir automaticamente o triciclo no centro da lateral do campo. Os resultados obtidos apesar de não serem conclusivos mostraram que o sistema estava operando corretamente, pois o triciclo se manteve no centro do caminho guiado pelas faixas de grama laterais.

Os testes subseqüentes foram realizados no campo sobre a fita de demarcação disposta diagonalmente. Esses testes apresentaram bons resultados. O sistema foi capaz de guiar o triciclo sobre a fita com um desvio lateral máximo de 10 *cm*. A velocidade média atingida pelo triciclo foi de 5 *km/h*. Nesse teste apenas o ganho proporcional foi ajustado sendo que o valor utilizado foi de 0,6.

Os resultados dos testes utilizando a fita de demarcação estão contidos nas Figuras 6.6, 6.7, 6.8, 6.9 e 6.10. Os gráficos apresentados em cada figura contêm a distância em *pixels* do ponto de fuga em relação ao centro da imagem capturada e os respectivos ângulos de esterçamento.

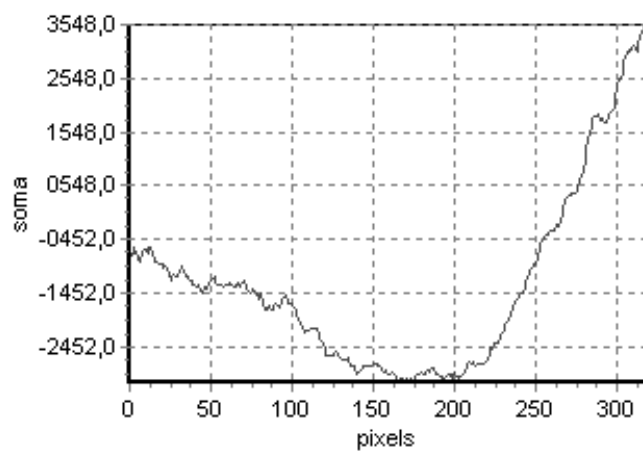


Figura 6.5: Exemplo de quadro processado - teste inicial.

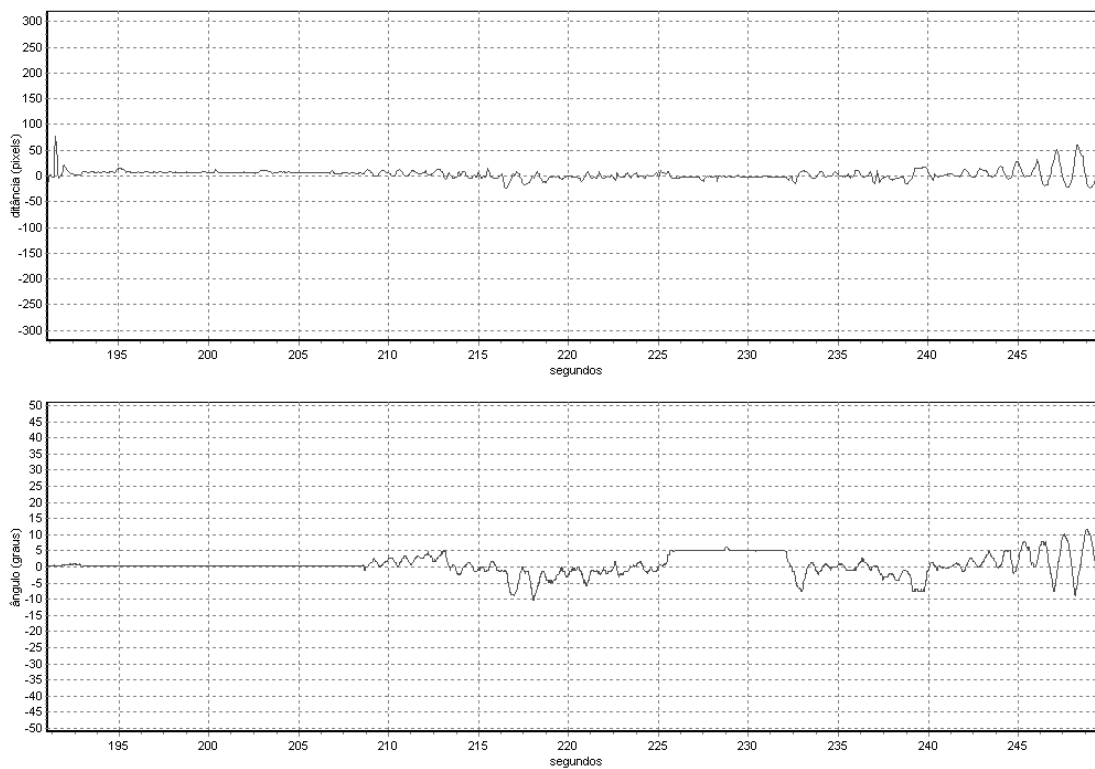


Figura 6.6: Teste 01 - área uniforme e fita imóvel.

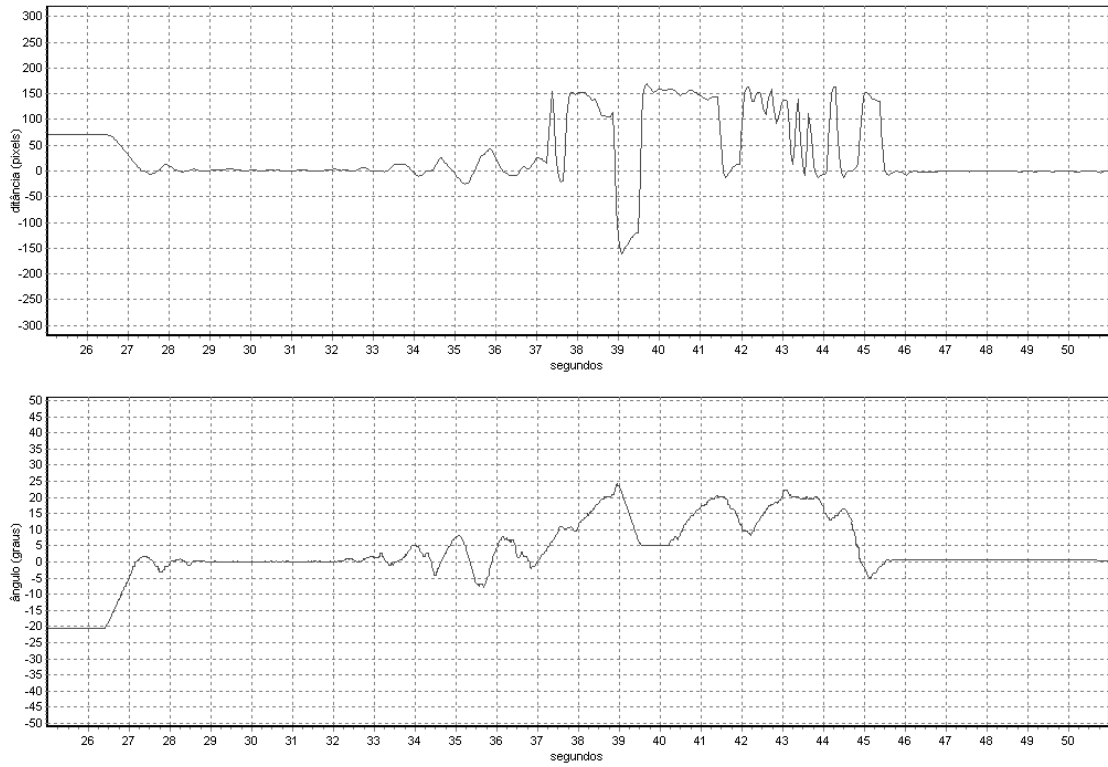


Figura 6.7: Teste 02 - região esbranquiçada.

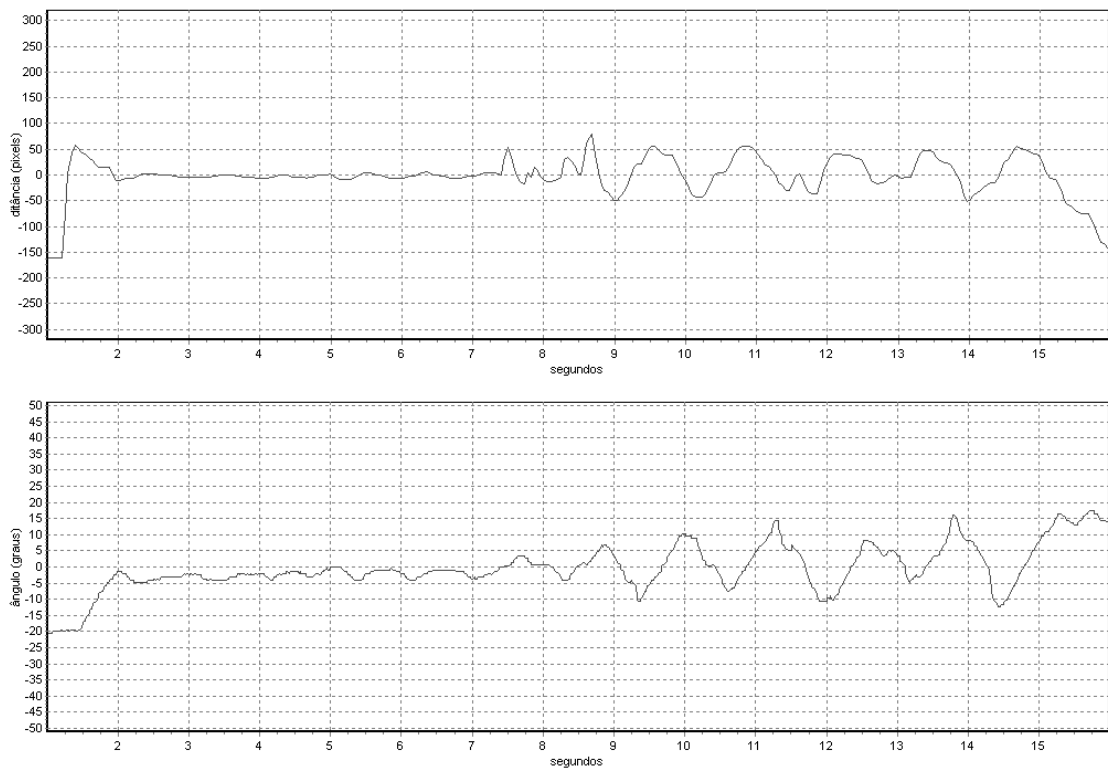


Figura 6.8: Teste 03 - fita esticada e virada.

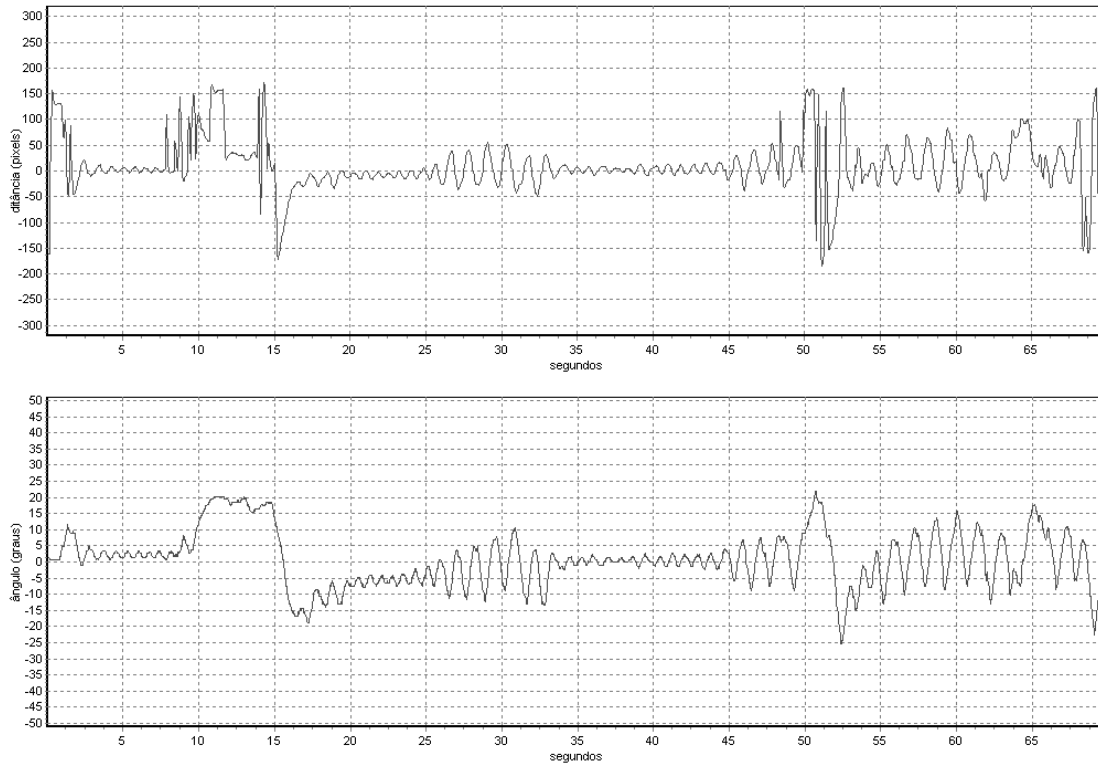


Figura 6.9: Teste 04 - região esbranquiçada, buracos, palha e fita em movimento.

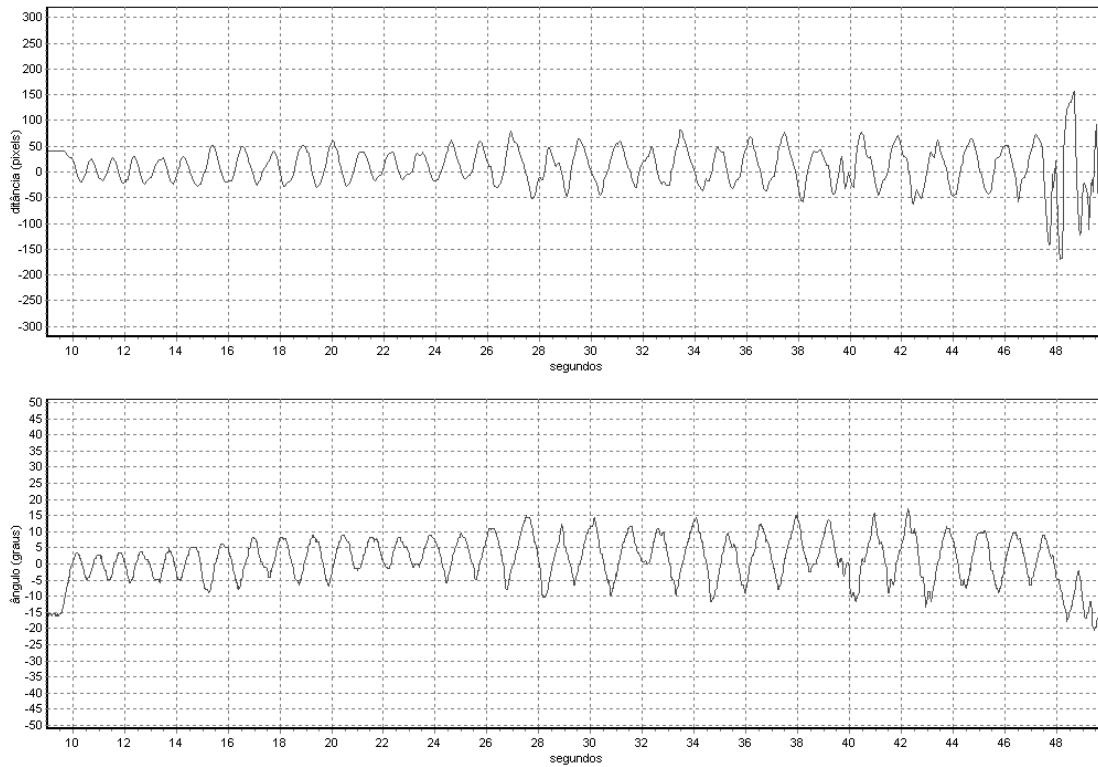


Figura 6.10: Teste 05 - fita em movimento e de dorso.

As oscilações e as mudanças bruscas das distâncias em alguns dos gráficos apresentados são decorrentes de problemas detectados durante a realização dos testes e após a análise do vídeo coletado em campo. Esses problemas ocorreram devido à movimentação da fita causada pelo vento e pela oscilação da roda dianteira do triciclo causada por irregularidades no terreno. Também foi observado que a posição da fita influenciou significativamente os resultados obtidos. O dorso da fita não forneceu um contraste significativo em relação ao verde do campo o que dificultou a obtenção do ponto de fuga. Essa posição foi constantemente alterada pelo vento durante a realização dos testes.

Os problemas detectados após a análise do vídeo mostraram que os buracos e a palha presentes no campo também influenciaram os testes realizados. A presença de manchas esbranquiçadas sobre a grama, devido a aplicação de formicida, também afetou os resultados obtidos. As imagens contidas nas Figuras 6.12, 6.13, 6.14 e 6.15 apresentam alguns dos quadros do vídeo, capturado durante os testes, onde foram detectados os problemas mencionados. A Figura 6.11 apresenta uma seqüência de quadros onde não foram observados problemas.

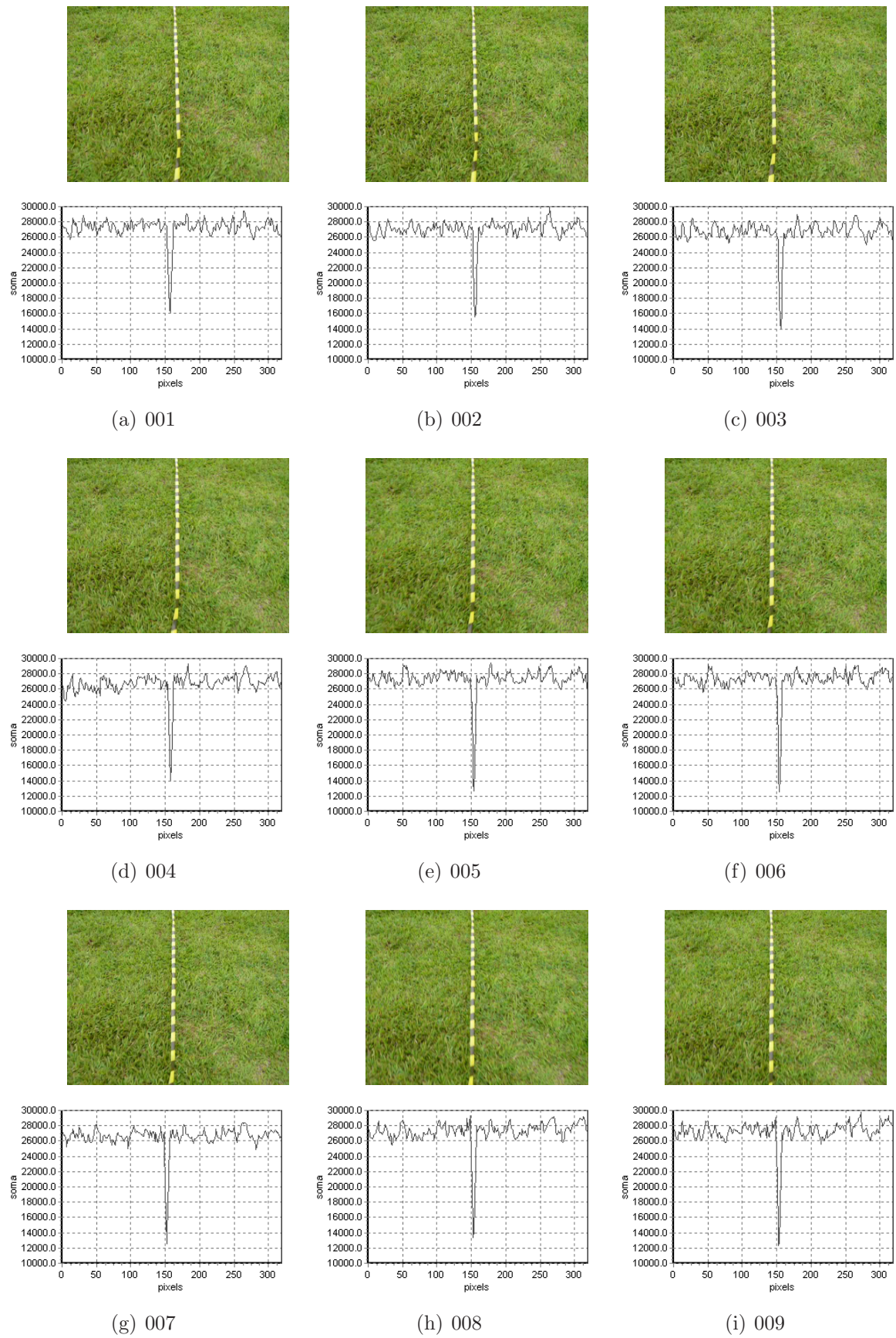


Figura 6.11: Exemplos de quadros normais.

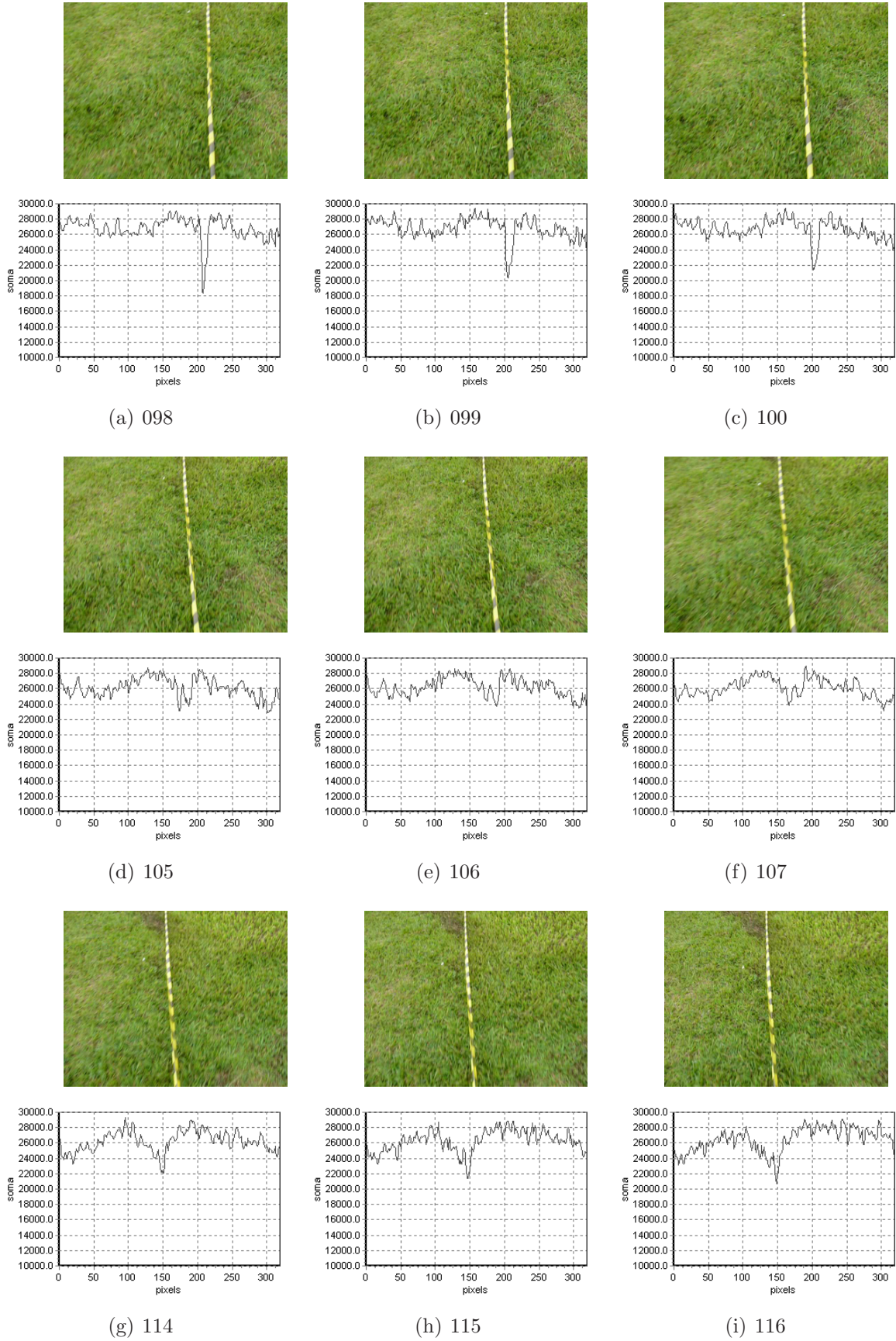


Figura 6.12: Exemplos de quadros com palha e mancha escura.

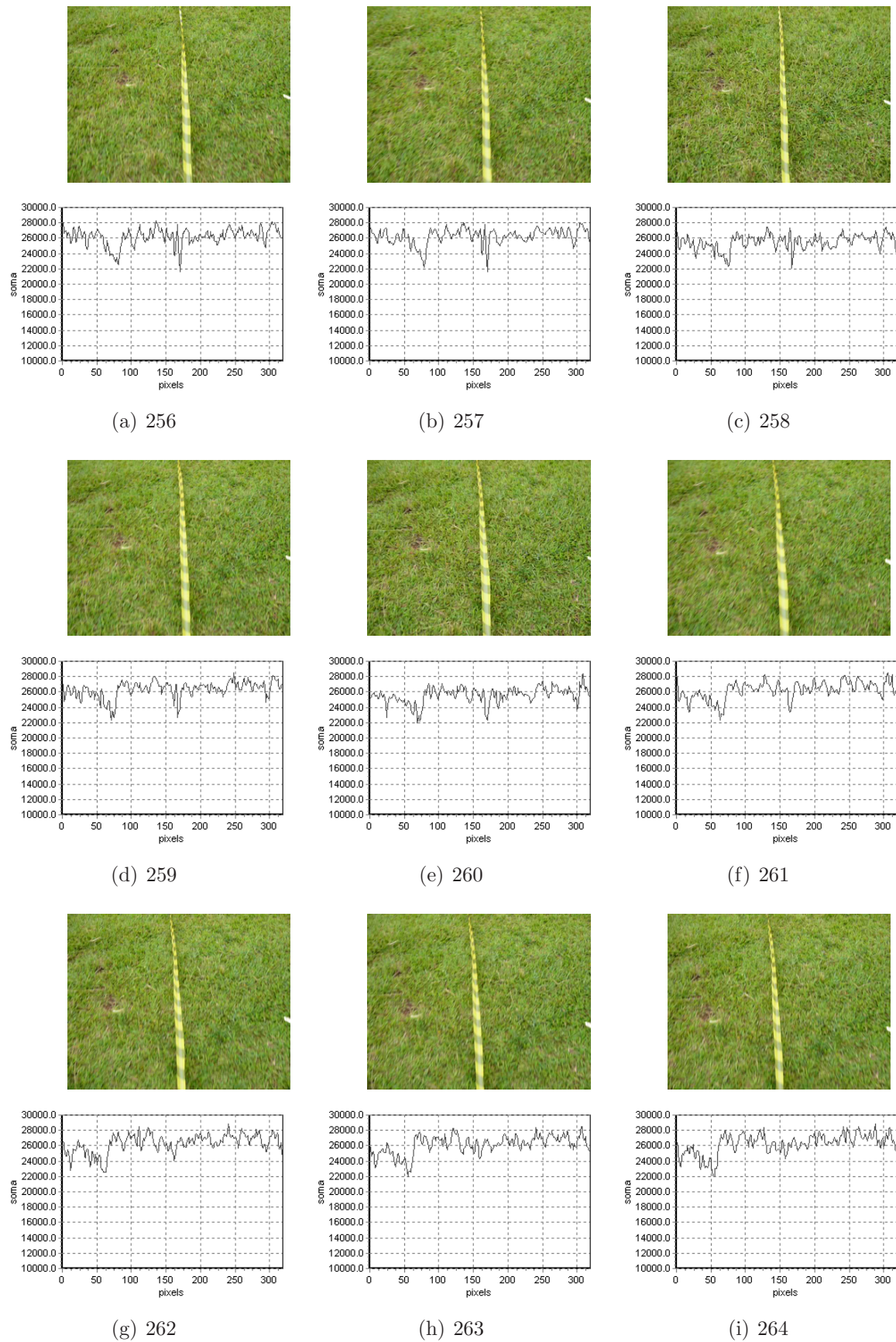


Figura 6.13: Exemplos de quadros com buraco e manchas brancas.

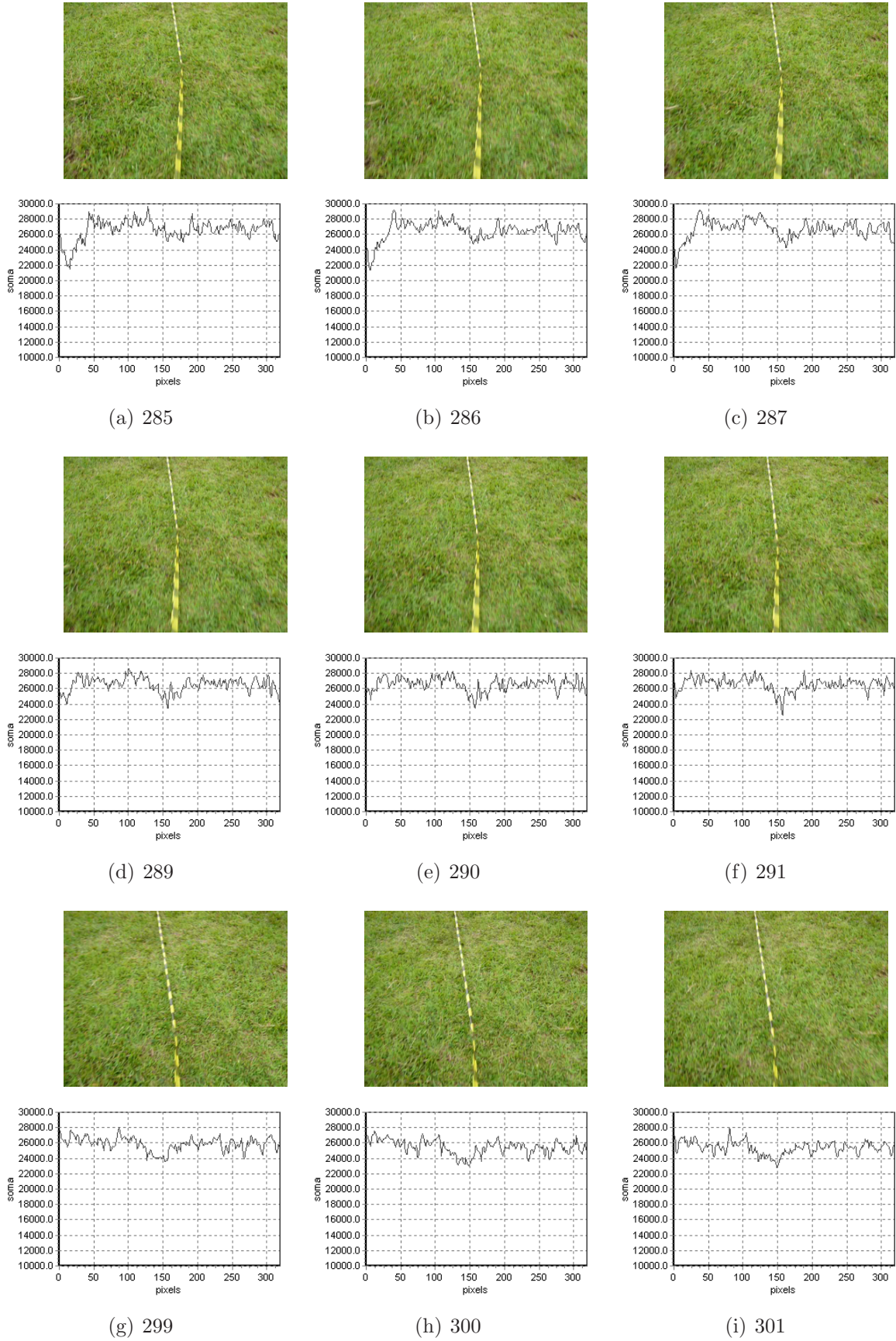


Figura 6.14: Exemplos de quadros com buraco.

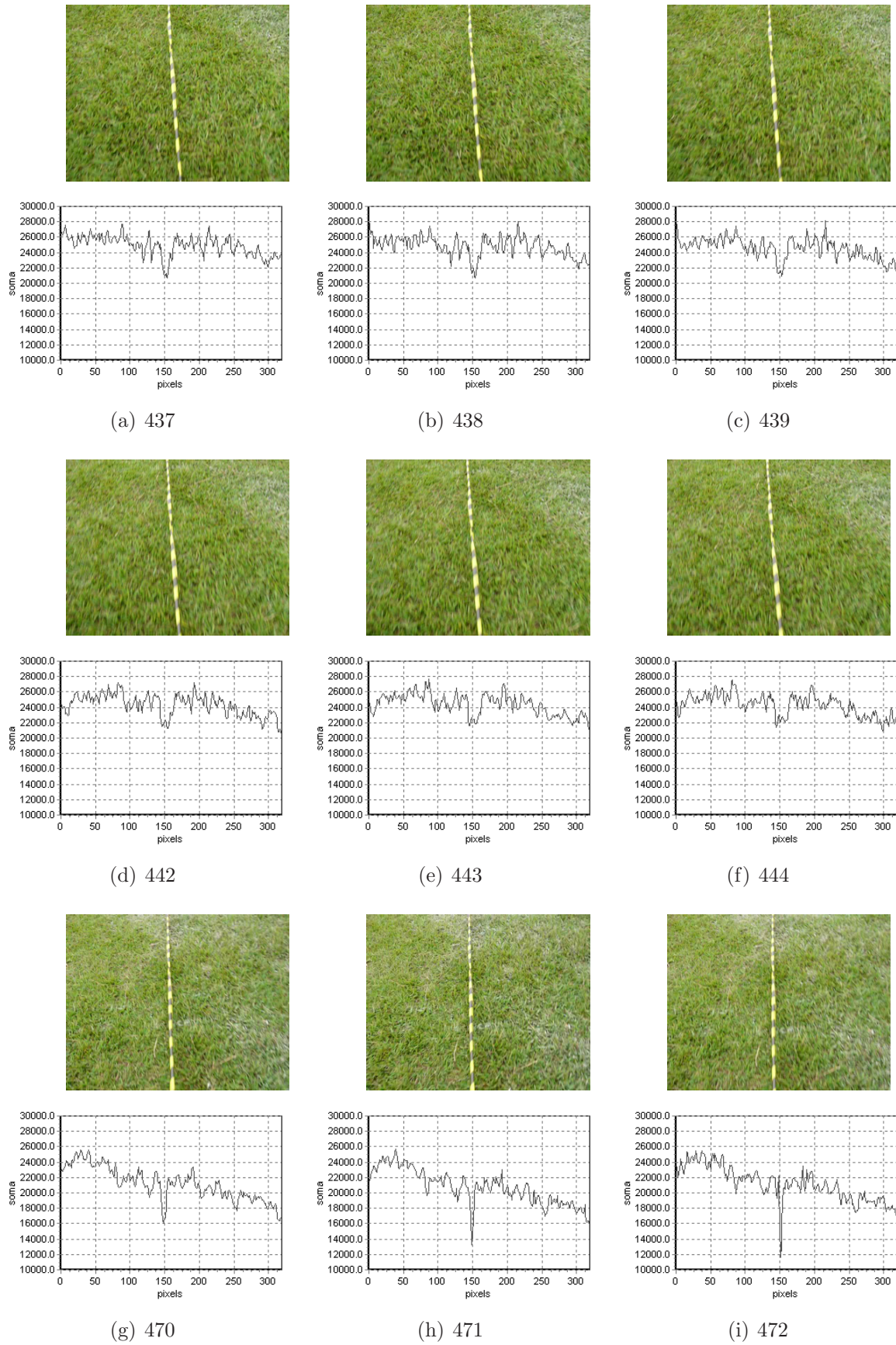


Figura 6.15: Exemplos de quadros com mancha branca.

6.3 Considerações Finais

Os testes experimentais do sistema de direcionamento automático, baseado no método proposto neste trabalho, mostraram que o sistema é capaz de conduzir satisfatoriamente a plataforma de testes sobre um campo demarcado. Os problemas ocorridos durante os testes poderiam ter sido evitados e resultados melhores poderiam ter sido obtidos com a utilização de uma fita de demarcação capaz de fornecer um bom contraste em relação ao campo de futebol.

O próximo capítulo destina-se a apresentar as conclusões e contribuições deste trabalho. Os trabalhos futuros relacionados à adaptação do sistema desenvolvido ao sistema AGpilot[®] também são apresentados no próximo capítulo.

Capítulo 7

Conclusões

O grande crescimento populacional observado nos últimos anos e o aumento da competitividade causado pela globalização tem impulsionado pesquisas nos diversos ramos da agroindústria. A busca recente por energias renováveis e o aumento da utilização das bioenergias tem impulsionando o desenvolvimento tecnológico desse setor. O interesse na utilização de novas tecnologias é normalmente alavancado pela redução de custos e pelo aumento da produção.

A otimização dos processos produtivos do setor agrícola possui um grande potencial de expansão, principalmente os relacionados à pulverização que representam uma grande parcela dos custos de produção. O direcionamento automático de máquinas agrícolas eleva consideravelmente a qualidade da pulverização, conseqüentemente reduz os custos, o impacto ambiental e aumenta a produção.

Neste trabalho é apresentado um método de obtenção da posição de máquinas agrícolas a partir de imagens da área cultivada. O método proposto está fundamentado nos conceitos de busca antecipada e ponto de fuga para a obtenção da posição. A combinação destes métodos permitiu obter de forma rápida e precisa a posição das imagens da área cultivada.

A abordagem apresentada neste trabalho simplifica os algoritmos de extração da posição, reduzindo o número de iterações necessárias para processar cada imagem capturada. A redução do tempo de processamento permite processar um maior número de imagens a cada segundo, elevando a taxa de dados fornecida e melhorando a resposta dos controladores. Altas taxas de dados de posição possibilitam o desenvolvimento de sistemas de direcionamento para veículos que operam em velocidades elevadas como os pulverizadores.

Os teste estáticos e dinâmicos realizados demonstraram a viabilidade técnica do uso do método proposto neste trabalho no desenvolvimento de um sistema de direcionamento automático baseado em visão computacional para máquinas agrícolas. A utilização do método proposto pode reduzir significativamente os custos dos sistemas de direcionamento quando comparados aos que utilizam receptores GPS de maior precisão necessários em algumas operações agrícolas.

Os resultados preliminares do método proposto mostraram-se aceitáveis. Os testes estáticos demonstraram que o método se comporta satisfatoriamente em imagens contendo falhas, sombras, ervas daninhas e em diferentes tipos de solo e de iluminação. Os resultados também mostraram que a abordagem utilizada pode ser utilizada em diferentes culturas sem a necessidade de modificações ou calibrações prévias.

7.1 Contribuições

Este trabalho contém uma nova abordagem para a obtenção da posição de máquinas agrícolas a partir de imagens das linhas de cultivo. Essa principal contribuição pode ser utilizada no desenvolvimento de sistemas de direcionamento automático de menor custo, permitindo a otimização de processos agrícolas como a pulverização.

Este trabalho também abordou o estudo das redes neurais pulsadas no pré-processamento

mento das imagens de áreas cultivadas. As redes neurais pulsadas estão sendo aplicadas com sucesso em diversas áreas relacionadas ao processamento de imagens. O estudo realizado foi direcionado à remoção de sombras e de ervas daninhas das imagens e para o realce das linhas de cultivo. Uma ferramenta foi desenvolvida para o processamento e para a análise dos resultados produzidos pela rede implementada. Os resultados obtidos mostraram bom desempenho do algoritmo. No entanto, o método de extração da posição proposto neste trabalho demonstrou-se satisfatoriamente robusto, não necessitando de algoritmos de pré-processamento.

O teste e a validação da plataforma de testes utilizada também são contribuições oriundas deste trabalho. Os testes realizados permitiram o melhoramento do hardware original. Modificações foram realizadas para tornar a plataforma de testes mais robusta e prática para a realização de testes em campo.

Como contribuição deste trabalho também pode-se citar o desenvolvimento do sistema de direcionamento automático para a plataforma de testes baseado no método proposto. Os algoritmos de controle desenvolvidos e o hardware utilizado podem ser a base para a validação e testes de futuros trabalhos.

Fruto deste trabalho também foi a publicação do artigo intitulado “*A method for agricultural machine guidance on row crops based on the vanishing point*” apresentado no ICVES07 (*IEEE International Conference on Vehicular Electronics and Safety*) em dezembro de 2007 (NERIS *et al.*, 2007). Em janeiro de 2008 foi submetido ao periódico internacional “*Computers and Electronics in Agriculture*” um artigo sobre o método proposto e os resultados obtidos com a plataforma de testes.

7.2 Trabalhos Futuros

Os direcionamentos futuros relacionados a este trabalho envolvem o projeto e o desenvolvimento de um hardware dedicado à digitalização e ao processamento das imagens. A seleção de uma câmera mais adequada à captura de imagens externas também deve ser realizada.

A adaptação do sistema desenvolvido para a plataforma de testes apresentada na Seção 5.1, deve ser realizada para o sistema AGpilot[®]. Estudos relacionados à utilização de duas câmeras posicionadas nas rodas dianteiras da máquina agrícola também devem ser realizados com o intuito de aumentar a confiabilidade do sistema.

Testes mais detalhados também devem ser realizados em uma área cultivada para a avaliação do método proposto em um ambiente real.

Referências Bibliográficas

- ACCIARESI, H. A.; BALBI, H. V.; BRAVO, M. L.; CHIDICHIMO, H. O. (2003). Response of weed populations to tillage, reduced herbicide and fertilizer rates in wheat (*Triticum aestivum*) production. *Planta Daninha*, v.21, n.1, p.105–110.
- AGCO. (2008). *AutoGuide Brochure*. AGCO Corporation. Disponível em: <http://www.autoguidenav.com/Agco/AutoGuide/AutoGuideUK/Support/Auto-guideBroUS.pdf>. Acesso em: 16 fev.
- ASTRAND, B.; BAERVELDT, A. J. (2005). A vision based row-following system for agricultural field machinery. *Mechatronics*, v.15, p.251–269.
- BARRETO, J.P.; MARTIN, F.; HORAUD, R. (2002). Visual Servoing/Tracking Using Central Catadioptric Images. In: INTERNATIONAL SYMPOSIUM ON EXPERIMENTAL ROBOTICS. 2002. Ischia - Italy, p.863–869.
- BECANOVIC, V. (2000). *Signal and image processing in engineering physics*. 174p. Tese (Doutorado), Royal Institute of Technology (KTH), Stockholm. 2000.
- BENSON, E. R.; STOMBAUGH, T. S.; NOGUCHI, N.; WILL, J. D.; REID, J. F. (1998). An Evaluation of a Geomagnetic Direction Sensor for Vehicle Guidance Precision Agriculture Applications. In: ASAE ANNUAL INTERNATIONAL MEETING. 1998. St. Joseph, Michigan, p.12.
- BENSON, E. R.; REID, J. F.; ZHANG, Q. (2000). Development of an Automated Combine Guidance System. In: ASAE ANNUAL INTERNATIONAL MEETING. 2000. Milwaukee - WI, p.11.
- BENSON, E.R.; REID, J.F.; ZHANG, Q. (2003a). Machine Vision-based Guidance System for Agricultural Grain Harvesters using Cut-edge Detection. *Biosystems Engineering*, v.86, n.4, p.389–398.
- BENSON, E.R.; REID, J.F.; ZHANG, Q. (2003b). Machine Vision-Based Guidance System for an Agricultural Small-Grain Harvest. *Transactions of the ASAE*, v.46, n.4, p.1255–1264.
- BERGQUIST, V. (1999). *Colour Vision and Hue for Autonomous Vehicle Guidance*. 82p. Dissertação (Mestrado), Linköping University, Linköping. 1999.
- BOUGLER, H. T.; FARRELL, B.; YANG, J. A. Y. (2003). Automatic vehicle steering controls: DGPS/INS and magnetic markers. In: AMERICAN CONTROL CONFERENCE. 2003. Denver - Colorado, p.60–65.

- BROGGI, A.; CELLARIO, M.; LOMBARDI, P.; PORTA, M. (2003). An Evolutionary Approach to Visual Sensing for Vehicle Navigation. *IEEE Transactions on Industrial Electronics*, v.50, n.1, p.180–29.
- BUICK, R. (2006). *GPS Guidance and Automated Steering Renew Interest in Precision Farming Techniques*. Tech. rept. Trimble Navigation, Westminster, Colorado, USA.
- CAMPBELL, S. R. (1998). *Synchrony and desynchrony in neural oscillators*. 230p. Tese (Doutorado), Ohio State University, Columbus, OH, USA. 1998.
- CANTONI, V.; LOMBARDI, L.; PORTA, M.; SICARD, N. (2001). Vanishing point detection: representation analysis and new approaches. In: INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING. 2001. Palermo - Italy, p.90–94.
- CHOI, J.Y.; HONG, S.J.; PARK, K.T.; YOO, W.; LEE, M.H. (2002). Lateral Control of Autonomous Vehicle by Yaw Rate Feedback. *KSME International Journal*, v.16, n.3, p.1207–1216.
- E. J. ROSSETTER, J. C. GERDES. (2002). A Study of Lateral Vehicle Control Under a ‘Virtual’ Force Framework. In: INTERNATIONAL SYMPOSIUM ON ADVANCED VEHICLE CONTROL. 2002. Hiroshima - Japan, p.
- ECKHORN, R.; REITBOECK, H. J.; ARNDT, M.; DICKE, P. (1990). Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex. *Neural Computation*, v.2, p.293–307.
- ELKAIM, G.; O’CONNOR, M.; BELL, T.; PARKINSON, B. (1997). System Identification and Robust Control of Farm Vehicles using CDGPS. In: INTERNATIONAL MEETING OF THE SATELLITE DIVISION OF THE INSTITUTE OF NAVIGATION. 1997. Kansas City - Missouri, p.1415–1424.
- FAUGERAS, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA, USA: MIT Press.
- FONTAINE, V.; CROWE, T. (2006). Development of Line-detection Algorithms for Local Positioning in Densely Seeded Crops. *Canadian Biosystems Engineering*, v.48, n.7, p.19–29.
- F.REID, J.; ZHANG, Q.; NOGUCHI, N.; DICKSON, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, v.25, n.1, p.155–167.
- FU, M.; RUAN, J.; DING, H. (2004). Study on virtual path track control for intelligent vehicle. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS. 2004. Washington - DC.
- GOMES, J.; VELHO, L. (1994). *Computação Gráfica: Imagem*. Rio de Janeiro: IMPA/SBM.
- GONZALES, R.C.; WOODS, R. E. (2000). *Processamento de Imagens Digitais*. São Paulo: Edgard Blücher.

- GORDON, G. P.; HOLMES, R. G. (1988). Laser positioning system for off-road vehicles. IN: WINTER MEETING OF THE AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS. 1988. St. Joseph, MI.
- GRASSI JÚNIOR, V. (2002). *Sistema de Visão Omnidirecional Aplicado no Controle de Robôs Móveis*. 104p. Dissertação (Mestrado) , Escola Politécnica da Universidade de São Paulo, São Paulo. 2002.
- GU, X.; YU, D.; ZHANG, L. (2005). Image shadow removal using pulse coupled neural network. *IEEE Transactions on Neural Networks*, v.16, n.3, p.692–698.
- GULDNER, J.; TAN, H. (1997). *On Fundamental Issues Of Vehicle Steering Control For Highway*. Tech. rept. UCB-ITS-PWP-97-11. University of California - Berkeley, Berkeley - CA.
- HAGUE, T.; MARCHANT, J.A.; TILLET, N.D. (2000). Ground Based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture*, v.25, n.1, p.11–28.
- HANA, S.; ZHANG, Q.; NI, J.F. REID. (2004). A Guidance Directrix Approach to Vision-Based Vehicle Guidance Systems. *Computers and Electronics in Agriculture*, v.43, n.3, p.179–195.
- HASSANIEN, A. E.; ALI, J. M. (2004). Digital Mammogram Segmentation Algorithm Using Pulse Coupled Neural Networks. In: ICIG - THIRD INTERNATIONAL CONFERENCE ON IMAGE AND GRAPHICS. 2004. Washington, DC, USA, p.92–95.
- HAVASI, L.; SZIRANYI, T. (2006). Use of Motion Statistics for Vanishing Point Estimation in Camera-Mirror Scenes. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICP06). 2006. Georgia - USA, p.2993–2996.
- HAYKIN, S. (1998). *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall.
- HEARN, D.; BAKER, M. P. (1997). *Computer graphics (2nd ed.): C version*. Upper Saddle River, NJ, USA: Prentice-Hall.
- HEMISPHERE GPS. (2008). *Outback eDrive User Guide*. Hemisphere GPS. Disponível em: <http://www.outbackguidance.com/downloads/files/875-0171-revB1.pdf>. Acesso em: 16 fev.
- JAHNS, G. (1997). Automatic Guidance Agriculture Field Machinery. In: INTERNATIONAL CONFERENCE ON AGRICULTURE ENGINEERING AND TECHNOLOGY. 1997. Dhaka - Bangladesh, p.70–79.
- JAIN, R.; KASTURI, R.; SCHUNCK, B.G. (1995). *Machine Vision*. New York: McGraw-Hill.
- JOHN DEERE. (2008). *AutoTrac Quick Reference Guide*. John Deere. Disponível em: http://stellarsupport.deere.com/en_US/support/atu/ATU_QRG.pdf. Acesso em: 16 fev.

- JOHNSON, J.L; RITTER, D. (1993). Observation of Periodic Waves in a Pulse-Coupled Neural Network. *Optics Letters*, v.18, p.1253–1265.
- JUNYING, Z.; DEFENG, W.; MEIHONG, S.; YUE, W. J. (2004). Output-threshold coupled neural network for solving the shortest path problems. *Science in China Series F-Information Sciences*, v.47, n.1, p.20–33.
- KEICHER, R.; SEUFERT, H. (2000). Automatic Guidance for Agricultural Vehicles in Europe. *Computers and Electronics in Agriculture*, v.25, n.1–2, p.169–194.
- KISE, M.; ZHANG, Q.; MÁŠ, F.R. (2005). A Stereovision-based Crop Row Detection Method for Tractor-automated Guidance. *Biosystems Engineering*, v.90, n.4, p.357–367.
- KOSMOPOULOS, D. I.; CHANDRINOS, K. (2002). Definition and Extraction of Visual Landmarks for Indoor Robot Navigation. In: PROCEEDINGS OF THE SECOND HELLENIC CONFERENCE ON AI (SETN'02). 2002. London - UK, p.401–412.
- KRÖSE, B.; SMAGT, P. VAN DER. (1996). *An introduction to neural networks*. The University of Amsterdam: URL ftp://ftp.informatik.uni-freiburg.de/papers/neuro/ann_intro_smag.ps.gz.
- KUO, B.C. (1985). *Sistemas de Controle Automático*. Rio de Janeiro: Prentice-Hall.
- LEEMANS, V.; DESTAIN, M. F. (2006). Line cluster detection using a variant of the Hough transform for culture row localization. *IMAGE AND VISION COMPUTING*, v.24, n.5, p.541–550.
- LINDBLAD, T.; KINSER, J. M. (2005). *Image Processing Using Pulse-Coupled Neural Networks*. Secaucus, NJ, USA: Springer.
- LINDBLAD, T.; BECANOVIC, V.; LINDSEY, C. S.; SZEKELY, G. (1997). Intelligent detectors modeled from the cat's eye. *Nuclear Instruments and Methods in Physics Research*, v.389, n.1, p.245–250.
- LIPPMANN, R. P. (1988). An introduction to computing with neural nets. *SIGARCH Computer Architecture News*, v.16, n.1, p.7–25.
- MACHADO, P. L. O. A.; SILVA, C. A. (2001). Soil management under no-tillage systems in the tropics with special reference to Brazil. *Nutrient Cycling In Agroecosystems*, v.31, n.1–2, p.119–130.
- MADARI, B.; MACHADO, P. L. O. A.; TORRES, E.; ANDRADE, A. G. DE; VALENCIA, L. I. O. (2005). No tillage and crop rotation effects on soil aggregation and organic carbon in a Rhodic Ferralsol from southern Brazil. *Soil and Tillage Research*, v.80, n.1–2, p.185–200.
- MARQUES FILHO, O.; VIEIRA NETO, H. (1999). *Processamento Digital de Imagens*. Rio de Janeiro: Brasport.
- MINAGAWA, A; TAGAWA, N.; MORIYA, T.; GOTOH, T. (1999). Line Clustering with Vanishing Point and Vanishing Line. IN: ICIAP99 - INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING. 1999. Los Alamitos, CA, USA, p.388.

- MITOV, B. (2007). *VisionLab 3.1 Quick Start*. Mitov Software. Disponível em: <http://www.mitov.com/VisionLab.pdf>. Acesso em: 16 fev.
- MONICO, J. F. G. (2000). *Posicionamento pelo NAVSTAR-GPS*. São Paulo: UNESP.
- NAZMY, T. M. (2004). Evaluation of the PCNN standard model from image processing purposes. *International Journal of Intelligent Computing and Information Sciences*, v.4, n.2, p.101–111.
- NELSON, R. C. (1998). *Flight Stability and Automatic Control*. New York: McGraw-Hill International Editions.
- NERIS, L. O.; RODA, V. O.; TRINDADE JUNIOR, O. (2007). A Method for Agricultural Machine Guidance on Row Crops Based on the Vanishing Point. IN: ICVES07 - IEEE INTERNATIONAL CONFERENCE ON VEHICULAR ELECTRONICS AND SAFETY. 2007. Pequim, China, p.182–187.
- NOGUCHI, N.; ISHII, K.; TERAOKA, H. (1997). Development of an agriculture mobile robot using a geomagnetic direction sensor and image sensors. *Agricultural Engineering Research*, p.1–15.
- OGATA, K. (1993). *Engenharia de Controle Moderno*. Rio de Janeiro: Prentice Hall.
- PATWARDHAN, S.; TAN, H. S.; GULDNER, J. (1997). A general framework for automatic steering control: system analysis. In: AMERICAN CONTROL CONFERENCE. 1997. Albuquerque, NM, USA, p.1598–1602.
- PERNOMIAN, V.A. (2002). *Identificação de Plantas Invasoras em Tempo Real*. 113p. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. 2002.
- PILARSKI, T.; HAPPOLD, M.; PANGELS, H.; OLLIS, M.; FITZPATRICK, K.; STENTZ, A. (2002). The Demeter System for Automated Harvesting. *Autonomous Robots*, v.13, n.1, p.9–20.
- PINTO, F. A. C.; REID, J. F. (1998). Heading Angle and Offset Determination Using Principal Component Analysis. In: ASAE ANNUAL INTERNATIONAL MEETING. 1998. Orlando - Florida, p.9.
- PINTO, F. A. C.; REID, J. F.; ZHANG, Q.; NOGUCHI, N. (1999). Guidance Parameter Determination Using Artificial Neural Network Classifier. In: ASAE/CSAE-SCGR ANNUAL INTERNATIONAL MEETING. 1999. Toronto - Canada, p.1–10.
- REID, J.F.; SEARCY, S.W. (1987). Vision-Based Guidance of an Agricultural Tractor. *IEEE Control Systems Magazine*, v.7, n.12, p.39–43.
- RICHEY, C. B. (1959). “Automatic pilot” for farm tractors. *Agricultural Engineering*, v.2, n.40, p.78–79.
- RUGHOOPUTH, H. C. S.; BOOTUN, H.; RUGHOOPUTH, S. D. D. V. (2003). Pulse coded neural network for sign recognition for navigation. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY. 2003. p.89–94.

- RUSHING, K. (1971). Developing the driverless tractor. *Agricultural Engineering*, v.5, n.52, p.260–262.
- SHUFELT, J. A. (1999). Performance Evaluation and Analysis of Vanishing Point Detection Techniques. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, v.21, n.3, p.282–288.
- SIMOND, N.; RIVES, P. (2003). Homography from a vanishing point in urban scenes. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOT AND SYSTEM (IROS03). 2003. Nevada - USA, p.1005–1010.
- SOTELO, M. A. (2003). Lateral control strategy for autonomous steering of Ackerman-like vehicles. *Robotics and Autonomous Systems*, v.45, p.223–233.
- STENTIFORD, F. (2006). Attention-Based Vanishing Point Detection. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICP06). 2006. Georgia - USA, p.417–420.
- STOMBAUGH, T. S.; SHEARER, S. A. (2001). DGPS-Based Guidance of High-Speed Application Equipment. IN: ASAE ANNUAL INTERNATIONAL MEETING. 2001. Sacramento, California, USA.
- TRUYENQUE, M.A.Q. (2005). *Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador*. 101p. Dissertação (Mestrado) , Departamento de Informática - Pontifícia Universidade Católica, Rio de Janeiro. 2005.
- VINCE, J. (2006). *Mathematics for computer graphics*. London: Springer.
- WILL, J. D.; STOMBAUGH, T. S.; BENSON, E. R.; NOGUCHI, N.; REID, J. F. (1998). Development of a Flexible Platform for Agricultural Automatic Guidance Research. ASAE - ANNUAL INTERNATIONAL MEETING. 1998. Orlando, FL.
- WOEBBECKE, D. M.; MEYER, G. E.; BARGEN, K. VON; MORTENSEN, D. A. (1995). Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASAE*, v.38, n.1, p.259–269.
- YOUNG, M. P. (1994). Large-scale organization of the primate cortical visual system. In: SPIE - THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING. 1994. p.185–193.
- YU, B.; ZHANG, L. (2004). Pulse-coupled neural networks for contour and motion matchings. *IEEE Transactions on Neural Networks*, v.15, p.1186–1201.
- ZHANG, F.; YING, Y.; JIANG, H.; SHIN, B. (2005). Correlation analysis-based image segmentation approach for automatic agriculture vehicle. *Zhejiang University Science*, v.6A, n.10, p.1158–1162.
- ZHANG, Q.; QIU, H. (2004). A dynamic path search algorithm for tractor automatic navigation. *ASAE*, v.4, n.2, p.639–646.

Glossário

CCD - *Charge Coupled Device* ou Dispositivo de Carga Acoplada. Dispositivo formado por elementos sensíveis ao espectro de energia eletromagnético onde cada elemento produz um sinal elétrico proporcional à luz incidente. Cada elemento sensitivo no CCD é baseado num capacitor MOS (*Metal Oxide Semiconductor*) que representa um *pixel* da imagem. Um digitalizador converte o sinal elétrico analógico produzido na saída do sensor em um sinal digital.

CEP - *Circular Error Probability* ou Erro Circular Provável. Valor do raio de um círculo, centrado na posição atual, do qual existem 50% de probabilidade das posições estarem localizadas em seu interior.

CMOS - *Complementary Metal Oxide Semiconductor* ou Metal Óxido Semicondutor Complementar. Dispositivo semelhante ao CCD exceto pelo fato que os elementos sensíveis serem compostos por componentes convencionais como fotodiodos, transistores e capacitores que podem ser integrados para formar um circuito de imageamento de menor custo e de baixo consumo de energia. Os sensores CMOS, no entanto, apresentam imagens de menor qualidade e mais ruidosas quando comparados aos sensores CCD.

Dither - oscilação intencional de pequena amplitude e de alta frequência superposto ao sinal de controle utilizada para eliminar o efeito do atrito.

DLL - *Dynamic Link Library* ou Biblioteca de Ligação Dinâmica. Implementação do conceito de bibliotecas compartilhadas nos sistemas operacionais Windows realizada pela Microsoft.

Espaço de cor HSI - modelo onde a cor é representada em termos de matiz (*Hue* - H), saturação (*Saturation* - S) e intensidade (*Intensity* - I). Nesse modelo, o componente de intensidade é desacoplado da informação de cor e os componentes de matiz e saturação são intimamente relacionados à percepção humana de cores.

Espaço de cor RGB - modelo de cor formada a partir das componentes espectrais primárias vermelho (*Red* - R), verde (*Green* - G) e azul (*Blue* - B). Atualmente esse modelo é muito utilizado na reprodução (monitores) e na captura (câmeras coloridas) de imagens. Apesar de ser muito utilizado, esse modelo de cor não é muito indicado para o processamento de imagens digitais.

Fire Wire - IEEE 1394. Interface serial de alta velocidade para computadores pessoais e aparelhos digitais de áudio e vídeo que oferece serviços de dados em tempo real.

GPS - *Global Positioning System* ou Sistema de Posicionamento Global. Sistema de rádio navegação baseado em satélites artificiais. A posição é obtida através da triangulação das posições conhecidas de um conjunto de satélites em relação a um referencial inercial. Esse sistema consiste de 24 satélites em 6 órbitas circulares a 20.200 km acima da superfície terrestre, com um período de 12 horas.

Heading - direção do movimento de um objeto definido como um ângulo, em graus, em relação ao norte.

Histerese - não linearidade encontrada na caracterização de instrumentos e que corresponde à diferença entre saídas obtidas a partir de um mesmo valor do sinal de entrada.

Limiarização (*thresholding*) - técnica de processamento de imagens que consiste em separar as regiões de uma imagem quando a mesma apresenta duas classes: o fundo e o objeto. A forma mais simples de limiarização é a bipartição do histograma convertendo em branco o *pixel* cujo tom de cinza é maior ou igual a um certo valor de limiar e os demais em pretos.

Marcador de espuma - dispositivo localizado nas pontas da barra de pulverização capaz de liberar uma espuma branca para auxiliar a orientação de máquinas agrícolas em faixas adjacentes.

Motorreductor - motor elétrico acoplado a um sistema reductor composto de engrenagens.

PCM - *Pulse Code Modulation* ou Modulação por Código de Pulsos. Representação digital de um sinal analógico no qual a magnitude do sinal é obtida em intervalos regulares e transformada em uma série digital de símbolos.

PID - *Proportional Integral Derivative controller* ou controlador Proporcional Integral Derivativo. Método clássico de controle em malha fechada de sistemas dinâmicos. O princípio básico do controle PID é agir sobre a variável a ser controlada através de uma combinação apropriada de três ações de controle: a ação de controle proporcional, a ação de controle integral e a ação de controle derivativa.

Pitch - ângulo produzido pelo movimento de arfagem em torno do eixo transversal do veículo.

PWM - *Pulse Width Modulation* ou Modulação por Largura de Pulso. Método que consiste em variar a largura do pulso da portadora, proporcionalmente ao sinal modulado, mantendo constantes a amplitude e o intervalo de tempo a que os pulsos se repetem.

RMS - *Root Mean Square* ou Erro Médio Quadrático. Raiz quadrada da média dos quadrados dos erros.

Roll - ângulo produzido pelo movimento de rolagem executado em torno do eixo longitudinal do veículo.

RS232 - *Recommended Standard 232*. Padrão para troca serial de dados binários entre um terminal de dados e um comunicador de dados. A codificação mais comumente utili-

zada é o “*start-stop* assíncrono” que usa um bit de início, seguido por sete ou oito bits de dados, possivelmente um bit de paridade, e um ou dois bits de parada sendo, nesse caso, necessários 10 bits para enviar um único caractere.

RS485 - padrão de comunicação serial que utiliza linhas de transmissão balanceadas. Esse padrão é extremamente imune a ruídos e permite a comunicação em rede.

RTK - *Real Time Kinematic* ou Posicionamento Cinemático em Tempo Real. Técnica de correção da posição em tempo real baseada no uso das observações de fase da portadora dos sinais do GPS onde uma estação de referência fornece correções em tempo real, possibilitando um nível centimétrico de posicionamento.

Set point - valor de referência utilizado em um controlador PID. É o valor que se deseja para a saída de um sistema.

Transformada de Hough - método para detecção de formas geométricas em imagens digitais que são facilmente parametrizadas como as linhas e os círculos. O princípio básico consiste em definir um mapeamento entre o espaço de imagem e o espaço de parâmetros.

USB - *Universal Serial Bus* ou Barramento Serial Universal. Protocolo de comunicação serial usado para conectar dispositivos periféricos a um computador.

VCL - *Visual Component Library* ou Biblioteca de Componentes Visuais. Conjunto de classes que consistem de objetos reutilizáveis incluindo objetos padrão de interface com o usuário, gerenciamento de dados, gráficos e multimídia e gerenciamento de arquivos.

Waypoints - pontos ou coordenadas que definem o início ou o fim de um trajeto.

Apêndice A

Implementação do Modelo ICM

Este apêndice apresenta alguns detalhes de implementação do modelo ICM descrito do Capítulo 2 e utilizado em alguns dos experimentos realizados neste trabalho. O modelo ICM foi implementado de forma a se adaptar a diferentes resoluções de imagens podendo ser otimizado com o intuito de reduzir o tempo de processamento.

A Figura A.1 apresenta a interface do software desenvolvido. As rotinas de implementação encontram-se a seguir.

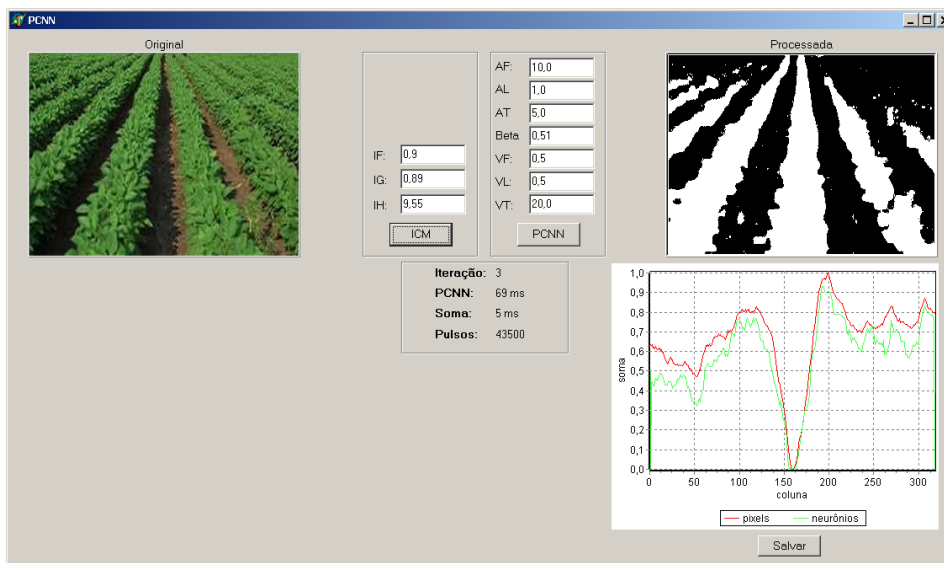


Figura A.1: Interface da ferramenta de teste desenvolvida.

```
// Implementação baseada em:
// http://web.archive.org/web/20010727031459/ib3.gmu.edu/kinser/pcnn/src/pcnn2d.cc
// A 2D Pulse-Coupled Neural Network
// Jason Kinser
// 15 April 1996
```

```
unit PCNN;
```

```
interface
```

```
uses Math;
```

```
type
```

```
  TMatrix = array of array of single;
```

```
  TPCNN = class(TObject)
```

```
  private
```

```
    vert: Integer;           // Largura da PCNN
```

```
    horz: Integer;          // Altura da PCNN
```

```
    K: TMatrix;             // Kernel
```

```
  procedure StdK(kv, kh: integer);
```

```
  procedure Fill(m: TMatrix; val: Single );
```

```
  function Convolve(m1, m2: TMatrix): TMatrix;
```

```
  public
```

```
    v_f: single;            //
```

```
    v_l: single;            //
```

```
    v_t: single;            //
```

```
    beta: single;           // Parâmetros da rede PCNN
```

```
    a_f: single;            //
```

```
    a_l: single;            //
```

```
    a_t: single;            //
```

```
    i_f: single;            //
```

```
    i_g: single;           // Parâmetros da rede ICM
```

```
    i_h: single;            //
```

```
    L: TMatrix;             // Linking
```

```
    U: TMatrix;             // Feeding
```

```
    F: TMatrix;             // Feeding
```

```
    T: TMatrix;             // Threshold
```

```
    Y: TMatrix;             // Saída da PCNN
```

```
    W: TMatrix;             // Resultado da convolução
```

```
  constructor Create(vt, hz: integer);
```

```
  destructor Destroy; override;
```

```
  function Iterate(S: TMatrix): Integer;
```

```
  function ICMIterate(S: TMatrix): Integer;
```

```
  procedure Reset;
```

```
end;
```

```
implementation
```

```

constructor TPCNN.Create(vt, hz: Integer);
begin
  inherited Create;

  vert := vt;   horz := hz;

  a_f := 10.0; a_l := 1.0;   a_t := 5.0;
  beta:= 0.1;
  v_f := 0.5;   v_l := 0.5;   v_t := 20.0;

  i_f := 0.9;   i_g := 0.89;   i_h := 60;

  SetLength(F, vert, horz); SetLength(L, vert, horz);
  SetLength(U, vert, horz); SetLength(Y, vert, horz);
  SetLength(T, vert, horz); SetLength(W, vert, horz);
  SetLength(R, vert, horz);

  Fill(F, 1); Fill(L, 0);
  Fill(U, 0); Fill(Y, 0);
  Fill(T, 1); Fill(W, 1);
  Fill(R, 0);

  StdK(3,3);
end;

destructor TPCNN.Destroy;
begin
  Finalize(F); Finalize(L);
  Finalize(U); Finalize(Y);
  Finalize(T); Finalize(W);
  Finalize(K); Finalize(R);

  inherited Destroy;
end;

procedure TPCNN.Reset;
begin
  Fill(F, 1); Fill(L, 0);
  Fill(U, 0); Fill(Y, 0);
  Fill(T, 1); Fill(W, 1);
  Fill(R, 0);
end;

procedure TPCNN.Fill(m: TMatrix; val: Single );
var
  i, j: integer;
begin
  for i := 0 to High(m) do
    for j := 0 to High(m[0]) do
      m[i,j] := val;

```

```
end;
```

```
// Kernel
```

```
procedure TPCNN.Stdk(kv, kh: integer);
```

```
var
```

```
  i, j: integer;
```

```
  val: single;
```

```
begin
```

```
  SetLength(K, kv, kh);
```

```
  for i := 0 to kv - 1 do
```

```
    for j := 0 to kh - 1 do
```

```
      begin
```

```
        val := Hypot(i - kv / 2, j - kh / 2 );
```

```
        if (val <> 0.0)
```

```
        then val := 1.0 / val
```

```
        else val := 1.0;
```

```
        K[i,j] := val;
```

```
      end;
```

```
end;
```

```
//Convolução
```

```
function TPCNN.Convolve(m1, m2: TMatrix):TMatrix;
```

```
var
```

```
  sum: single;
```

```
  ans: TMatrix;
```

```
  dv, dh: integer;
```

```
  n, i, j, k, l: integer;
```

```
begin
```

```
  dv := (High(m2) + 1) div 2;
```

```
  dh := (High(m2[0]) + 1) div 2;
```

```
  SetLength(ans, High(m1) + 1, High(m1[0]) + 1);
```

```
  for i := 0 to High(m1) do
```

```
    for j := 0 to High(m1[0]) do
```

```
      begin
```

```
        sum := 0.0;          n := 0;
```

```
        for k := -dv to dv do
```

```
          for l := -dh to dh do
```

```
            begin
```

```
              if (((i + k) >= 0) and ((i + k) < vert) and ((j + l) >= 0) and ((j + l) < horz))
```

```
              then begin
```

```
                sum := sum + m1[i + k, j + l] * m2[k + dv, l + dh];
```

```
                Inc(n);
```

```
            end;
```

```
          end;
```

```
        ans[i, j] := sum;
```

```

    end;

    result := ans;

    Finalize(ans);
end;

//PCNN
function TPCNN.Iterate(S: TMatrix): Integer;
var
    sum, i, j : integer;
    work: TMatrix;
begin
    SetLength(work, vert, horz);

    work := Convolve(Y, K);

    sum := 0;
    for i := 0 to vert - 1 do
        for j := 0 to horz - 1 do
            begin
                F[i,j] := Exp(-1.0 / a_f) * F[i,j] + v_f * work[i,j] + S[i,j];
                L[i,j] := Exp(-1.0 / a_l) * L[i,j] + v_l * work[i,j];
                U[i,j] := F[i,j] * (1.0 + beta * L[i,j]);

                if U[i,j] > T[i,j]
                then Y[i,j] := 1.0
                else Y[i,j] := 0.0;

                T[i,j] := Exp(-1.0 / a_t) * T[i,j] + v_t * Y[i,j];

                sum := sum + Trunc(Y[i,j]);
            end;
        end;
    end;

    result := sum;
end;

//ICM
function TPCNN.ICMIterate(S: TMatrix): Integer;
var
    W: TMatrix;
    sum, i, j : integer;
begin
    SetLength(W, vert, horz);

    W := Convolve(Y, K);

    sum := 0;
    for i := 0 to vert - 1 do
        for j := 0 to horz - 1 do
            begin

```



```
F[i,j] := i-f * F[i,j] + W[i,j] + S[i,j];
```

```
if F[i,j] > T[i,j]
```

```
then Y[i,j] := 1.0
```

```
else Y[i,j] := 0.0;
```

```
T[i,j] := i-g * T[i,j] + i-h * Y[i,j];
```

```
sum := sum + Trunc(Y[i,j]);
```

```
end;
```

```
result := sum;
```

```
end;
```

```
end.
```

Apêndice B

Funções do Sistema Desenvolvido

Este apêndice tem como objetivo apresentar as principais funções do sistema de direcionamento automático, baseado no método proposto neste trabalho, implementado para a plataforma de testes descrita no Capítulo 5. O sistema foi implementado orientado a objetos utilizando a linguagem de programação *object pascal*.

B.1 Gerador PWM

```
unit UPWM;

interface

uses

  Classes , Windows;

type

  TPWM = class(TThread)
    period      : Extended;
    dutyCycle: Extended;
    diff        : Extended;
  private
    { Private declarations }
  protected
    procedure Execute; override;
  public
    constructor Create(periodValue: Extended);
    procedure SetDutyCycle(dcValue : Extended);
  end;
```

```
procedure Out32(wAddr:word; bOut:byte); stdcall; external 'inpout32.dll'
```

```
implementation
```

```
constructor TPWM.Create(periodValue: Extended); // ms
```

```
begin
```

```
    period := periodValue * 1000; // period in us
```

```
    dutyCycle := 0;
```

```
    diff := 0;
```

```
    FreeOnTerminate := False;
```

```
    inherited Create(True);
```

```
end;
```

```
procedure TPWM.SetDutyCycle(dcValue : Extended);
```

```
begin
```

```
    dutyCycle := (dcValue * 1000) - 5; // duty cycle in us (-5 -> compensation)
```

```
    diff := (period - dutyCycle) - 5; // -5 -> compensation
```

```
end;
```

```
procedure TPWM.Execute;
```

```
var
```

```
    beginTime, endTime, freq: Int64;
```

```
    auxdutyCycle, auxdiff : Extended;
```

```
begin
```

```
    QueryPerformanceFrequency(freq);
```

```
    while (not Terminated) do
```

```
    begin
```

```
        auxdutyCycle := dutyCycle;
```

```
        auxdiff:=diff;
```

```
        //wait duty cycle
```

```
        QueryPerformanceCounter(beginTime);
```

```
        repeat
```

```
            Out32($378,$80); // turn on D7 - PIN 9 (turn left)
```

```
            QueryPerformanceCounter(endTime);
```

```
        until (((endTime - beginTime) * 1000000) / freq) >= auxdutyCycle;
```

```
        //wait period
```

```
        QueryPerformanceCounter(beginTime);
```

```
        repeat
```

```
            Out32($378,$00); // turn off D1..D7 - PIN 2..9 (turn off)
```

```
            QueryPerformanceCounter(endTime);
```

```
        until (((endTime - beginTime) * 1000000) / freq) >= auxdiff;
```

```
    end;
```

```
end;
```

```
end.
```

B.2 Controlador PID

```

// Implementação baseada em:
// http://www.rabbit.com/downloads/files/PID.C
// Proportional-integral-derivative function
// Greg Young, Z-World

unit UPID;

interface

type
  TPID = class
  private
    setPoint    : real;           // Desired Value
    proportional: real;          // Proportional Const
    integral    : real;          // Integral Const
    derivative  : real;          // Derivative Const
    lastError   : real;          // Error[-1]
    prevError   : real;          // Error[-2]
    sumError    : real;          // Sums of Errors
    dError      : real;
    error       : real;
    offSet      : real;          // SetPoint OffSet
  protected
    { Protected declarations }
  public
    constructor Create;
    procedure Initialize(p:real; i:real; d:real; o:real);
    procedure SetSetPoint (value: real);
    procedure SetOffSet (value: real);
    function Iterate(value: real):real;
    procedure ResetI;
  end;

implementation

constructor TPID.Create;
begin
  Initialize(0, 0, 0, 0);
end;

procedure TPID.Initialize(p:real; i:real; d:real; o:real);
begin
  proportional:= p;
  integral     := i;
  derivative   := d;
  offSet       := o;
  setPoint     := 0;
  lastError    := 0;
  prevError    := 0;
  sumError     := 0;

```

```

end;

procedure TPID.SetSetPoint (value: real);
begin
  setPoint := value;
end;

procedure TPID.SetOffSet (value: real);
begin
  offSet := value;
end;

procedure TPID.ResetI;
begin
  sumError := 0;
end;

function TPID.Iterate(value: real): real;
begin
  error      := setPoint - (value + offSet);
  sumError   := sumError + error;
  dError     := lastError - prevError;
  prevError  := lastError;
  lastError  := error;

  result     := (proportional * error + Integral * sumError + Derivative * dError);
end;

end.

```

B.3 Controlador de Ângulo

```

unit UAngControl;

interface

uses
  Classes, Windows, CPort, UDLEPacket, UPID, UPWM, UFIR, Graphics;

const
  MAXDTC      : real = 1800;           //1.800ms
  INIDIRDTC   : real = 1572;         //1.572ms - start movements
  INILEFTDTC  : real = 1348;         //1.348ms - start movements
  MEDDTC      : real = 1500;         //1.500ms
  MINDTC      : real = 1120;         //1.120ms

type
  TAngControl = class(TThread)
    comPort    : TComPort;
    angPacket  : TDLEPacket;
    pid        : TPID;

```

```

    pwm      : TPWM;
    filt     : TFIR;
    ang      : Single;
    pos      : Single;
private
    { Private declarations }
protected
    beginTime, endTime, tickPersec: Int64;
    procedure Execute; override;
    procedure SendGetAngCmd;
    procedure SteerWheel;
    procedure CalcSteeringAngle;
    procedure CalcPWM;
    function BytesToInt(position, nbytes: byte): integer;
    procedure UpdateAngle;
public
    constructor Create(port: String; baud: TBaudRate);
end;

implementation

uses UMain;

constructor TAngControl.Create(port: String; baud: TBaudRate);
begin
    filt := TFIR.Create;
    filt.Initialize(0.25, 0.25);

    pwm := TPWM.Create(22.0);           // 22ms
    pwm.SetDutyCycle(1.5);             // 1.5ms
    pwm.Priority := tpTimeCritical;

    comPort := TComPort.Create(nil);
    comPort.BaudRate:= baud;
    comPort.Port    := port;
    comPort.Events  := [];
    comPort.Timeouts.ReadInterval      := -1;
    comPort.Timeouts.ReadTotalMultiplier:= -1;
    comPort.Timeouts.ReadTotalConstant := 7;    // 7ms
    comPort.Open;

    angPacket := TDLEPacket.Create;

    pid := TPID.Create;

    FreeOnTerminate := False;

inherited Create(True);

    pwm.Resume;
end;

```

```

procedure TAngControl.SendGetAngCmd;
var
    aux: byte;
begin
    aux:= DLE;
    comPort.Write(aux,1);
    aux:= STX;
    comPort.Write(aux,1);

    aux := $0B;
    comPort.Write(aux,1);

    aux:= DLE;
    comPort.Write(aux,1);
    aux:= ETX;
    comPort.Write(aux,1);
end;

function TAngControl.BytesToInt(position, nbytes: byte): integer;
var
    i, n: integer;
begin
    n := 0;
    i := position + nbytes - 1;
    while (i >= position + 1) do
        begin
            n := n + angPacket.rxp[i];
            n := n shl 8;
            dec(i);
        end;

    Result := n + angPacket.rxp[i];
end;

procedure TAngControl.CalcSteeringAngle;
begin
    if (angPacket.rxp[3] and 128) = 128
    then begin
        angPacket.rxp[3] := (angPacket.rxp[3] and 127);
        ang := BytesToInt(2, 2) / 4;
    end
    else begin
        ang := (BytesToInt(2, 2) / 4)*(-1);
    end;

    Synchronize(UpdateAngle);

    QueryPerformanceCounter(endTime);
    if frmMain.chbGrafAng.Checked
    then frmMain.seriesAngle.AddXY((endTime - beginTime) / tickPersec, ang);
end;

```

```

procedure TAngControl.UpdateAngle;
begin
    QueryPerformanceCounter(endTime);
    if frmMain.chbGrafAng.Checked
    then frmMain.seriesAngle.AddXY((endTime - beginTime) / tickPersec, ang);
end;

procedure TAngControl.SteerWheel;
begin
    CalcSteeringAngle;

    pos := pid.Iterate(ang);

    if pos >= 70
    then begin
        pos := 70;
        pid.ResetI
    end
    else if pos < -70
    then begin
        pos := -70;
        pid.ResetI
    end;

    CalcPWM;
end;

procedure TAngControl.CalcPWM;
begin
    if pos = 0
    then pwm.SetDutyCycle(MEDDTC / 1000)
    else if pos > 0
    then pwm.SetDutyCycle((((MAXDTC - INIDIRDTC) * pos) / 100.0) + INIDIRDTC) / 1000)
    else if pos < 0
    then pwm.SetDutyCycle((((MINDTC - INILEFTDTC) * pos) / -100.0) + INILEFTDTC) / 1000);
end;

procedure TAngControl.Execute;
var
    buff: Byte;

begin
    QueryPerformanceFrequency(tickPersec);
    QueryPerformanceCounter(beginTime);

    while (not Terminated) do
    begin
        comPort.ClearBuffer(True, True);
        comPort.Read(buff, 1); // wait 7ms... + 3ms of proc = 10 ms -> 100hz

        comPort.ClearBuffer(True, True);
        SendGetAngCmd;
    end;

```



```

while true do
begin
  if comPort.Read(buff,1) <> 1
  then Break;

  if angPacket.Parse(buff)
  then begin
    SteerWheel;
    Break;
  end;
end;
end;

comPort.Close;
comPort.Free;

angPacket.Free;

pid.Free;

pwm.Terminate;
pwm.WaitFor;
pwm.Free;

filt.Free;
end;

end.

```

B.4 Filtro

```

// Implementação baseada em:
// http://www.maxim.abel.co.uk
// Simple resonant low-pass filter
// Paul Kellett

unit UFIR;

interface

type
  TFIR = class
  private
    f : single;
    k : single;
    fb : single;
    buf0: single;
    buf1: single;
  protected
    { Protected declarations }

```

```

public
  constructor Create;
  procedure Initialize( fValue: real; kValue: real);
  function Iterate(input: real): real;
end;

implementation

constructor TFIR.Create;
begin
  Initialize(0.25, 0.25);
end;

procedure TFIR.Initialize( fValue: real; kValue: real);
begin
  f := fValue;
  k := kValue;
  fb := k + k / (1.0 - f);
  buf0:= 0.0;
  buf1:= 0.0;
end;

function TFIR.Iterate(input: real): real;
begin
  buf0 := buf0 + f * (input - buf0 + fb * (buf0 - buf1));
  buf1 := buf1 + f * (buf0 - buf1);

  result :=buf1;
end;

end.

```

B.5 Controlador de Rota

```

unit UMain;

interface

uses
  .
  .
  .

procedure TfrmMain.bntCtrlAngClick(Sender: TObject);
var
  FormImage: TBitmap;

begin
  if (bntCtrlAng.Tag = 0)
  then begin
    thrAngCTRL := TAngControl.Create(comPort.Port, comPort.BaudRate);
    thrAngCTRL.pid.Initialize(StrToFloat(edtKP.Text), StrToFloat(edtKI.Text),

```

```

        StrToFloat(edtKD.Text), StrToFloat(edtOffset.Text));

bntCtrlAng.Tag := 1;
bntCtrlAng.Caption := 'Turn Off';

pnlPID.Enabled := False;
pnlTunning.Enabled := True;
pnlSetPoint.Enabled := True;
seriesAngle.Clear;

tabMntAng.Enabled := False;

btnOpenPort.Enabled := False;

thrAngCTRL.Priority := tpTimeCritical;
thrAngCTRL.Resume;
end
else begin
    thrAngCTRL.Terminate;
    thrAngCTRL.WaitFor;
    thrAngCTRL.Free;

    bntCtrlAng.Tag := 0;
    bntCtrlAng.Caption := 'Turn On';

    pnlPID.Enabled := True;
    pnlTunning.Enabled := False;
    pnlSetPoint.Enabled := False;
    tabMntAng.Enabled := True;

    FormImage := GetFormImage;
    FormImage.SaveToFile(FormatDateTime('' + ExtractFilePath(Application.EXENAME) +
        '\Data\Ang"dd"- "mm"- "yyyy"- "hh"- "nn"- "ss ".bmp' , Now));
    FormImage.Free;

    SavechartValues(seriesAngle , FormatDateTime('' + ExtractFilePath(Application.EXENAME) +
        '\Data\Ang"dd"- "mm"- "yyyy"- "hh"- "nn"- "ss ".txt' , Now));

    btnOpenPort.Enabled := True;
end;
end;

procedure TfrmMain.bntCtrlPathClick(Sender: TObject);
var
    FormImage: TBitmap;

begin
    if (bntCtrlPath.Tag = 0)
    then begin
        filtPath.Initialize(0.65, 0.25);
    end;
end;

```

```

bntCtrlPath.Tag := 1;
bntCtrlPath.Caption := 'Turn Off';

vlCapture.Enabled := True;

seriesPath.Clear;
seriesDist.Clear;

bntCtrlAngClick(Sender);
pnlPathPID.Enabled := False;
tabCtrlAng.Enabled := False;

pathPID.Initialize(StrToFloat(edtPathCtrlKP.Text), StrToFloat(edtPathCtrlKI.Text),
  StrToFloat(edtPathCtrlKD.Text), StrToFloat(edtOffSet.Text));

QueryPerformanceFrequency(pathCtrlTickPersec);
QueryPerformanceCounter(pathCtrlIniTm);

vlGenFilt.Enabled := True;
end
else begin
  bntCtrlPath.Tag := 0;
  bntCtrlPath.Caption := 'Turn On';

  if bntCtrlAng.Tag = 1
  then bntCtrlAngClick(Sender);

  vlGenFilt.Enabled := False;
  vlCapture.Enabled := False;

  SavechartValues(seriesDist, FormatDateTime('' + ExtractFilePath(Application.EXENAME) +
    '\Data\Path"dd"-mm"-yyyy"_"hh"-nn"-ss".txt'', Now));

  vlImgDisp.Graph.SaveToFile(FormatDateTime('' + ExtractFilePath(Application.EXENAME) +
    '\Data\Graph"dd"-mm"-yyyy"_"hh"-nn"-ss".grf'', Now));

  FormImage := GetFormImage;
  FormImage.SaveToFile(FormatDateTime('' + ExtractFilePath(Application.EXENAME) +
    '\Data\Path"dd"-mm"-yyyy"_"hh"-nn"-ss".bmp'', Now));
  FormImage.Free;

  tabCtrlAng.Enabled := True;
  pnlPathPID.Enabled := True;
end;
end;

procedure TfrmMain.vlGenFiltProcessData(Sender: TObject;
  InBuffer: IVLImageBuffer; var OutBuffer: IVLImageBuffer;
  var SendOutputData: Boolean);

var
angCTRL, sum, min, max : real;

```

```

iPath, jPath, col : Integer;

begin
  QueryPerformanceFrequency(procTickPerSec);
  QueryPerformanceCounter(procIniTime);

  seriesPath.Clear;

  min := 150000;
  max := -1;
  col := -1;

  for jPath := 0 to InBuffer.Width - 1 do
  begin
    sum := 0;
    for iPath := 0 to InBuffer.Height - 1 do
      sum := sum + ((2*InBuffer.Green[jPath,iPath]) - InBuffer.Red[jPath,iPath] - InBuffer.Blue[jPath,iPath]);

      if min > sum
      then begin
        min := sum;
        col := jPath;
      end;

      if chbGrafSoma.Checked
      then seriesPath.AddXY(jPath,sum);
    end;

    col := Trunc(filtPath.Iterate(col));

    angCTRL := pathPID.Iterate(-(col - 160));

    if angCTRL > 20
    then angCTRL := 20
    else if angCTRL < -20
    then angCTRL := -20;

    memol.Lines.Add(IntToStr(Trunc(angCTRL)));
    thrAngCTRL.pid.SetSetPoint(Trunc(angCTRL));

    QueryPerformanceCounter(procEndTime);
    lblTime.Caption := IntToStr(Round((procEndTime - procIniTime) / procTickPerSec * 1000));
    lblCol.Caption := IntToStr(col - 160);

    QueryPerformanceCounter(pathCtrlEndTm);
    if chbGrafDist.Checked
    then seriesDist.AddXY((pathCtrlEndTm - pathCtrlIniTm) / pathCtrlTickPerSec, col - 160);
  end;
  .
  .
  .
end.

```

Apêndice C

Testes do Sistema AGpilot[®]

O objetivo deste Apêndice é apresentar os resultados obtidos com o sistema AGpilot[®] descrito no Capítulo 5. Os resultados apresentados referem-se aos testes do sistema utilizando os dados de posição fornecidos pelo receptor GPS.

Diversos testes foram realizados em campo para avaliar o desempenho, a precisão e para obter a velocidade máxima de operação suportada pelo sistema.

C.0.1 Resultados

Os testes iniciais do sistema AGpilot[®] foram realizados em uma área de 200 *ha* da Fazenda Campo Bom a ser dessecada para o cultivo de soja. A área selecionada é relativamente plana e com poucas irregularidades. Os controladores PID foram ajustados utilizando as técnicas desenvolvidas por Ziegler e Nichols (OGATA, 1993; NELSON, 1998).

Inicialmente, ajustou-se o controlador de fluxo do bloco hidráulico e o controlador de ângulo das rodas de forma a se aproximar da velocidade de esterçamento alcançada por um operador. O gráfico ilustrado na Figura C.1 apresenta 5 alterações do *set point* do controlador de ângulo realizadas consecutivamente durante um dos testes: -5° , 5° , -10° , 10° e 0° . A velocidade de esterçamento foi de 8 *graus/s*. Os testes mostraram que

apenas a utilização de um controlador proporcional é suficiente para controlar o ângulo das rodas do pulverizador.

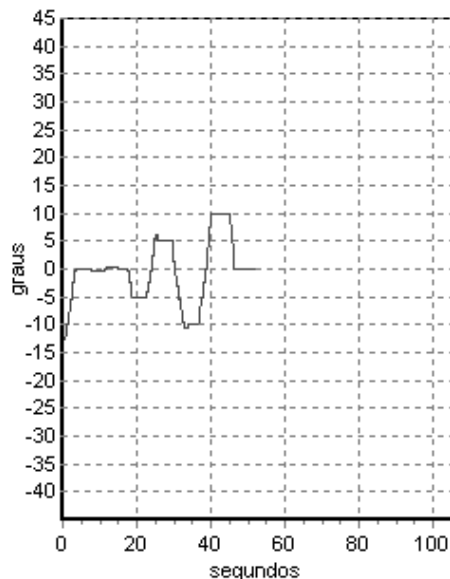


Figura C.1: Teste do módulo Controlador de Ângulo ($K_P = 2,0$).

A segunda etapa de ajustes abordou a calibração do controlador de rota. Um filtro FIR (*Finite Impulse Response*) foi utilizado na distância real fornecida pelo receptor GPS e no *heading* fornecido pela bússola para suprimir os eventuais sobressaltos causados principalmente pela ausência de um método de correção da posição.

As figuras a seguir apresentam as distâncias reais e as distâncias à frente do pulverizador em três dos testes realizados. O controlador foi ajustado de forma a responder o mais suave possível para garantir um nível satisfatório de conforto dentro da cabine. A constante utilizada no cálculo da distância à frente foi obtida empiricamente tendo como ponto inicial o trabalho de Benson *et al.* (1998). Os ganhos dos controladores e a constante de cálculo da distância à frente utilizados em todos os testes apresentados estão contidos na Tabela C.1.

No primeiro teste, ilustrado na Figura C.2, o pulverizador foi posicionado próximo do trajeto de trabalho. O desvio lateral máximo foi de aproximadamente 25 *cm* e a velocidade máxima foi de 10 *km/h*.

Tabela C.1: Parâmetros do módulo Controlador de Rota.

Descrição	Valor
KP	2
KI	0
KD	0,5
DF	2s

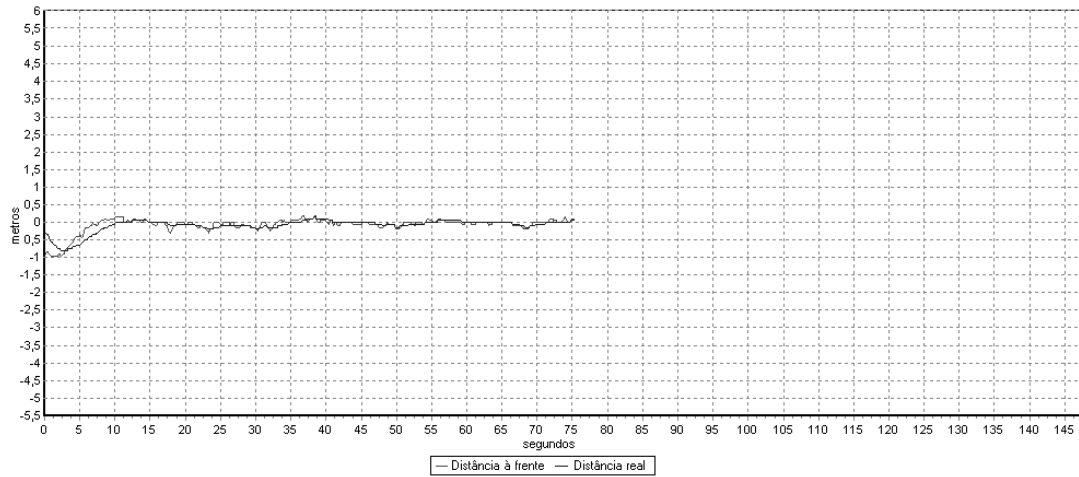


Figura C.2: Erro observado - Teste 1.

Na Figura C.3 o pulverizador foi posicionado a uma distância de $1,8\text{ m}$ do trajeto de trabalho. A velocidade máxima alcançada foi de 15 km/h e o desvio lateral máximo também foi de 25 cm .

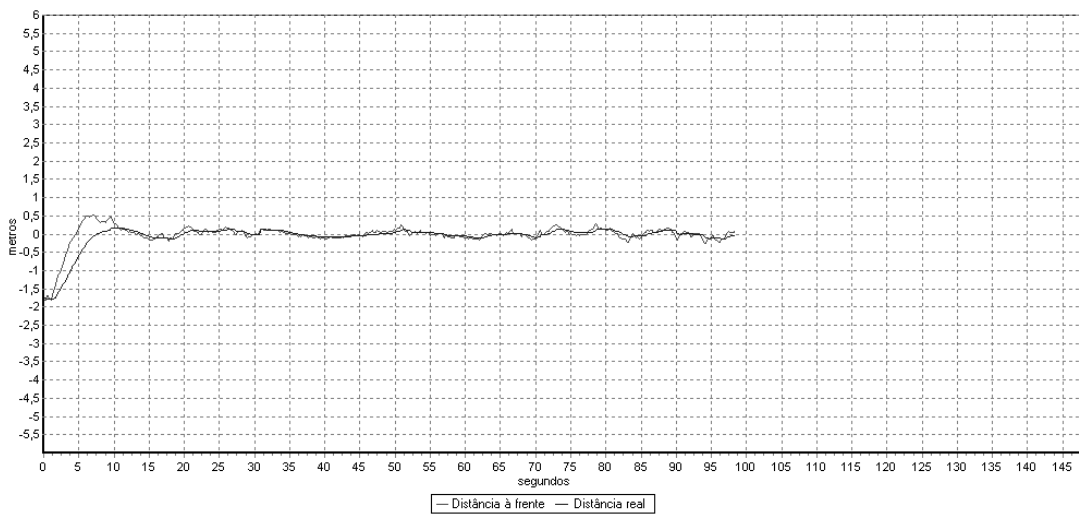


Figura C.3: Erro observado - Teste 2.

No último teste, ilustrado na Figura C.4, o pulverizador foi posicionado a uma distância de 2,7 m do trajeto a ser seguido. A velocidade máxima alcançada foi de 18 km/h e o desvio lateral máximo foi de 30 cm.

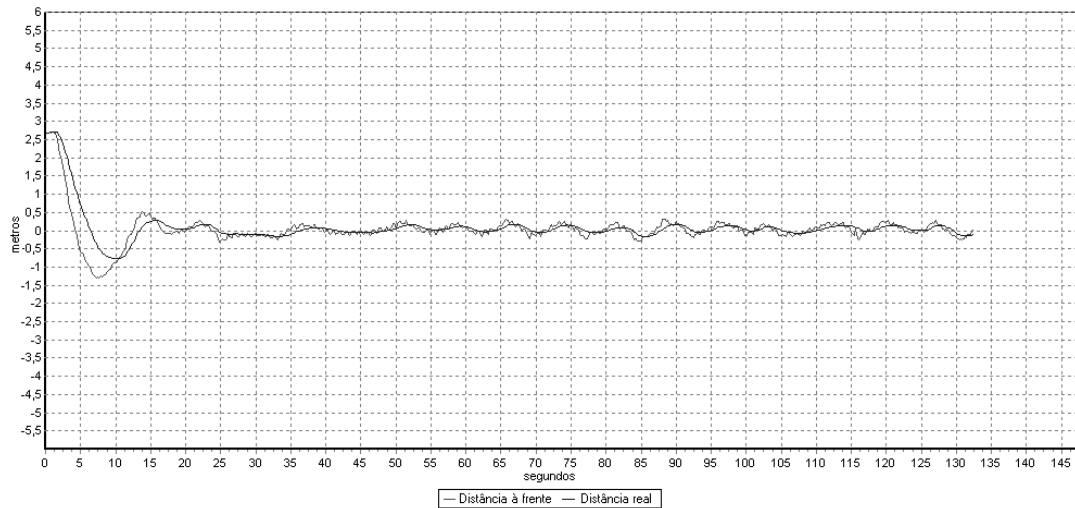


Figura C.4: Erro observado - Teste 3.

Em condições ideais o sistema apresentou variações de no máximo 30 cm na linha de trabalho. Variações iguais a 3 m foram observadas entre as linhas quando a qualidade do sinal recebido pelo receptor GPS estava baixa. Uma das grandes limitações do sistema desenvolvido é a baixa precisão do receptor GPS utilizado. Melhores resultados podem ser alcançados com a utilização de uma estação de correção DGPS. Para o direcionamento automático sobre as linhas de cultivo torna-se necessário a utilização de receptores RTK de precisão centimétrica ou de um sistema baseado em visão computacional.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)