



21 - 25
AGOSTO DE 2016
FORTALEZA - CEARÁ

PROJETO DE HARDWARE DEDICADO PARA O PROCESSAMENTO DE IMAGENS EM APLICAÇÕES DE NAVEGAÇÃO AUTÔNOMA DE ROBÔS MÓVEIS AGRÍCOLAS

CON-2016-0125

Alexandre Padilha Senni, engsenni@usp.br

Mario Luiz Tronco, mltronco@sc.usp.br

Universidade de São Paulo, Av. Trabalhador São-carlense, 400, São Carlos, São Paulo.

Resumo: O emprego de veículos autônomos é uma prática comumente adotada para a melhoria da produtividade no setor agrícola. No entanto, o custo computacional é um fator limitante na implementação desses dispositivos autônomos. A alternativa apresentada nesse trabalho consiste no desenvolvimento de um hardware dedicado para a navegação de robôs móveis agrícolas. O desenvolvimento do projeto foi baseado em um método de extração de características visuais locais através do processamento de imagens coloridas obtidas por uma câmera de vídeo.

Palavras-chave: Visão Computacional, Navegação, FPGA

1. INTRODUÇÃO

Sistemas automáticos baseados em dispositivos robóticos são utilizados para a otimização da produção e execução de tarefas nos diversos setores da indústria. Na Agricultura de Precisão os sistemas automáticos auxiliam em tarefas voltadas tanto para a inspeção e análise de culturas quanto para a locomoção autônoma de robôs móveis (Lulio, 2011). Tais sistemas frequentemente utilizam métodos que empregam técnicas de visão computacional. No entanto, a visão computacional é considerada um problema desafiador para a computação de alto desempenho, tornando a implementação desses sistemas um investimento de alto custo (Molz, 2001). O sistema computacional proposto no presente trabalho consiste em um circuito lógico digital dedicado para a navegação de robôs móveis agrícolas a partir do processamento de imagens fornecidas por uma câmera de vídeo colorida.

O circuito foi implementado através da plataforma de desenvolvimento DE2-70 fabricada pela Terasic. O DE2-70 é baseado no FPGA Cyclone II 2C70 da Altera. O FPGA possui 4 PLLs, 1152000 células de memórias, 150 multiplicadores e 68416 LEs. Além disso, a plataforma possui um conjunto de módulos periféricos integrados ao FPGA para a expansão dos recursos. Dentre esses periféricos, inclui-se o decodificador de vídeo NTSC/PAL/SECAM TV ADV7180 e o decodificador analógico-digital de vídeo ADC7123 que são utilizados para a captura de imagens e monitoramento das etapas de processamento, respectivamente (Terasic, 2015).

2. IMPLEMENTAÇÕES

O hardware proposto no presente trabalho consiste em um circuito dedicado para a navegação autônoma de robôs móveis agrícolas. O hardware desenvolvido pode ser reprogramado de forma a adequar o sistema para as condições específicas de uma determinada área de plantio. O método implementado consiste na classificação de cores e extração das linhas de navegação através de características locais da cena.

O fluxograma do sistema proposto é ilustrado na Fig. 1. A primeira etapa consiste em um circuito para a classificação dos pixels. A segunda etapa consiste em um grupo de processadores morfológicos que podem ser configurados para diversas tarefas, tais como a filtragem e a extração de bordas. Por último, um circuito dedicado permite a extração das linhas de navegação através do método de regressão linear. A linha de navegação é representada na saída do sistema através dos coeficientes angular e linear. A arquitetura proposta também inclui um canal auxiliar, destacado em azul no fluxograma da Fig. 1, que permite a monitoração em tempo real do processamento nas etapas de classificação, processamento morfológico e regressão linear.

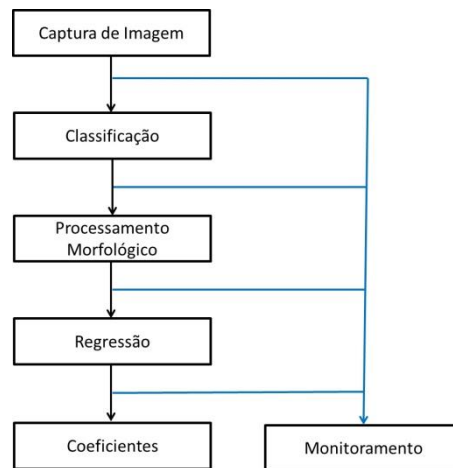


Figura 1. Fluxograma do sistema de navegação.

2.1. CLASSIFICADOR

Na etapa de classificação é utilizado o método de limiarização para classificar os *pixels* da imagem dentre várias classes de interesse. As imagens de cena agrícola tipicamente possuem regiões com características de cor distintas, tais características podem ser utilizadas para a identificação da área navegável. O espaço de cor utilizado para o processo de classificação dos *pixels* é o HSL (*hue*, *saturation* e *lightness*). Esse modelo de cores apresenta vantagens práticas na descrição de cores, permitindo medir a similaridades entre cores de forma simples e intuitiva (Gonzales e Woods, 2007). O circuito desenvolvido para o processo de classificação é ilustrado no diagrama da Fig. 2. A primeira etapa do circuito consiste em um conversor de espaço de cores RGB para o HSL, bloco “rgb_2_hsv”. Os blocos “classificador” configuram um arranjo que é capaz de classificar os *pixels* da imagem dentre três classes definidas pelo usuário. As classes correspondem às saídas C1, C2 e C3.

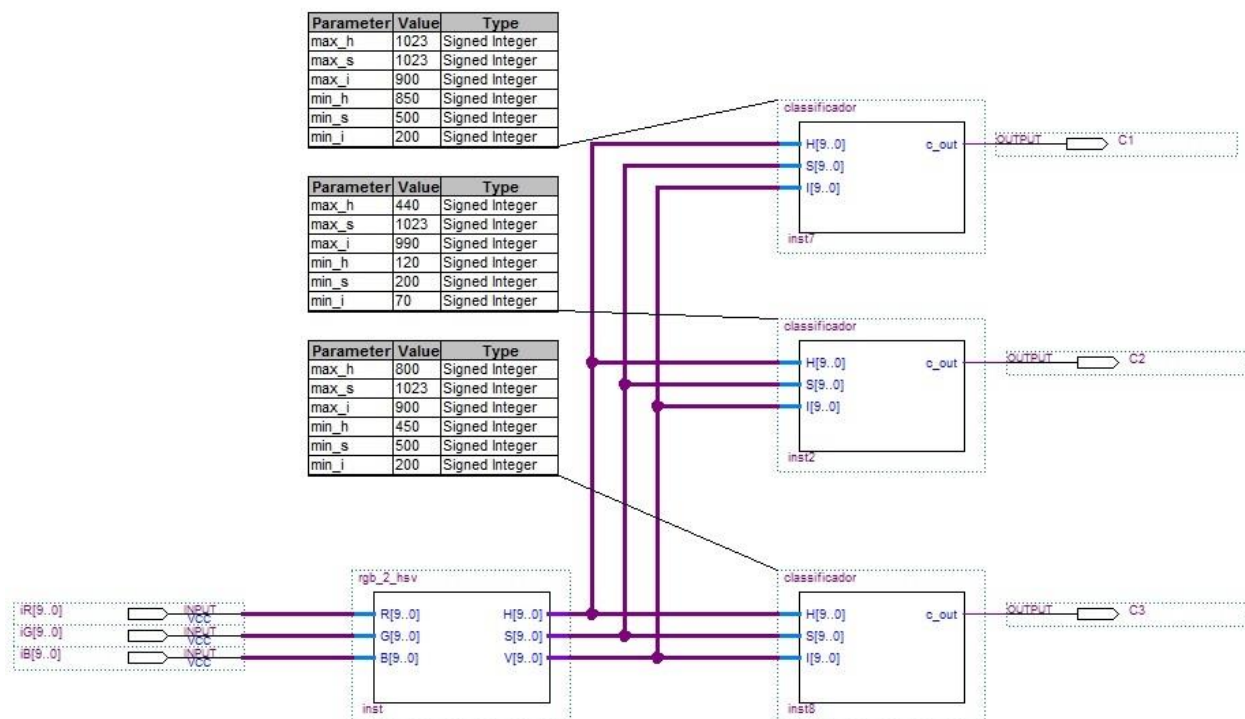


Figura 2. Diagrama do circuito de classificação dos pixels.

O processo de conversão do espaço de cores RGB para o HSL requer operadores matemáticos complexos, tais como, por exemplo, operadores trigonométricos e exponenciais. No entanto, em termos práticos, a implementação desses operadores exige circuitos mais elaborados, exigindo um maior número de recursos lógicos. Uma alternativa comumente adotada para a conversão dos espaços de cores com um custo computacional reduzido é a utilização da aproximação dada pelas equações:

$$H = \begin{cases} 60 \frac{G - B}{\max(R, G, B) - \min(R, G, B)} & \text{se } \max(R, G, B) = R \text{ e } G \geq B \\ 60 \frac{G - B}{\max(R, G, B) - \min(R, G, B)} + 360 & \text{se } \max(R, G, B) = R \text{ e } G < B \\ 60 \frac{B - R}{\max(R, G, B) - \min(R, G, B)} + 120 & \text{se } \max(R, G, B) = G \\ 60 \frac{R - G}{\max(R, G, B) - \min(R, G, B)} + 240 & \text{se } \max(R, G, B) = B \end{cases} \quad (1)$$

$$S = \frac{\{\max(R, G, B) - \min(R, G, B)\}}{\max(R, G, B)} \quad (2)$$

$$L = \frac{1}{2} [\max(R, G, B) + \min(R, G, B)] \quad (3)$$

H, S e L são os componentes do espaço de cores HSL e R, G e B são os componentes do espaço de cores RGB.

A classificação dos *pixels* da imagem no espaço de cores HSL ocorre através de um processo de limiarização. O fundamento desse processo consiste em separar uma imagem $f(x,y)$ através de um limiar T em dois modos, conforme a equação:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (4)$$

A função $g(x,y)$ é a imagem binária limiarizada por T . Os pontos (x,y) da imagem em que $f(x,y) > T$ são chamados de ponto do objeto e os demais de ponto de fundo.

No processo de classificação de imagens coloridas no modelo HSL foi utilizado um limiar mínimo e um limiar máximo para cada componente de cor, conforme as equações:

$$g_H(x, y) = \begin{cases} 1 & \text{se } T_{\min,H} \leq H(x, y) \leq T_{\max,H} \\ 0 & \text{caso contrário} \end{cases} \quad (5)$$

$$g_S(x, y) = \begin{cases} 1 & \text{se } T_{\min,S} \leq S(x, y) \leq T_{\max,S} \\ 0 & \text{caso contrário} \end{cases} \quad (6)$$

$$g_L(x, y) = \begin{cases} 1 & \text{se } T_{\min,L} \leq L(x, y) \leq T_{\max,L} \\ 0 & \text{caso contrário} \end{cases} \quad (7)$$

As funções g_h , g_s e g_l correspondem ao processo de limiarização dos componentes H, S e L, respectivamente. $T_{\min,H}$, $T_{\min,S}$, $T_{\min,L}$, $T_{\max,H}$, $T_{\max,S}$ e $T_{\max,L}$ são os limiares.

O conjunto dos *pixels* que correspondem à classe estabelecida pelo usuário consiste nos *pixels* cujas funções limiarizadas g_h , g_s e g_l são iguais a 1 simultaneamente.

2.2. PROCESSAMENTO MORFOLÓGICO

O circuito para o processamento morfológico apresentado no presente trabalho foi baseado na arquitetura proposta por Pedrino (Pedrino, 2003). A arquitetura consiste em um arranjo de múltiplos estágios em uma arquitetura *pipeline* serial para o processamento de imagens em tempo real por operações básicas de morfologia matemática, tais como erosão e dilatação.

As operações são realizadas através dos blocos “processador_morfologico”. Cada bloco possui um conjunto de entradas para a imagem binária, obtida através do circuito de classificação, e sinais de sincronismo de vídeo. As saídas correspondem à imagem binária processada e seus respectivos sinais de sincronismo. Os sinais de sincronismo, nas saídas dos blocos, são atrasados em relação aos sinais de entrada em um período que corresponde à latência necessária para o processamento dos *pixels*. Dessa forma, essas unidades de processamento podem ser conectadas em arranjos do tipo cascata, formando uma sequência de operações morfológicas básicas que correspondem ao algoritmo morfológico proposto. A simbologia adotada para a unidade de processamento “processador_morfologico” é ilustrada na Fig. 3. A operação realizada pela unidade de processamento e o elemento estruturante utilizado são estabelecidos através dos parâmetros Op e St, respectivamente.

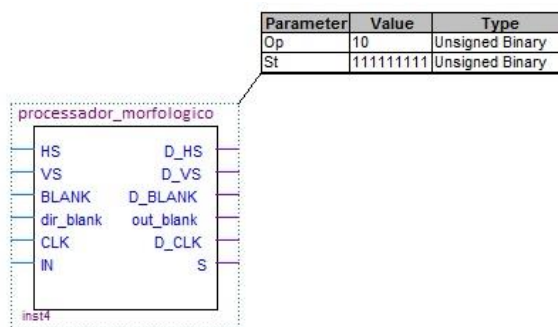


Figura 3. Simbologia da unidade de processamento morfológico.

O circuito da unidade de processamento é constituído por três blocos, conforme ilustrado na Fig. 4. São esses: “memoria_de_deslocamento”, “operador_logico” e “delay_sincronismo”.

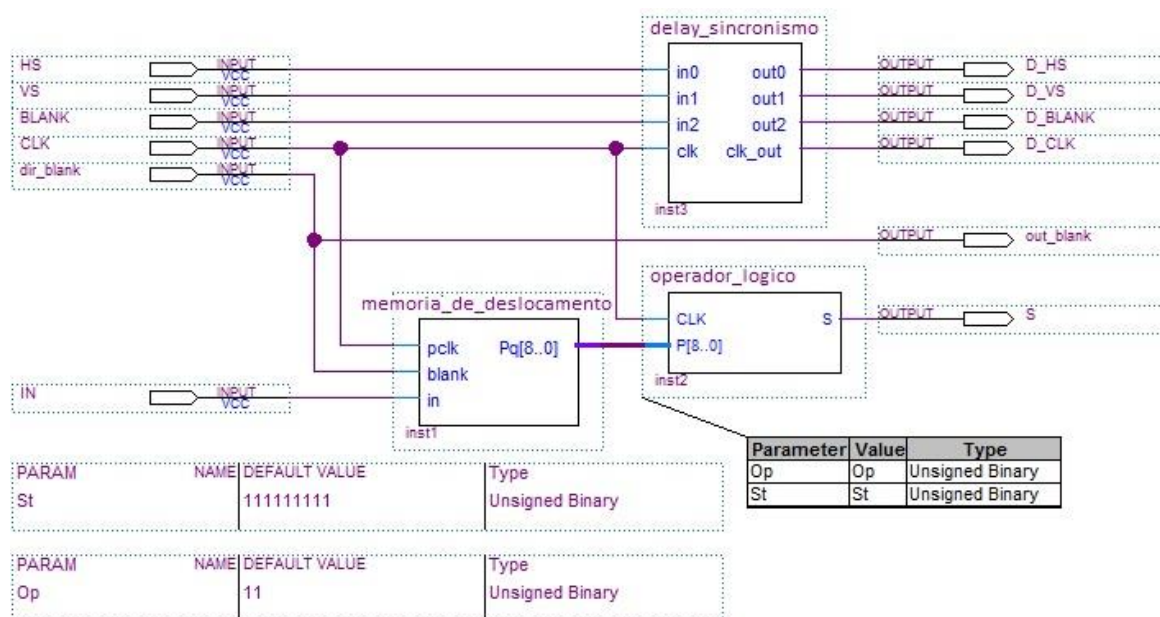


Figura 4. Circuito da unidade de processamento morfológico.

O bloco “memoria_de_deslocamento” é um arranjo de memórias sequenciais que permite o “operador_logico” acessar os *pixels* de uma janela de dimensão 3×3 . Essa janela é deslocada sobre a imagem à medida que os *pixels* são encaminhados ao processador. Essa estratégia permite o acesso simultâneo dos *pixels* que compõem a janela, permitindo o processamento da imagem em um fluxo contínuo (Lim e Kleeman, 2012). Na Fig. 5 é ilustrado o seu princípio de funcionamento.

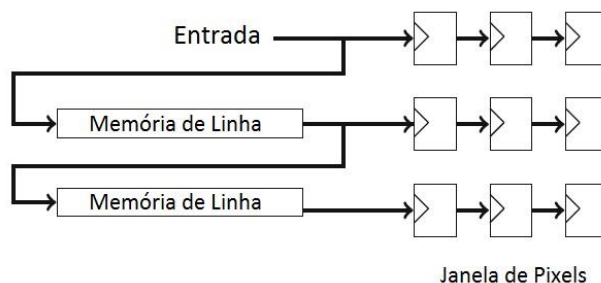


Figura 5. Princípio de funcionamento do circuito de acesso à janela de pixels.

O circuito do bloco “memoria_de_deslocamento” foi implementado através de um arranjo de blocos gerados automaticamente através da ferramenta MegaWizard Plug-in Manager do software Quartus II.

O bloco “operador_logico” é um circuito lógico combinacional desenvolvido em VHDL que realiza a operação morfológica no pixel central da janela de pixels. A operação é realizada conforme os parâmetros estabelecidos para a unidade de processamento.

O bloco “delay_sincronismo” é o circuito responsável por realizar o retardo dos sinais de sincronismo de vídeo. O seu circuito também foi implementado através de um arranjo de blocos gerados automaticamente pela ferramenta MegaWizard Plug-in Manager.

A fim de atender as necessidades de algoritmos diversos, as operações lógicas também podem ser combinadas em conjunto com as operações morfológicas. Na Fig. 6 é ilustrado o arranjo proposto para a extração de bordas. Esse arranjo é composto por três unidades de processamento morfológico e uma porta lógica XOR. A extração de bordas é um requisito fundamental para o propósito de extração das linhas de navegação realizada na próxima fase do sistema. O fluxograma do algoritmo é ilustrado na Fig. 7. Para que a operação lógica XOR seja realizada entre imagens sincronizadas, a operação “nulo” foi realizada em paralelo com a segunda erosão.

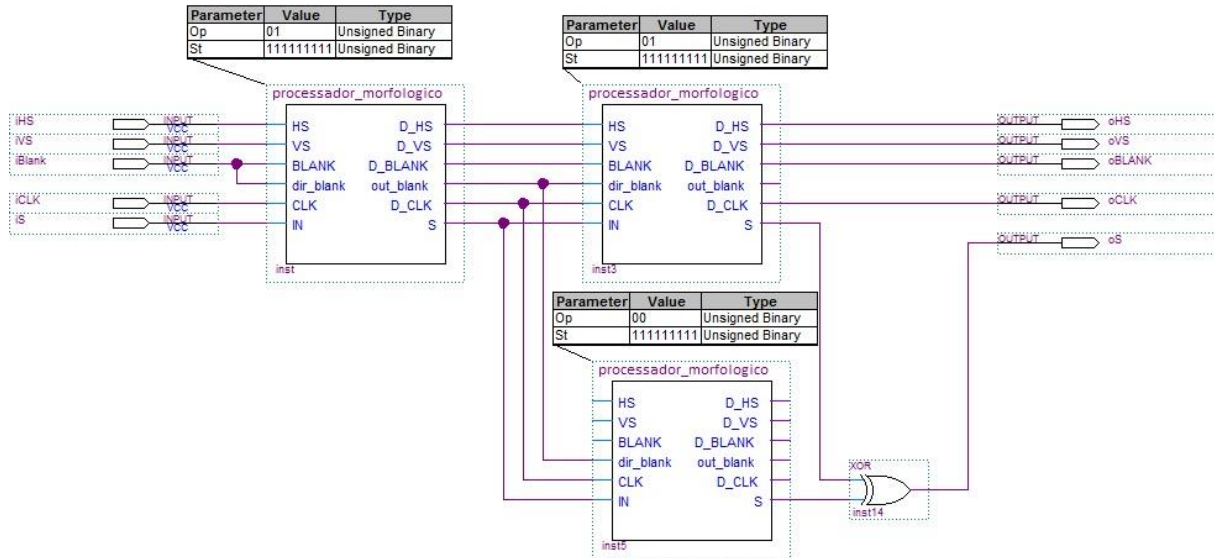


Figura 6. Arranjo para a extração de bordas.

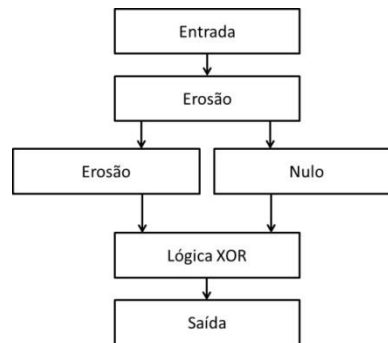


Figura 7. Fluxograma do algoritmo para extração de bordas.

2.3. EXTRAÇÃO DAS LINHAS DE NAVEGAÇÃO

Para a extração das linhas de navegação, no presente trabalho foi desenvolvido um circuito dedicado baseado no método de regressão por mínimos quadrados. O método consiste em uma análise estatística com o objetivo de verificar a existência de uma relação funcional entre uma variável dependente com uma ou mais variáveis independentes.

Os parâmetros da regressão são os fatores α e β , conforme as equações:

$$\alpha = \frac{\sum X^2 \sum Y - \sum (XY) \sum X}{n \sum X^2 - (\sum X)^2} \quad (8)$$

$$\beta = \frac{n \sum (XY) - \sum X \sum Y}{n \sum X^2 - (\sum X)^2} \quad (9)$$

Os fatores α e β representam a interceptação da reta com o eixo vertical e a inclinação da reta em relação ao eixo horizontal, respectivamente. As variáveis X e Y são as coordenadas dos pixels que compõem os objetos na imagem binária e n é o número de amostras.

No contexto do presente trabalho, a análise consiste no cálculo dos fatores que definem as retas que separam a região navegável da região não navegável. Posteriormente, esses parâmetros são utilizados para a estimativa da linha de navegação. O processo de regressão ocorre em duas regiões distintas da imagem que, de acordo com as características

locais e do posicionamento da câmera, são definidas pelo usuário através de parâmetros. Tais regiões correspondem à área onde ocorrem com maior frequência as linhas que contornam a região navegável, minimizando os erros no processo de extração da linha de navegação.

O circuito desenvolvido para o cálculo da estimativa da linha de navegação é ilustrado na Fig. 8. O circuito é composto pelos blocos “regressao”, “filtro” e “linha_media”.

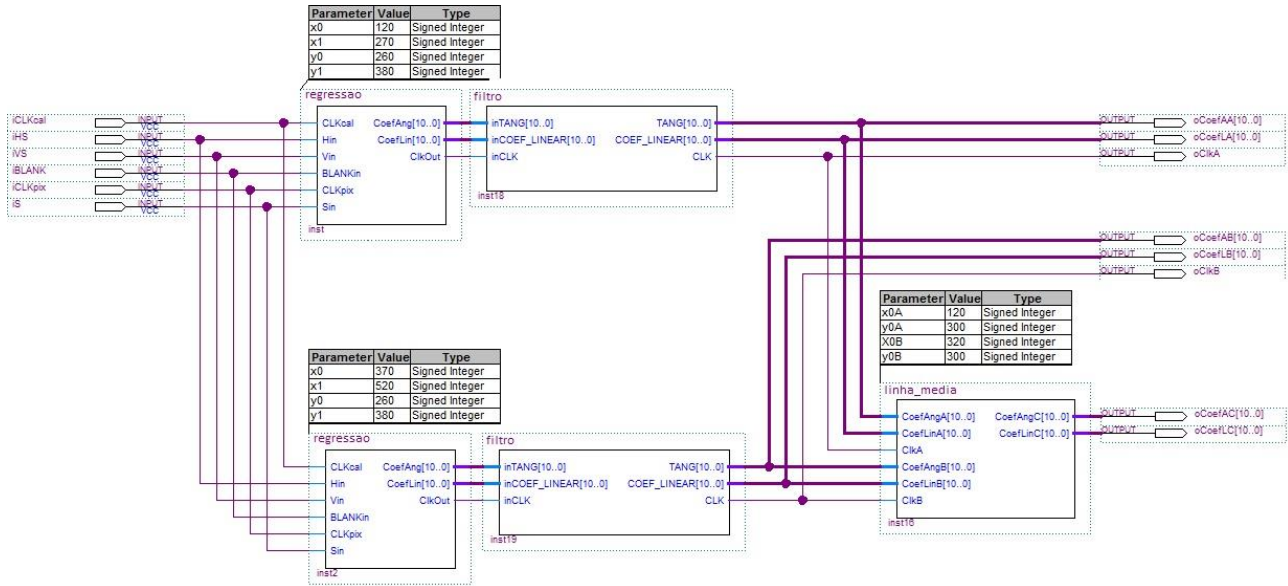


Figura 8. Circuito para a extração das linhas de navegação.

O cálculo dos parâmetros das retas é realizado através dos blocos “regressao”, sendo um bloco responsável pela reta que delimita a região navegável à direita da imagem e o outro à esquerda. O circuito foi implementado de forma que α mede a tangente do ângulo θ formado a partir do eixo vertical da imagem, conforme ilustrado na Fig. 9. Os parâmetros x_0 , x_1 , y_0 e y_1 correspondem à área delimitada para o cálculo da regressão.

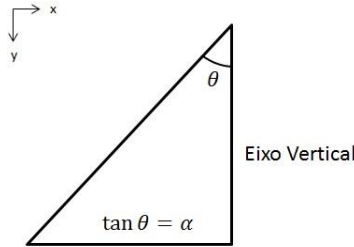


Figura 9. Orientação do parâmetro α .

Os blocos “filtro” realizam uma média aritmética em cada parâmetro obtido no processo de regressão, minimizando os efeitos causados por resultados muito dispersos. A média é realizada em conjuntos de vinte amostras, embora, de acordo com a aplicação, esse parâmetro possa ser modificado no projeto.

Na etapa final do circuito para a extração das linhas de navegação, a linha média que define a orientação do veículo é calculada pelo bloco “linha_media” através das equações:

$$\dot{\alpha} = \frac{\alpha_A + \alpha_B}{2} \quad (10)$$

$$\dot{\beta} = \frac{(\beta_A + x_{0,A} - y_{0,A} \times \alpha_A) + (\beta_B + x_{0,B} - y_{0,B} \times \alpha_B)}{2} \quad (11)$$

Os fatores $\dot{\alpha}$ e $\dot{\beta}$ são os coeficientes angular e linear, respectivamente, da função que define a linha de navegação. Os fatores α_A e β_A são os coeficientes que correspondem à reta que delimita a área navegável à esquerda da imagem e os fatores α_B e β_B correspondem ao lado direito da imagem. Os parâmetros $x_{0,A}$, $y_{0,A}$, $x_{0,B}$ e $y_{0,B}$ são os mesmos utilizados no bloco “regressao” para delimitar a região de interesse. Os subscritos A e B desses parâmetros indicam o bloco para a regressão no lado esquerdo e direito, respectivamente.

3. RESULTADOS

A seguir são apresentados alguns resultados obtidos em ensaios realizados na fase de desenvolvimento do projeto. Os testes foram realizados nas etapas de classificação, processamento morfológico e extração das linhas de navegação. As imagens foram alimentadas no sistema através de um computador pessoal com uma saída de vídeo analógica.

Nos testes do circuito de classificação, os *pixels* das imagens de cenas agrícolas foram classificados em três classes correspondentes ao céu, solo e vegetação. Os parâmetros máximos e mínimos dos componentes de cores utilizados nos testes são apresentados na Tab. 1 e os resultados obtidos nessa etapa são ilustrados na Fig. 10.

Tabela 1. Parâmetros do circuito de classificação de cores.

			Mínimo (10 bits)	Máximo (10 bits)
Céu	Azul	H	480	680
		S	120	1023
		L	160	840
	Branco	H	0	1023
		S	0	1023
		L	840	1023
Solo	Vermelho ($H \geq 0$)	H	0	140
		S	160	1023
		L	160	840
	Vermelho ($H < 0$)	H	880	1023
		S	160	1023
		L	160	840
Vegetação	Verde	H	180	400
		S	120	1023
		L	160	840

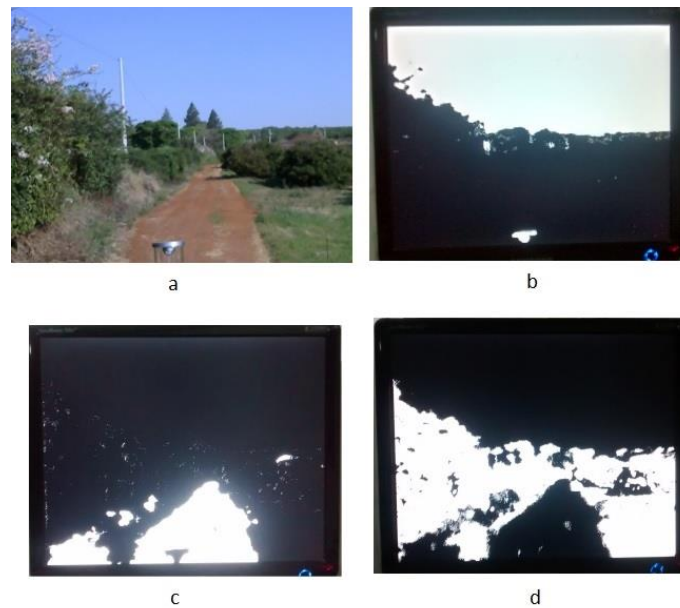


Figura 10. Resultado do processo de classificação dos pixels: (a) imagem original; (b) imagem binária da classe correspondente à região do céu; (c) imagem binária da classe correspondente à região do solo; (d) imagem binária da classe correspondente à região da vegetação.

Na Fig. 11 é ilustrado o resultado obtido através de uma sequência de operações morfológicas para o agrupamento dos *pixels* e extração das bordas. A imagem binária utilizada no processamento morfológico corresponde à classe do solo.

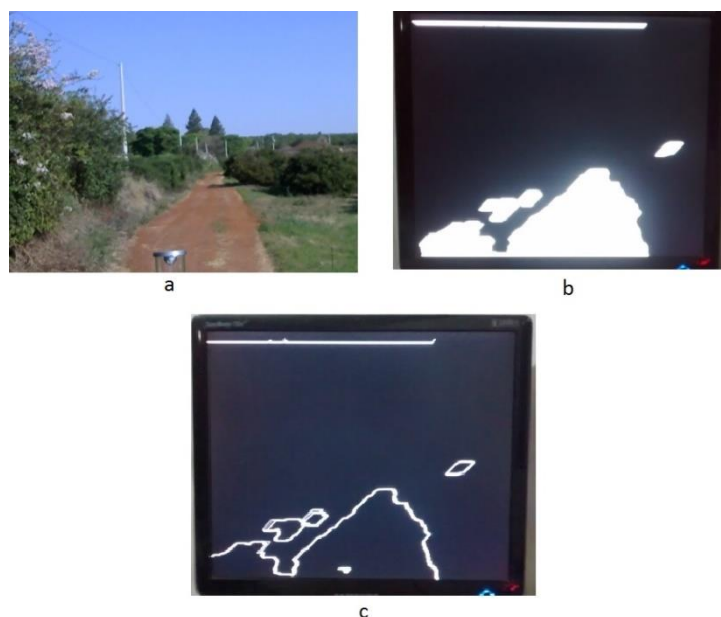


Figura 11. Resultado do processamento morfológico: (a) imagem original; (b) agrupamento dos pixels da classe correspondente à região do solo; (c) extração de bordas.

Na Fig. 12 são ilustrados os resultados do processo de extração das linhas de navegação através de imagens de teste utilizadas para a validação do circuito. Os retângulos destacados em vermelho na imagem de saída do sistema correspondem às áreas de aplicação da regressão, os pixels destacados em verde correspondem ao processo de extração de bordas e as linhas destacadas em azul correspondem ao resultado do processo de regressão, sendo a linha central correspondente à orientação do veículo.

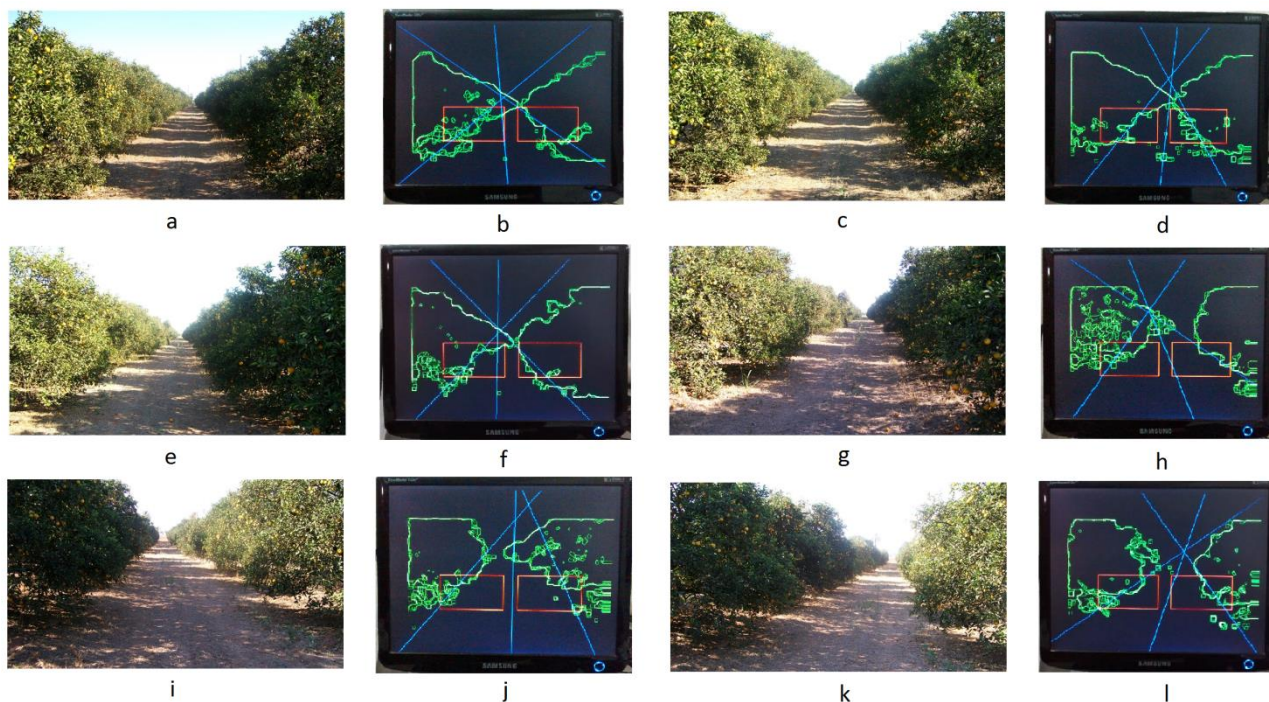


Figura 12. Resultado do processo de extração de bordas: (a), (c), (e), (g), (i) e (k) imagem original; (b), (d), (f), (h), (j) e (l) imagem de saída do sistema.

4. CONCLUSÕES

O emprego de circuitos dedicados para a navegação de robôs móveis agrícolas é uma alternativa barata para aplicações cujas cenas apresentam características visuais relevantes que podem ser extraídas para determinar a orientação do veículo. A utilização de um *hardware* dedicado elimina a necessidade de embarcar computadores industriais, reduzindo o custo final do projeto. Além disso, o sistema de navegação apresentado permite o processamento de imagens em tempo real, possibilitando o seu emprego em aplicações que exigem velocidade de deslocamento relativamente elevada, como, por exemplo, a pulverização de lavouras.

5. REFERÊNCIAS

- CHOI, K. H., HAN, S. K., HAN, S. H., PARK, K., KIM, K., KIM, S. Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields, *Computers and Electronics in Agriculture*, Volume 113, April 2015, Pages 266-274, ISSN 0168-1699.
- ENGLISH A., ROSS P., BALL D., CORKE P. Vision based guidance for robot navigation in agriculture (2014) *Proceedings - IEEE International Conference on Robotics and Automation*, art. no. 6907079, pp. 1693-1698.
- GONZALES, R. C. & WOODS, R. E. *Digital Image Processing*. 3.ed. New Jersey, EUA: Prentice-Hall Inc. 2007. ISBN 013168728X.
- HAIGANG, Y., et al. Review of Advanced FPGA Architectures and Technologies. *Journal of Electronics*, v. 31, n. 5, 2014.
- LIM, Y. K., KLEEMAN, L. Algorithmic methodologies for FPGA-based vision. *Machine Vision and Applications*. 2012.
- LULIO, L. C. *Técnicas de Visão Computacional Aplicadas ao Reconhecimento de Cenas Naturais e Locomoção Autônoma em Robôs Agrícolas Móveis*. Dissertação – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2011.
- MENG Q., QIU R., HE J., ZHANG M., MA X., LIU G., Development of agricultural implement system based on machine vision and fuzzy control, *Computers and Electronics in Agriculture*, Volume 112, Março 2015, p. 128-138.
- MOLZ, R. F. *Uma Metodologia para o Desenvolvimento de Aplicações de Visão Computacional utilizando um projeto conjunto de Hardware e Software*. Tese – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2001.
- PEDRINO, E. C. *Arquitetura Pipeline para Processamento Morfológico de Imagens Binárias em Tempo Real Utilizando Dispositivos de Lógica Programável Complexa*. Dissertação – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2003.
- TERASIC. Altera DE2-70 Board. Disponível em: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?No=226>. Acesso em: 22 de jun. 2015.
- TRIEU, D.B.K., MARUYAMA, T. Real-time color image segmentation based on mean shift algorithm using an FPGA (2015) *Journal of Real-Time Image Processing*, 10 (2), pp. 345-356.
- TTOFIS, C., KYRKOU, C., THEOCHARIDES, T. A hardware-efficient architecture for accurate real-time disparity map estimation (2015) *ACM Transactions on Embedded Computing Systems*, 14 (2), p. 36.

6. RESPONSABILIDADE AUTURAL

Os autores são os únicos responsáveis pelo conteúdo deste trabalho

DEDICATED HARDWARE DESIGN FOR IMAGE PROCESSING IN APPLICATIONS OF AUTONOMOUS AGRICULTURAL ROBOT NAVIGATION

CON-2016-0125

First Author's Name, e-mail¹

Second Author's Name, e-mail²

Third Author's Name, e-mail²

¹Institution and address for first author

²Institution and address for second and third author

Abstract. *The use of autonomous vehicles is a generally adopted practice to improve the productivity in the agriculture sector. However, the computer requirements are a limiting factor for implementation of these autonomous devices. The alternative shown in this paper is the design of a dedicated hardware for the autonomous agricultural robot navigation. The project development was based on a local visual feature extraction method by processing digital images obtained from a color video camera.*

Keywords: *Computer Vision, Navigation System, FPGA*